

ბ. კიზირია, ა. ვაშაკიძე

ამოცანათა კრებული ხელოვნურ ინტელექტში

(I ნაწილი)



დამტკიცებულია სტუ-ს
სარედაქციო-საგამომცემლო
საბჭოს მიერ

ნაშრომში წარმოდგენილია კომპიუტერულ მეცნიერებათა ერთერთი მიმართულების „ხელოვნური ინტელექტის“ ამოცანათა კრებული სხვადასხვა საპრებლემო არისათვის. მოცემულია ამოცანის დასმისა და მისი კომპიუტერული გადაწყვეტის გზები. პროგრამულ ინსტრუმენტად გამოყენებულია ლოგიკური პროგრამირების ალგორითმული ენის „პროლოგ“-ს უახლესი ვერსია „Strawberry Prolog“ (Light edition, version 2.5., for Windows XP/NT). ვერსია შემუშავებულია სოფიის უნივერსიტეტში 2005 წელს, დოდბრევის მიერ და გამოყენებულია მრავალ ქვეყანაში ხელოვნური ინტელექტის ამოცანების გადასაწყვეტად.

სახელმძღვანელოში ამოცანები დაჯგუფებულია თემატიკის მიხედვით: რელაციური პროგრამის ტანში ჩაშენებული ცოდნის ბაზების პროექტირება, იმიტატორის გრაფიკული ოპერატორების (პრედიკატების) გამოყენება, გადაწყვეტილების მიღების (მართვის) პროცესების მოდელირება, ტესტ-პროგრამების პროექტირება და სხვა. მოცემული ნაშრომის პირველ ნაწილში წარმოდგენილი ამოცანები, ავტორთა აზრით, სასურველია გამოყენებული იქნეს დისციპლინის: „ხელოვნური ინტელექტის საფუძვლების“ შესწავლისას, I სემესტრის ლაბორატორიული სამუშაოების ჩასატარებლად. II სემესტრის შესაბამისი ამოცანები მოყვანილი იქნება ნაშრომის მეორე ნაწილში.

სახელმძღვანელო განკუთვნილია სტუ-ს იმს-ს ფაკულტეტის სტუდენტებისა და მაგისტრებისათვის. დადინტერესებს აგრეთვე მეცნიერების ამ მიმართულებით მომუშავე სპეციალისტებს და პროფესორ-მასწავლებელთა ფართო წრეს.

რეკენსენტები:

პროფესორი ა. ფრანგიშვილი

პროფესორი ნ. ლომინაძე

„ხელოვნური ინტელექტი“, კომპიუტერულ მეცნიერებათა ერთერთი მთავარი მიმართულებაა. მეცნიერების ამ დარგმა თავისი განვითარების 50 წლიანი გზა განვლო. ეს გზა არ იყო მარტივად გასავლელი. იყო ბევრი იმედგაცრუება, ამ მეცნიერებისადმი სკეპტიკური მიდგომა, მაგრამ იყო ასევე დიდი გამარჯვებებიც. XXI საუკუნეს „ხელოვნური ინტელექტი“ შეეგება, როგორც ზღადასული, დიდი პოტენციალის მქონე მეცნიერება. მან შეაღწია ადამიანის გონებრივი მოღვაწეობის თითქმის ყველა სფეროში. დღეისათვის გამოიკვეთა „ხელოვნური ინტელექტის“ (ხი) მრავალი, როგორც თეორიული ასევე პრაქტიკული მიმართულება: თეორემების ავტომატური დამტკიცება, სახეობათა ამოცნობა, ნეირონული ქსელების დაპროექტება, მანიპულატორები და რობოტები, მართვის ამოცანები, ხი-ს „ყველაზე ჯანმრთელი შვილი“, როგორც მას უწოდებენ ცივილიზებულ სამყაროში, ექსპერტული სისტემები (ეს) და ა.შ.

ნაშრომის სპეციფიკიდან გამომდინარე, ჩვენ ძირითადად შევხებით ინტელექტუალური სისტემების დაპროექტების, კერძოდ კი ექსპერტული სისტემების პროექტირების საკითხებს. ექსპერტული სისტემები ეს ადამიანი-ექსპერტის ცოდნაზე დაფუძნებული ინტელექტუალური სისტემებია. სპეციალურ ლიტერატურაში ექსპერტული სისტემები მოიხსენიება, როგორც დიაგნოსტიკური, მაკონსულტირებელი სისტემები.

თანამედროვე ცივილურ სამყაროში, ექსპერტული სისტემები გამოიყენება ადამიანის გონებრივი მოღვაწეობის თითქმის ყველა მიმართულებით: მართვა (გადაწყვეტილების მიღება), მედიცინა, გეოლოგია, დიზაინი, ისტორია, არქეოლოგია და ა.შ. მსოფლიოს მრავალ ქვეყანაში შექმნილია და ფუნქციონირებს, როგორც სახელმწიფო ასევე კერძო სამეცნიერო მიმართულების დაწესებულებები და ფირმები, რომლებიც აპროექტებენ ეს-ს. სპეციალისტებს, რომლებიც მოღვაწეობენ ამ მიმართულებით, უწოდებენ „ცოდნის ინჟინრებს“, ხოლო ეს-ს დაპროექტების პროცესს „ცოდნის ინჟინერიას“. გამომდინარე ეს-ს დანიშნულებისა და საპრობლემო არის სირთულისა, დაპროექტების პროცესი დროში შესაძლებელია გაიწელოს რამდენიმე თეიდან, რამდენიმე წლამდეც კი. სამუშაო მოითხოვს დიდ ინტელექტუალურ და ფინანსურ დანახარჯებს. ზოგიერთი სახეობის მაგ. მფრინავი ან წყალქვეშა ობიექტების ამოცნობა, სამხედრო და კოსმოსური პროექტის დაფინანსება მრავალი მილიარდი დოლარი ჯდება და ფინანსდება შესაბამისი უწყებებისა (პენტაგონი, NASA) და სახელმწიფოს მიერ.

ექსპერტული სისტემების დაპროექტებისას შესაძლებელია გამოიყოს ორი ძირითადი მომქმედი პირი (ან პირთა ჯგუფი): „ცოდნის ინჟინერი“ და მოცემული დარგის (საპრობლემო არის) ექსპერტი. ეს-ს დაპროექტების ეტაპები აღწერილია მრავალ სპეციალურ ლიტერატურაში, ასევე მოცემული ნაშრომის ერთერთი თანაავტორის წიგნში

„ხელოვნური ინტელექტის საფუძვლები“ [7], რომელიც გამოსცა სტუ-ს გამომცემლობამ 2005 წელს.

მოცემული სახელმძღვანელოს ძირითადი დანიშნულებაა [7] წიგნში წარმოდგენილი თეორიული მასალის წარმოჩენა ამოცანებისა და პროგრამების სახით. ეს კი თავის მხრივ მნიშვნელოვნად ააძლიერებს ამ დისციპლინით დაინტერესებული სტუდენტების, მაგისტრების, მასწავლებლების უნარ-ჩვევებს პრაქტიკული და ლაბორატორიული სამუშაოების ჩატარებისას.

აღსანიშნავია ერთი მნიშვნელოვანი გარემოებაც. როგორც ზემოთ დასახელებული სახელმძღვანელო „ხელოვნური ინტელექტის საფუძვლები“ [7], ასევე წინამდებარე ნაშრომიც ამ ფორმატით (წიგნში მოცემული თეორიული მასალის მოცულობით, დასმული და გადაწყვეტილი ამოცანების სიუხვით) ქართულ ენაზე შესრულებული პირველი ნაშრომია. ამის გამო იგი რასაკვირველია არ იქნება დაზღვეული გარკვეული ხარვეზებისგან. წიგნის ავტორთა უმორჩილესი თხოვნაა: გთხოვთ, მოგვაწოდოთ თქვენი შენიშვნები და წინადადებები. მათ ჩვენ აუცილებლად გაეითვალისწინებთ წიგნის შემდგომ გამოცემებში.

ასევე, ნაშრომის ავტორთა ჯგუფი, დიდ მადლობას უხდის ყველა იმ კეთილისმსურველ პირს, რომელმაც პროფესიული რჩევა არ დაიშურა, თავისი ცოდნა და გამოცდილება უხვად გასცა, ამ საშური საქმის დასასრულებლათ. მათ შორის მრავლად არიან ჩვენი ნიჭიერი სტუდენტები და მაგისტრები, ჩვენი ძვირფასი კოლეგები და უფროსი თაობის ჩვენი პედაგოგები: ბი ნ. ლომინაძე, ბი კ. კამკამიძე, ბი ა. ფრანგიშვილი, ბი ზ. გასიტაშვილი, ბი გ. სურგულაძე, ბი გ. გოგიჩაიშვილი და სხვები.

თავი 1. ხელოვნური ინტელიქტის სისტემების დაპროექტების პროგრამული საშუალებები

ხელოვნური ინტელექტის (ხი) სისტემების დაპროექტება შრომატევადი და ხანგრძლივი პროცესია. ეს პროცესი დამოკიდებულია საპრობლემო არეზე (მართვა, მათემატიკა, ფიზიკა, ქიმია, არქიტექტურა, მოდელირება, დიზაინი, გეოლოგია და ა.შ.), გადასაწყვეტი ამოცანის სირთულეზე, სისტემის დამპროექტებლის („ცოდნის ინჟინრის“) კვალიფიკაციაზე, დარგის ექსპერტის მზაობის ხარისხზე, ითანამშრომლოს „ცოდნის ინჟინერთან“, მათ ხელთ არსებული ტექნიკური და პროგრამული საშუალებების ხარისხზე და ა.შ.

ჩვენ ძირითადად შეეხებით ხი-ს სისტემების დაპროექტების პროგრამულ საშუალებებს. პირობითად, ბუნებაში არსებული ყველა ალგორითმული ენა შესაძლებელია დაიყოს ორ კლასად:

1. იმპერატიული ალგორითმული ენები. მათ მიეკუთვნება ისეთი ალგორითმული ენები როგორებიცაა: ასემბლერი, პასკალი, ფორტრანი, სი-შარპი, ჯავეა, ადა და სხვა. ამ ენებს „როგორ“ ტიპის ალგორითმულ ენებს უწოდებენ.

2. დეკლარაციული ალგორითმული ენები. ამ სახის ალგორითმული ენაა პროლოგი (Prolog-ი). იგი „რა“ ტიპის ალგორითმული ენაა. ტერმინების „როგორ“ და „რა“ არსი დაწერილობით არის აღწერილი [7]-ს შესავალ ნაწილში.

ნებისმიერ ალგორითმულ ენაზე დაწერილი პროგრამა ეფუძნება გარკვეულ ლოგიკას, რომელიც ასახავს გადასაწყვეტი პრობლემის არსს. მათემატიკური ლოგიკის ელემენტების გამოყენება, უშუალოდ ალგორითმული ენის ტრანსლიატორის (იმპიტატორის) დასაპროექტებლად, განხორციელდა სწორედ ალგორითმული ენა „Prolog“-ს შექმნისას. დაახლოებით 40 წლის წინათ, მარსელში შექმნილ ალგორითმულ ენას საფუძვლად დაედო XIX საუკუნეში შემუშავებული „პრედიკატების აღრიცხვის“ თეორიული პოსტულატები, კერძოდ კი პირველი რიგის პრედიკატები. „პრედიკატების აღრიცხვა“-ს ეძღვნება დასახელებული [7] სახელმძღვანელოს მეორე თავი. სიტყვა „Prolog“-ი ხელოვნურად შექმნილი სიტყვაა და ორი ინგლისური სიტყვის: **Programming** და **logic** შერწყმას წარმოადგენს.

„Prolog“-ი მიზნობრივი პროგრამირების ენაა. ხი-ს სისტემის დამპროექტებელი გაანალიზებს რა დასახული პრობლემის არსს, პირველ რიგში აფორმირებს ცოდნის ბაზას (ც/ბ). ც/ბ შედგება ფაქტებისა (Facts) და წესებისგან (Rules). შემდგომ ეტაპზე, დარგის ექსპერტთან კონსულტაციების ბაზაზე, იგი აყალიბებს მიზნების (Goals) გარკვეულ სიმრავლეს. როგორც ზემოთ აღნიშნეთ, ამოცანების გადაწყვეტისას, ვიყენებთ „პროლოგ“-ს უახლეს ვერსიას „Strawberry Prolog“-ს (ე.წ. მარწყვის პროლოგს), რომელიც „Windows XP“ გარემოში მუშაობს. „Strawberry Prolog“-ს სამუშაო გარემოს 1.1. ქვეთავში დაწერილობით განვიხილავთ.

ხელოვნური ინტელექტის სისტემის დაპროექტების ორივე სუბიექტი: „ცოდნის ინჟინერი“ და ღირსის ექსპერტი ინდივიდები არიან. მათ გააჩნიათ თავისი შეფასებად გამოხატული ინდივიდუალიზმი: ცოდნა, გამოცდილება, ინტუიცია, სხვადასხვა ჩვევები და ა.შ. გადაწყვეტილების მიღებისას ისინი იყენებენ ზემოთ ჩამოთვლილ კომპონენტებს. გადაწყვეტილების მიღების პროცესი არჩევის პროცესია. არჩევა წარმოებს ალტერნატივების სიმრავლიდან. არჩევის პროცესისას $\min$ ორი ალტერნატივა მაინც უნდა არსებობდეს. წინააღმდეგ შემთხვევაში არჩევის პროცესი შეუძლებელია. ინდივიდის გადაწყვეტილებაზე მოქმედებს მრავალი ფაქტორი. მათ შორის შესაძლებელია გამოყოფით ორი ძირითადი:

1. სუბიექტური ფაქტორები. ეს ფაქტორები მომდინარეობს უშუალოდ ინდივიდისგან: მისი ცოდნა, ინტუიცია, განწყობა, სხვადასხვა ჩვევები და ა.შ.

2. ობიექტური ფაქტორები. ეს ფაქტორები არსებობენ ინდივიდისგან დამოუკიდებლად და მათზე ზემოქმედება ინდივიდს არ შეუძლია. ასეთი ფაქტორებია მაგ-თ: ქვეყნის ეკონომიკური მდგომარეობა, ეროვნული ვალუტის მიმართება დოლარის კურსთან, კლიმატური პირობები და ა.შ.

ყოველივე ზემოთ თქმულის გათვალისწინებით, ინდივიდი იღებს მმართველ გადაწყვეტილებებს.

გარდა პროგრამირების ენებისა, ხი-ს სისტემების დამპროექტებელი ასევე იყენებს პროგრამულ ინტერფეისულ საშუალებებს, ბაზების მართვის მის ხელთ არსებულ რომელიმე სისტემას (Microsoft Access, SQL, ორაკლი და სხვა).

ხელოვნური ინტელექტის სისტემების პროექტირების ინსტრუმენტად ჩვენ ვიყენებთ „Strawberry Prolog“-ს. ეს სრულებით არ ნიშნავს იმას, რომ ხი-ს სისტემების პროექტირება შეუძლებელია სხვა ალგორითმული ენების ბაზაზე. მრავალი მაღალი დონის ხი-ს სისტემებია შექმნილი ისეთ ალგორითმულ ენებზე, როგორებიცაა: Java, სი-შარპი, Lisp-ი და სხვა.

„Strawberry Prolog“-ს იმიტატორი ეფუძნება: პირველი რიგის პრედიკატებს, დასკვნის მიღების დედუქციურ მეთოდს (უკუ-ჯაჭვი). იტერაციული პროცესების (ციკლი) ორგანიზებისათვის გამოყენებულია რეკურსიის მექანიზმი. პროლოგ-პროგრამის ყველა ობიექტს ეწოდება განზოგადებული სახელი „ფერმი“. პროლოგის იმიტატორი მას წარმოაჩენს გრაფის სახით. ამოცანის ამონახსნის ძიებას კომპილატორი აწარმოებს შესაბამის გრაფში. ყველა ზემოთ ჩამოთვლილი ცნებისა და მექანიზმის არსი მოცემულია [7] ნაშრომში. აქვეა მოყვანილი თეორიული მასალის შესატყვისი მაგალითები და პროლოგ-პროგრამები.

თავისი არსით „Strawberry Prolog“-ი ინტერაქტიული იმიტატორია. ხი-ს სისტემების დამპროექტებლები ცდილობენ ინტერფეისი (კავშირი) „მომხმარებელი-კომპიუტერი“ იყოს ჩაქსიმალურად „კეთილმოსურნე“ მომხმარებლისთვის. პროლოგ-პროგრამაში

ჩამოყალიბებულ მისწრაფებებზე (კითხვებზე) კომპიუტერის მიერ პასუხი გაიცემა დიალოგურ რეჟიმში.

თუ პროგრამისტის მიერ ფორმირებულ მიზანს განვიხილავთ, როგორც დასამტკიცებელ თეორემას (მათემატიკურ ლოგიკაზე დაყრდნობით), მაშინ იმიტატორი ცდილობს დაამტკიცოს ეს თეორემა პროგრამაში არსებული აქსიომების (ფაქტებისა და წესების) ბაზაზე.

ხი-ის სისტემების დაპროექტების საწყის ეტაპზე, „ცოდნის ინჟინერი“ აანალიზებს რა გადასაწყვეტ პრობლემას, ახდენს მის ფორმალიზებას ხელოენური ინტელექტის ე.წ. „წარმოდგენის“ რომელიმე ხელოენური ენის ბაზაზე. ხი-ის სისტემების „წარმოდგენის“ ენების თეორიული და პრაქტიკული ასპექტები მოყვანილია მოცემული ნაშრომის ბოლოს წარმოდგენილ ლიტერატურაში. ძირითადად, ხი-ის სისტემების დაპროექტებისას, გამოიყენება: პრედიკატების აღრიცხვა, გრაფების თეორია, სემანტიკური ქსელები, გადაწყვეტილების მიღების თეორია, არაცხადი (ბუნდოვანი) სიმრავლეების თეორია და ა.შ.

მას შემდეგ, რაც „ცოდნის ინჟინერი“ აღწერს გადასაწყვეტ პრობლემას „წარმოდგენის“ რომელიმე ენაზე, იგი დარგის ექსპერტთან კონსულტაციების საფუძველზე ირჩევს პრობლემის რეალიზაციის „პროგრამულ ინსტრუმენტს“. წარმოდგენილ ნაშრომში ჩვენს მიერ გამოყენებულია ლოგიკური პროგრამირების ალგორითმული ენის „Prolog“-ის უახლესი ვერსია „Strawberry Prolog“-ი.

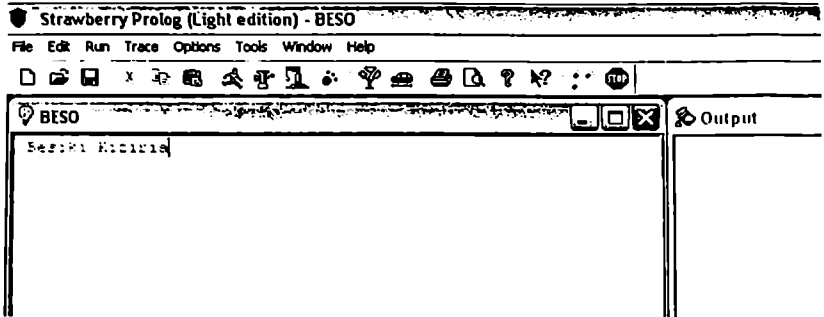
მოვიყვანოთ აღნიშნული იმიტატორის აღწერა.

1.1. „Strawberry Prolog“ იმიტატორის სამუშაო ბარემოს აღწერა

ნახ.1.1. მოცემულია „Strawberry Prolog“ იმიტატორის Desktop (სამუშაო მაგიდა)-არე, რომელსაც უკავია მთელი ეკრანი. Windows XP-ს ზოგადი კონფიგურაცია მდგომარეობს მაქსიმალურ ფორმალიზმსა და უნიფიკაციაში, იმისათვის რომ ახალი პროგრამების გარჩევისას არ იყოს საჭირო პროგრამაში მუშაობის ტექნიკისა და მართვის ხერხების ხელახლა ათვისება. ასეთი მიდგომა უადვილებს დამწყებ მომხმარებელს (ჩვენს შემთხვევაში სტუდენტს, მაგისტრს, ლაბორატორიული მეცადინეობის მასწავლებელს) მუშაობა აწარმოოს Windows სტანდარტულ გარემოში მსგავსი წესებით. რასაკვირველია, ისევე როგორც Windows გარსში მომუშავე ყველა პროგრამას: Microsoft Word, Microsoft Excel, Microsoft Access და სხვა, აქვს მისთვის დამახასიათებელი ნიშანთვისებები, ასევე Strawberry Prolog-ც არ არის გამონაკლისი. ჩვენც, ძირითადად სწორედ ამ ნიუანსებზე გაემახვილებთ თქვენს ყურადღებას.

პროლოგის ნახ.1.1-ზე წარმოდგენილი სამუშაო მაგიდის (ფანჯრის) ზედა პირველ სტრიქონს ეწოდება პროგრამის სათაურის სტრიქონი. იგი იწყება სლოგანით მარწყვის გამოსახულებით. შემდეგ მოდის პროგრამის სათაური: Strawberry Prolog(Light edition) და

პროგრამის სახელი: beso. Light edition ((მსუბუქი გამოცემა(ვერსია)) მიუთითებს იმ გარემოებაზე, რომ იმიტატორის ამ ვერსიის გარდა ასევე არსებობს იმიტატორის პროფესიული ვერსიაც.



ნახ. 1.1. პროლოგ-პროგრამის სამუშაო ფანჯარა

პროფესიული ვერსიის შეძენა შესაძლებელია ინტერნეტის მეშვეობით. ამისათვის საჭიროა შევიდეთ საიტზე www.dobrev.com. ჩვენ ძირითადად აღწერთ ე.წ. მსუბუქ (სტუდენტურ) ვერსიას, რომელიც ავტორთა აზრით სრულიათ საკმარისია მოცემული კურსისათვის „ხელოვნური ინტელექტის საფუძვლები“.

„Strawberry Prolog“ იმიტატორის ფანჯრის შემდგომი მომდევნო ზოლია მენიუს ზოლი. იგი ძირითადად შეიცავს Windows-ის სტანდარტული მენიუს კომპონენტებს: File (ფაილი), Edit (შესწორება), Run (სირბილი, ჩვენს შემთხვევაში ამოცანის შესრულებაზე გაშვება).

Trace (ნაკეალევი). მისი დანიშნულებაა ამოცანის გაშვება ე.წ. ტრასირების (ბიჯობრივი) რეჟიმში. მენიუს ამ კომპონენტს ჩვენ ქვემოთ უფრო დაწვრილებით შევხვებით.

Options (არჩევანი). მისი დახმარებით შესაძლებელია ფანჯრის სხვადასხვა კომპონენტების Settings (გაწყობა). მაგალითად შრიფტის შეცვლა, შრიფტის ზომისა და ფერის შეცვლა და ა. შ.

Tools (საშუალებები, ინსტრუმენტები). მენიუს ამ კომპონენტის დაწვრილებითი აღწერა მოცემულია პროლოგის ფანჯრის მესამე ზოლში, სპეციალური „იარაღიების“ (Shortcut) საშუალებით.

Window ასევე მენიუს სტანდარტული კომპონენტია. Windows ფანჯრებს ნიშნავს. როგორც ცნობილია, შესაძლებელია რამდენიმე ფანჯრის ერთდროული გახსნა და მათთან მუშაობა. ამასთან როგორც Windows-ში, ასევე „Strawberry Prolog“-ში შესაძლებელია ფანჯრები ეკრანზე განათავსოთ სხვადასხვა გზით: Tile Horizontally (ჰორიზონტალურად), Tile Vertically (ვერტიკალურად), Cascade (კასკადურად). ნახ.1.1-ზე პროლოგის ფანჯრები

განლაგებულია პორისონტალურად. ამ ფანჯრებში მომხმარებელს ეძლევა შესაძლებლობა ერთდროულად წაწეროს, როგორც პროლოგ-პროგრამის ტექსტი (მარცხენა ფანჯარა), ასევე მიიღოს ამოცანის ამოხსნის შედეგი (მარჯვენა-Output) ფანჯარაში. ნაშრომის ავტორთა რჩევაა, ძირითადად ვიმუშაოთ ამ რეჟიმში. ეს მომხმარებელს მისცემს საშუალებას ერთდროულად ოველი მიადგენოს, როგორც პროლოგ-პროგრამის ტექსტს, ასევე ამოცანის ამოხსნის შედეგებს. ასევე მარჯვენა Output ფანჯარაში იმიტატორს გამოაქვს სინტაქსური შეცდომები და სხვა სახის საჭირო ინფორმაცია.

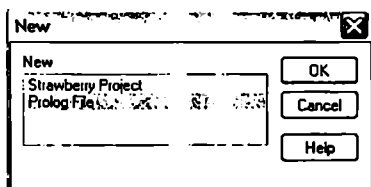
Help (დახმარება)-ც მენიუს სტანდარტული კომპონენტია. მისი დახმარებით პროგრამაში მომუშავე მომხმარებელს ეძლევა საშუალება გამოიძახოს იმიტატორის საცნობარო სისტემა და პასუხი მიიღოს მისთვის საჭირო მრავალი სახის შეკითხვაზე.

იმიტატორის ფანჯრის მესამე ზოლი ეს ინსტრუმენტების ზოლია. იგი იძლევა საშუალებას ოპერატიულად გამოვიძახოთ და შევასრულოთ მოცემული მომენტისათვის საჭირო რაიმე სახის ოპერაცია. ამ ზოლზე მოთავსებულია, როგორც Windows-თვის დამახასიათებელი სტანდარტული იარაღები, ასევე სპეციალურად „Strawberry Prolog“ იმიტატორისათვის შემოღებული იარაღებიც. ნახ.12. მოცემულია ინსტრუმენტების ზოლი.



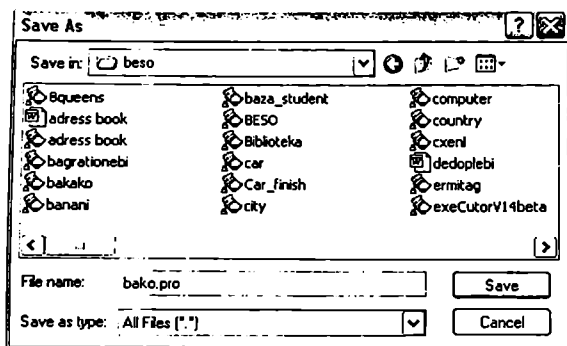
ნახ.12. ინსტრუმენტების სტრიქონი

იარაღიუე „ახალი ფურცელი“  თავის ერთჯერადი დაწკაპუნებით, გამოდის ფანჯარა, რომლითაც მომხმარებელი ხსნის ახალ გვერდს. ნახ.13. მოცემულია ეს ფანჯარა. ამის შემდეგ მოენიშნავთ წარწერას „File“ და მიმითებლით ვაწკაპუნებთ ღილაკზე „OK“. ეკრანზე გამოდის ახალი სუფთა გვერდი. შემდეგ ბოჯზე ვხუროთ პროლოგ-პროგრამის სატექსტო მარცხენა ფანჯარას, ვაწკაპუნებთ ინსტრუმენტების ზოლის „Window“ ღილაკზე და მიმითებლით ვაწკაპუნებთ „Tile“. სუფთა გვერდი ავტომატურად ჯდება ფანჯრის მარცხენა მხარეს. ფანჯარა მზადაა პროგრამის ახალი ტექსტის მისაღებად. ამით პროგრამის შეტანის მოსამზადებელი ეტაპი დასრულებულია.



ნახ.13. ახალი ფაილის გამოშვენი ფანჯარა

შემდგომ ეტაპზე იწყება მომხმარებლის მიერ სამუშაო მაგიდის მარცხენა ფანჯარაში ტექსტის შეტანა. ტექსტის შეტანისას სასურველია პერიოდულად მოეახდინოთ მისი Save (შენახვა) ეინენსტერის შესაბამისი საქაღალდის, მომხმარებლის მიერ მითითებულ ადგილზე. შესაბამის ადგილის თავდაპირველი მითითება წარმოებს ინსტრუმენტების პანელის „დისკეტის“ აღმნიშვნელ იარაღიზე  მიმითებლის ერთჯერ დაწკაპუნებით. სამუშაო მაგიდაზე გამოდის შესაბამისი ფანჯარა, რომელიც მოცემულია ნახ.1.4.



ნახ. 1.4. პროლოგ-პროგრამის ტექსტის შენახვის ფანჯარა

მომხმარებელმა „File name“ ელში უნდა ჩაწეროს მისთვის სასურველი ახალი ფაილის სახელი „pro“ გაფართოებით და მიმითებლით დააწკაპუნოს „Save“ ლილაკზე. ამივე ფანჯარაში არის „Save as type“ (შეინახე როგორც ტიპი) ეელი. მასში მითითებულია პროლოგის სხვა ტიპის გაფართოებები: spj, txt, rtf. პროლოგ-პროგრამის ეს გაფართოებები გამოიყენება სხვადასხვა დანიშნულებით. ფაილის სახელის მითითების შემდეგ იგი თავსდება აღნიშნულ ფანჯარაში პროლოგ ფაილის სპეციალური იარაღიყთ  beso.

ამოცანის ტექსტის ფანჯარაში შეტანისა და მისი შენახვის შემდეგ სასურველია ამოცანა გაშეგებული იქნეს ე.წ. კომპილაციის (Compile) რეჟიმში. ამისათვის მიმითებელს ეაწკაპუნებთ ინსტრუმენტების სტრიქონის შესაბამის  ლილაკზე. პროლოგის იმიტატორი დაამუშავეს რა პროგრამის ტექსტს, აფიქსირებს სინტაქსურ შეცდომებს და გამოიყუეს შესაბამისი შეტყობინებები ფანჯრის მარცხენა „Output“ ეელში. მომხმარებელი გაანალიზებს სინტაქსურ შეცდომებს (Syntactic Error) და ახდენს მათ გასწორებას. შემდეგ

ბიჯზე მომხმარებელი კელავ გაუშვებს ამოცანას კომპილაციის რეჟიმში. ეს პროცესი მრავალჯერადი პროცესია და დამოკიდებულია პროგრამისტის პროფესიონალიზმზე, ამოცანის სირთულეზე და სხვა მრავალ ფაქტორზე. ამ პროცესის დასრულების შემდეგ შესაძლებელია ამოცანა გაშვებულ იქნეს ე.წ. შესრულების (Run) რეჟიმში. ამისთვის მომხმარებელი მიმთითებლით აწკაპუნებს ინსტრუმენტების სტრიქონის  ღილაკზე.

გარდა ე.წ. სინტაქსური შეცდომებისა არსებობს ასევე სემანტიკური (შინაარსობრივი) ტიპის შეცდომებიც. მათ პროლოგის იმიტატორი ვერ აღმოაჩენს, რადგანაც ეს თვით პროგრამისტის მიერ დაშვებული აზრობრივი შეცდომებია. ასეთი შემთხვევისათვის „Strawberry Prolog“ იმიტატორს, ისევე როგორც მაღალი დონის თითქმის ყველა ალგორითმულ ენას, გააჩნია პროგრამის შესრულებაზე გაშვების ე.წ. ბიჯობრივი რეჟიმი. მისი რეალიზაციისათვის პროლოგის იმიტატორს ინსტრუმენტების სტრიქონზე აქვს სპეციალური ღილაკი „Proof Tree“ (დამამტკიცებელი ხე) . მის მოქმედებას კონკრეტული მაგალითის ბაზაზე შემდეგ ქვეთავეში განვიხილავთ.

1.3. „Strawberry Prolog“ იმიტატორის ბიჯობრივი რეჟიმი

სემანტიკური შეცდომის აღმოსაფხვრელად მომხმარებელი უნდა მოიქცეს შემდეგნაირად. განვიხილოთ მარტივი ლოგიკური მაგალითი, რომელიც მოყვანილია [7] ნაშრომის მეოთხე თავში. ეს ამოცანა შეეხება ობიექტების რაიმე თვისებით ამოცნობას და ამ ნიშან-თვისებებით მათ დაღაგებას. მომხმარებლისათვის (ჩვენს შემთხვევაში სტუდენტებისათვის) ყოველივე ზემოთ თქმულის შემდეგ, ეს ამოცანა ნაშრომის ავტორთა აზრით, შესაძლებელია გამოყენებული იქნეს, როგორც პირველი ლაბორატორიული სამუშაო „ხელოვნური ინტელექტის საფუძვლებში“. თუმცა ჩვენ არ გამოვიცხავთ სხვა ამოცანების გამოყენებასაც.

როგორც პროლოგის თეორიული ნაწილიდანაა ცნობილი (იხ. ლიტ. [7]), პროლოგ-პროგრამა ძირითადად შედგება სამი ნაწილისგან: ფაქტები, წესები და მიზნები (შეკითხვები პროლოგ-სისტემას). ამოცანის ცოდნის ბაზის (ფაქტები+წესები) ფორმირების შემდეგ, მომხმარებელი გაანალიზებს რა, „რა“ სურს მიიღოს ამოცანის შედეგის სახით (მოცემული ც/ბ-ის ბაზაზე), აყალიბებს Goal (მიზნებს). როგორც ზემოთ აღვნიშნეთ, პროლოგი ეს ინტერაქტიული (შეკითხვა-პასუხი) ალგორითმული ენაა. აქედან გამომდინარე, პროლოგ-სისტემასთან მომხმარებელმა უნდა აწარმოოს დიალოგი, ანუ სისტემას შეკითხვები უნდა დაუსვას გარკვეული რიგითობით. ეს შეკითხვები მომხმარებელს წინასწარ უნდა აქონდეს ფორმირებული და ამოცანის ტექსტში ჩართული ე.წ. კომენტარის (სტრიქონული ან აბზაცური) სახით.

მოვიყვანოთ ზემოთ მოხსენიებული პროლოგ-პროგრამის მუშა ეარიანტის ამონაბეჭდი.

%*****logikuri programa*****

%*****G O A L!!!*****

?-dark(W),big(W),write("MUKI DA DIDI RA ARIS?"),

nl,nl,write(W),nl,nl.

/*?-write(" RA ARIS DIDI?"),nl,nl,
write("SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS"),nl,nl,
big(Didi),write(Didi),nl,nl,fail.*/

/*?-write(" RA ARIS PATARA?"),nl,
write("*****"),nl,nl,
small(Patara),write(Patara),nl,nl,fail.*/

%*****PAQTEBI:*****

big(bear).

big(elephant).

small(cat). % cat-kata;

brawn(bear). % brawn-kavisperi;

black(cat). % black-shavi;

grey(elephant). % grey-nacrisperi;

%***** CESEBI*****

dark(R) :- black(R).

dark(R) :- brawn(R).

%*****programis dasasruli.*****

როგორც პროგრამის ამონაბეჭდიდან ჩანს, ამოცანაში მოცემულია სამი მიზანი (შეითხება), რომლებიც მოთაგებულება კომენტარის სახით. როგორც ზოგადად ჟველა აღგორითმულ ენაში, ასევე პროლოგის იმიტატორი არ განიხილავს კომენტარს პროგრამის ობიექტად. იმისათვის რომ, კომენტარი ჩაჯდეს პროგრამის ტანში, მომხმარებელმა მას უნდა მოუხსნას სტრიქონული ან აბზაცური კომენტარის აღმნიშვნელი სპეციალური სიმბოლოები (ჩვენს შემთხვევაში: % -სტრიქონული ან /* */ აბზაცური). ასე მაგალითად: მოუხსნათ აბზაცური კომენტარის სიმბოლოები მეორე მიზანს. მიზანი იმიტატორის მიერ აღიქმება, როგორც პროგრამის ობიექტი. ამის შემდეგ მომხმარებელი უშვებს ამოცანას

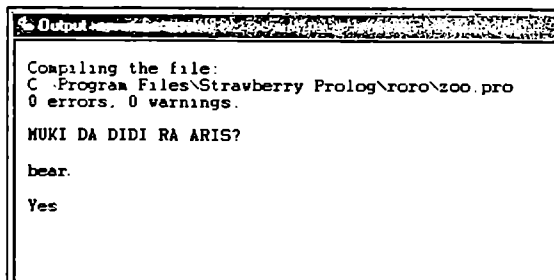
შესრულებაზე, ანუ უსვამს პროლოგ-სისტემას შეკითხვას (უსახავს მისანს). მოცემულ შეკითხვაზე პროლოგ-სისტემის ადექვატური პასუხი მოცემულია ნახ. 15.

პუნქტს „Tile Windows Horizontally“. იმიტატორის ფანჯარაში ერთდროულად გამოდის სამივე ფანჯარა: პროლოგის იმიტატორის ფანჯარა (StrProlog is TRACING“)

■ **StrProlog is TRACING.** ამოსახსნელი პროგრამის ტექსტით, ტრასირების რეჟიმის ფანჯარა

■ **Proof Tree - step 1.** და შედეგების გამოტანის ფანჯარა **Output**. მომხმარებელს ეძლევა საშუალება საჭიროებისამებრ გააქტიუროს მოცემულ მომენტში მისთვის საჭირო ფანჯარა. „Step 1“ აღნიშნავს იმ ფაქტს, რომ იმიტატორმა გაიარა ამოცანის ამოხსნის პირველი ბიჯი-კომპილაცია.

4. მოცემულ ბიჯზე მომხმარებელი იწვევს ამოცანის ამოხსნას ბიჯობრივ რეჟიმში. წინა სამი ეტაპი მეოთხე ბიჯის მოსამზადებელ ეტაპებად შეიძლება ჩაითვალოს. ამ ეტაპების დაწერილობითი აღწერა მომხმარებელს დაეხმარება ნებისმიერი ამოცანის გადაწყვეტილის წარმოშობილი სემანტიკური დაბრკოლების გადალახვაში. მომხმარებელი უნდა შევიდეს იმიტატორის ფანჯრის მარცხნივ მოთავსებული **StrProlog is TRACING.** ქვეფანჯრის მენიუს სტრიქონის „Trace“ კომპონენტში და მიმთითებლით დააწკაპუნოს მისი ქვემენიუს პუნქტზე „One Step“. ამის შემდეგ კლავიატურაზე მოთავსებული ე.წ. ფუნქციური კლავიშის „F9“ საშუალებით შევასრულოთ ამოცანის ამოხსნა ბიჯობრივ რეჟიმში. განვიხილოთ ამოცანის შედარებით „რთული“ მიზანი (შეკითხვა): „ერთდროულად მუქი და ღიდი ობიექტი რომელია?“ მივიღებთ შედეგს, რომელიც ნახ. 17. ასახული.



ნახ. 17. შეკითხვაზე იმიტატორის პასუხის ამონაბეჭდი

■ **Proof Tree - step 1.**

„დამამტკიცებელი ხის“ დამატებით ქვეფანჯარაში ქვემიზნები გამოდის სხვადასხვა შეფერილობით. მოცემულ ბიჯზე უკვე მიღწეული ქვემიზნები გამოდის ღურჯი ან მწვანე ფერით. ნაწილობრივ მიღწეული ქვემიზნები წითელი ფერისაა, ხოლო მიუღწეადი ქვემიზნები შავი ფერის.

ამრიგად, ჩვენ შეძლებისამებრ აღეწერეთ Strawberry Prolog იმიტატორის რაობა, მასში მუშაობის რეჟიმები და ნიუანსები. შემდგომ თავებში ჩვენ შევეხებით ხი-ის სისტემების დაპროექტების სხვადასხვა მაგალითებს. ამ განხილვებში ასევე იქნება მოყვანილი იმიტატორის სხვადასხვა შესაძლებლობებიც.

თავი 2. რელაციური ცოდნის ბაზების პროექტირება. მათი გამოყენება ხელოვნური ინტელექტის ამოცანებში

ინტელექტუალური სისტემები, ადამიანი-ექსპერტის ცოდნაზე დაფუძნებული სისტემებია. ლოგიკური პროგრამირების ალგორითმული ენა „პროლოგი“ ძირითადად იყენებს ე.წ. რელაციური ცოდნის ბაზას. ამ ტიპის ცოდნის ბაზები (ც/ბ) პროლოგ-პროგრამის ტანში „ჩაშენებულ“ ცოდნის ბაზებია. ეს არ ნიშნავს იმას, რომ ხელოვნური ინტელექტის (ხი) სისტემების პროექტირებისას არ შეიძლება სხვა სახის ბაზების მართვის სისტემების გამოყენება. ხი-ის ამოცანების გადაწყვეტისას, ასევე შესაძლებელია ვისარგებლოთ ბაზების მართვის სისტემებით: „Microsoft Access“, „Foxpro“, „dbase“, „ორაკლი“ და სხვა. ჩვენ პროგრამულ ინსტრუმენტად ვიყენებთ „Strawberry Prolog“-ის კომპილატორს. ც/ბ-ის დაპროექტების თეორიული ასპექტები მოყვანილია [7] ნაშრომში.

როგორც არაერთჯერ აღნიშნეთ, ც/ბ შედგება ორი კომპონენტისგან: ფაქტებისა (fact) და წესებისგან (rule). ცოდნის ბაზის ფორმირების შემდეგ პროგრამისტი აყალიბებს ამოცანის მიზნებს (goal). „Strawberry Prolog“-ი ინტერაქტიული კომპილატორია, მომხმარებელი პროლოგ-პროგრამას დაუსახავს გადასაწყვეტ პრობლემას, შეკითხვის სახით. პროლოგ-სისტემა (პ/ს) დასახულ მიზანს აღიქვამს, როგორც დასამტკიცებელ თეორემას. იგი ცდილობს დაამტკიცოს ეს თეორემა პროგრამაში არსებული აქსიომების ბაზაზე. აქსიომები ეს „ჩაშენებულ“ ც/ბ-ში მოთავსებული ფაქტები და წესებია. პრობლემის გადაწყვეტის შემთხვევაში, შედეგების  ელში გამოდის დადასტურების შეტყობინება „Yes“. ვერ გადაწყვეტის შემთხვევაში კი „no“ შეტყობინება. სხვა მრავალ ფაქტორს, რომლებიც გამოიყენება რელაციური ცოდნის ბაზების ფორმირებისას, უშუალოდ მაგალითების განხილვისას შევხებით.

რელაციური ცოდნის ბაზების (ც/ბ) დაპროექტების პირველ მაგალითად ავიღოთ საქართველოსათვის ისტორიულად ტრადიციული დარგი მელვინეობა, კონკრეტულად კი ქართული ღვინოების კატალოგი (ცნობარი).

ამოცანა 1. ქართული მარანი.

მოვიყვანოთ ამოცანის ზოგადი აღწერა. ქართული ღვინის სახეობები აღწერით შემდეგი ძირითადი კომპონენტებით:

- ღვინის დასახელება;
- ღვინის პირველადი ჩამოსხმის წელი;
- ღვინომასალა, ანუ ენახის ჯიში (ან ჯიშები), რისგანაც მსადადება აღნიშნული ღვინო;
- ღვინის დამზადების რეგიონი;

- აღკოპოლის შემცველობა;
- ღვინის მიერ სხედასხვა საკაეშირო თუ საერთაშორისო გამოფენებზე მიღებული ოქროს მედლების რაოდენობა
- ეერცხლის მედლების რაოდენობა;
- ბრინჯაოს მედლების რაოდენობა;
- ღვინის შაქრიანობა;
- ღვინის სახეობა.

ღვინის სახეობის დამახასიათებელი კომპონენტების რაოდენობა რასაკვირველია შესაძლებელი იყო უფრო გაზრდილიყო, მაგრამ ეს ჩვენს მიზნებში არ შედიოდა. ამასთან, აღსანიშნავია ის გარემოება, რომ „პროლოგი“-ის კომპილატორი რელაციურ ც/ბ-ში ფაქტებისა და მათში შემავალი კომპონენტების რაოდენობას ამ მხრივ არავითარ შეზღუდვას არ უწყებს.

საქართველოში ძირითადად მზადდება შემდეგი სახეობის ღვინოები:

- სამარკო ღვინო;
- ორდინარული (ჩვეულებრივი) ღვინო;
- ნახევრადმშრალი ღვინო;
- ნატურალური ნახევრადტკბილი ღვინო;
- დასპირტული (პორტვინი) ღვინო;
- სადესერტო ღვინო;
- შუშხუნა ღვინო.

ღვინოები თავისი „წოდებების“ მიხედვით შესაძლებელია დაიყოს შემდეგ ჯგუფებად:

- ახალი ღვინო;
- შედარებით ახალი ღვინო;
- ძველი ღვინო.

სპირტიანობისა და შაქრიანობის მიხედვით ღვინოები იყოფა:

- სუსტი ღვინო;
- საშუალო სიძლიერის ღვინო;
- ძლიერი ღვინო.

ყველა ზემოთ ჩამოთვლილი ღვინის მახასიათებელი ასახულია ცოდნის ბაზაში ფაქტის ან წესის საშუალებით.

საქმოდ სოლიდური (დიდი ინფორმაციის შემცველი), ცოდნის ბაზის ფორმირების შემდეგ, პროგრამისტი ითვალისწინებს რა ამოცანის დანიშნულებასა და პროგრამის პოტენციური მომხმარებლის მოთხოვნებს, აყალიბებს მიზნების (შეკითხვების) სიმრავლეს. პროლოგი ლოგიკური პროგრამირების დეკლარაციული ტიპის. არარეცხობრივი, მიზნობრივი, „რა“ ტიპის ალგორითმული ენაა. მისი ზემოთ ჩამოთვლილი ყველა თვისება

დაწერილებით არის აღწერილი ლიტერატურაში მოყვანილ წიგნებში, მათ შორის [7] ნაშრომში.

ამ პროგრამის სრული მუშა ტექსტი მოყვანილია დანართი 1-ში. დაინტერესებულ მკითხველს აქვს შესაძლებლობა „Strawberry Prolog“ იმიტატორის ფილოსოფიის შემთხვევაში გაუშვას ამოცანა სრულად და მიიღოს პასუხი მისთვის საინტერესო ყველა შეკითხვაზე. ჩვენ კი ნაშრომის მოცულობის შესაზღუდულობის გამო მოვიყვანთ ამოცანის ზოგიერთ საინტერესო მომენტს.

ამოცანას შესაძლებელია დაეუსახოთ სხვადასხვა სახის მიზნები (დაეუსვათ შეკითხვები), რომლებზეც მივიღებთ ადექატურ პასუხს. ასე მაგალითად, შეკითხვაზე:

როდის არის პირველად დამზადებული ც/ბ-ში მოთავსებული ღვინოები? **Output**
ველში ვლგებულობთ პასუხს, რომელიც ნახ. 12.-ზეა ასახული. შეკითხვა პროლოგზე ასე ჩაიწერება იხ. ნახ. 2.1 :

```
?- write( " MANUFACTURED YEAR: " ),  
   nl, nl, write( " (pirveladi gamoshvebis celi) " ),  
   nl, nl, wine( Name, Year, _ _ _ _ _ ), write( Name ), write( " " ),  
   write( Year ), nl, nl, fail.
```

ნახ. 2.1.

მიზანში (შეკითხვაში) გამოყენებულია პროლოგის ორი მნიშვნელოვანი კომპონენტი, რომელსაც აუცილებლად უნდა მიეძღვას ყურადღება ამოცანის კომპიუტერზე ამოხსნისას.

პროლოგის კომპილატორში „ჩაშენებულია“ პრედიკატი „fail“ (ჩაეარდნა). მისი არსი შემდეგში მდგომარეობს: „მიზანი, რომელიც ჩვენს მიერ იყო დასახული დამთავრდა ჩაეარდნით (კრახით)“. იგი პროლოგში თამაშობს „ცრუ“ მიზნის როლს და მისი შესრულება ყოველთვის მთავრდება კრახით. ბუნებრივია ლოგიკური კითხვა: რა როლს თამაშობს ეს პრედიკატი ამოცანის მოცემულ ფრაგმენტში?

როგორც პროლოგის თეორიული კურსიდან (ნაშრომი [7]) ვიცით, ამ ალგორითმულ ენაში (განსხვავებით „იმპერატიული“ ალ. ენებისა) გვაქვს სამი სახის ცვლადი:

– ჩვეულებრივი ცვლადი, რომლის ჩაწერაც აუცილებელია დაიწყოს დიდი ასოთი ნიშნით. მისი მოქმედების არეალი მხოლოდ ერთი წინადადებაა.

– გლობალური ცვლადი, რომლის მოქმედების არეალი მთელი პროგრამაა. იგი აუცილებლად უნდა დაიწყოს „G“ სიმბოლოთი, რაც სიტყვა „Global“-ის პირველი ასოა და

– ფიქტიური ცვლადი, რომელიც პროლოგში დეფინის ნიშნით „-“ ჩაიწერება. მოცემულ შეკითხვაში (ნახ. 2.1.) ჩვენს მიერ შემოტანილ ღვინის აღმნიშვნელ პრედიკატში „wine“ გამოყენებული გვაქვს ფიქტიური ცვლადი, ღვინო, ანუ მისი შესაბამისი პრედიკატი „wine“ ც/ბ-ში, როგორც ზევით ენახეთ, აღწერილია სულ 10 პარამეტრით. მოცემული შეკითხვის არსიდან გამომდინარე, მომხმარებელს მოცემულ მომენტში აიტერესებს მხოლოდ ც/ბ-ში მოთავსებული ღვინოების პირველადი ჩამოსხმის წელი. ამის გამო „wine“ პრედიკატში ცვლადების სახით ჩაწერილია ღვინის დასახელება და მისი

პირველადი ჩამოსხმის წელი. დანარჩენი კომპონენტები მოცემულ შეკითხვაში ჩანაცვლებულია ფიქტიური ცვლადით.

პროლოგის იმიტატორში ამ შემთხვევაში არ გამოთქმება მასივის ცნება. დაიწყებს რა ც/ბ-ში შეკითხვის შესატყვისი შედეგების ძებნას, ჰს პირველივე პასუხის მოძებნისას (როგორც ზემოთ აღვნიშნეთ, ძებნა ც/ბ-ში წარმოებს ზევიდან ქვევით და მარცხნიდან მარჯვნივ) ცვლადს „Name“-ს შეუთავსებს ც/ბ-ში მოთავსებული პირველივე ფაქტის „wine“-ის პირველ კომპონენტს: წინანდალს, ხოლო ცვლადს „Year“-ს ღვინო წინანდალის პირველადი ჩამოსხმის თარიღს: 1886 წ. ამის შემდეგ ჰს გადადის მეორე ფაქტზე და სწორედ აქ წარმოიქმნება პრობლემა. იმის გამო რომ ცვლადებს: Name და Year-ს უკვე შეთავსებული აქვთ წინანდალის მონაცემები და ეს ცვლადები არ არის აღწერილი, როგორც მასივები, ახალი ფაქტის შესაბამისი ახალი მონაცემები ვერსად ვერ თავსდება. სწორედ აქ ამუშავდება „fail“ მექანიზმი. იგი ჯერ ერთი ახდენს ე.წ. „ჩაეარდნას“, გამოაქვს

რა წინანდალის ღვინის შესაბამისი მონაცემები **გა-Output** ველში, შემდეგ აუქმებს ცვლადებისადმი: Name და Year შეთავსებული ღვინო წინანდალის შესაბამისი მონაცემებს და ადგილს უნთავისუფლებს მომდევნო ფაქტის შესაბამის ინფორმაციას. ჩვენს შემთხვევაში ეს არის ღვინო გურჯაანი. გარდა ამისა „fail“ პრედიკატი ანხორციელებს ც/ბ-ის გადასინჯვას ზემოთ მოცემული მიმდევრობით. და ა.შ. მოცემულ პირველ შეკითხვაზე პასუხი ც/ბ-ის ბოლომდე გავლის შემდეგ იქნება:

MANUFACTURED YEAR:

(pirveladi gamoshvebis celi)

tsinandali	1886
gurdjaani	1887
napareuli	1882
manavi	1938
cicka	1890
colikauri	1890
rkaciteli	1892
tibaani	1948
sviri	1890
teliani	1897
napareuli	1890
mukuzani	1888
ereti	1977
garedji	1977
gelati	1977
kaxeti	1948
dimi	1977
shuamta	1984
saperavi	1886
anakopia	1978
aguna	1948

და ა.შ.

ნახ. 2.2.

პროლოგ-პროგრამას დაეუძახოთ შემდეგი მიზანი: ამოვიღოთ ც/ბ-დან ოქროს მედალოსანი ლეინოები.

```
?- write( " OCROS MEDALOSANI GVINOEBIA. " ),
nl, write( "*****" ),
nl, nl, wine( Name, _, _, Gold, _,
_, _, _ ), Gold >= 1, write( Name ), write( " dajildovebulia " ),
write( Gold ), write( " medlit." ), nl, nl, fail.
```

ამ მიზანს პ/ს პასუხობს  Output ელში შემდეგნაირად იხ. ნახ. 23.:

OCROS MEDALOSANI GVINOEBIA.

tsinandali dajildovebulia 10 medlit.

gurdjaani dajildovebulia 1 medlit.

napareuli dajildovebulia 1 medlit.

rkaciteli dajildovebulia 1 medlit.

tibaani dajildovebulia 2 medlit.

sviri dajildovebulia 1 medlit.

teliani dajildovebulia 4 medlit.

napareuli dajildovebulia 6 medlit.

mukuzani dajildovebulia 8 medlit.

saperavi dajildovebulia 1 medlit.

axmeta dajildovebulia 1 medlit.

tvishi dajildovebulia 1 medlit.

chxaveri dajildovebulia 1 medlit.

xvanchkara dajildovebulia 2 medlit.

kindzmarauli dajildovebulia 3 medlit.

axasheni dajildovebulia 6 medlit.

ojaleshi dajildovebulia 2 medlit.

usaxelaui dajildovebulia 2 medlit.

kardanaxi dajildovebulia 8 medlit.

iveria dajildovebulia 1 medlit.

xixvi dajildovebulia 4 medlit.

saamo dajildovebulia 4 medlit.

salxino dajildovebulia 6 medlit.

No.

ნახ. 2.3.

ქასუხის ბოლოს „No“ ნიშნავს იმას, რომ ოქროს მედალოსანი ღვინოები ც/ბ-ში მეტი არ არის.

შეკითხვაზე, თუ რომელი ჯიშის ყურძნისგან მზადდება ესა თუ ის ღვინო, მიზანი პროლოგზე ასე ჩაიწერება. იხ. ნახ. 2.4.

```
?- write("  M A D E F R O M:  "),
nl,nl, write("  ( D A M Z A D E B U L I A)  "),
nl,nl,nl,wine(Name,_Made_from,
_ _ _ _ _),
write(Name),write("  "),
write(Made_from),nl,nl,fail.
```

ნახ. 2.4.

შედეგი გამოდის  **Output** ეელში. იხ. ნახ. 2.5.

M A D E F R O M:

(D A M Z A D E B U L I A)

tsinandali rkatsiteli_and_mcvane

gurdjaani rkatsiteli_and_mcvane

napareuli rkatsiteli

manavi mcvane

cicka cicka

colikauri grape_variety

rkaciteli rkaciteli

tibaani rkaciteli

sviri colikauri_and_eicka_and_kraxuna
teliani kaberne
napareuli saperavi
mukuzani saperavi და ა.შ.

ნახ. 25.

თუ პ/ს-ს დავეუსახეთ ისეთ მიზანს, რომლის პროლოგ-ტექსტშიც არ იქნება პრედიკატი „fail“, მაშინ პ/ს მონახავს ც/ბ-ში შეკითხვაზე პირველივე შესაბამის პასუხს და არ გადავა კითხვის შესაბამისი სხვა შესაძლებელი პასუხების მოძებნაზე (ე.ი. არ განაგრძობს შედეგების ძებნას ც/ბ-ში). შედეგების  ველში გამოვა შეტყობინება „Yes“. ასე მაგალითად:

```
?-write(" DAMZADEBULIA KAXETSHI. "),
nl,write("*****"),
nl,nl,wine(Name,_kaxeti,_
_),write(Name),nl.
```

შეკითხვაში „fail“ პრედიკატი არ არის. შეკითხვაზე პასუხი გამოდის 
ქვეყანაჯარაში

DAMZADEBULIA KAXETSHI.

tsinandali
Yes.

ეს მაშინ როცა საქართველოში, კახეთი ითვლება ღვინის მწარმოებელ ძირითად რეგიონად. ეს ფაქტი პროლოგ-პროგრამის ხელმეორედ შესრულებაზე გაშვების შემდეგაც კარგად ჩანს. ჩაუმატოთ კითხვაში „fail“ პრედიკატი და ამოცანა ხელმეორედ გაეუშვათ შესრულებაზე. მივიღებთ კახეთის რეგიონში წარმოებულ, ც/ბ-ში მოთაყესებული ღვინოების სრულ ჩამონათვალს:

DAMZADEBULIA KAXETSHI.

tsinandali
gurdjaani
napareuli
manavi
rkaciteli
tibaani
teliani
napareuli

mukuzani
garedji
kaxeti
shuamta
saperavi
pirosmani
axmeta
alaznis_veli
kindzmarauli
axasheni
kardanaxi
anaga
signagi
iveria
taribana
marabda
xixvi
saamo
No.

აქვე აღსანიშნავია ერთი ძირითადი გარემოება. პროლოგ-პროგრამებში, თითქმის ყველა მიზნის (შეკითხვის) ფორმირებისას, მიზანი ყალიბდება, როგორც კონიუნქციური სტრუქტურის წინადადება, ანუ ძირითადი მიზანი შედგება ქვემიზნებისგან და ეს ქვემიზნები ერთმანეთთან გაერთიანებულია ძირითადად კონიუნქციის ნიშნით „...“ (შესაძლებელია ასევე დიზიუნქციის გამოყენებაც).

როგორც არაერთგზის აღინიშნა, ც/ბ-ში ასევე მოთავსებულია წესები (rules). სწორედ წესებში ათავსებს პროგრამისტი კონკრეტული მიზნის ძებნისადმი წაყენებულ სხვადასხვა სახის შეზღუდვებს და ლოგიკურ პირობებს. სისტემის დამპროექტებელი „ცოდნის ინჟინერი“ წესების ამ პირობებში და შეზღუდვებში „ღებს“ თავის „ჟერისტიკებს“.

მოიყვანოთ განხილული ამოცანის ც/ბ-ში მოთავსებული ზოგიერთი წესი და მათში ჩადებული „ჟერისტიკები“. მაგ. ავიღოთ წესი „იქველი ღვინო“. წესი პროლოგზე ასე ჩაეწეროს:

old_wines(Name):-
wine(Name,Year,_,_,_,_,_),
Year=<1900.

პროგრამისტი ამ წესში „ძველ ღვინოდ“ ნათელა ყველა ის ღვინო, რომელიც გამოშვებულია 1900 წლამდე. შესაძლებელი იყო აღნიშნული წესის სხვა სახით ჩაწერაც. პროგრამირება ეს შემოქმედებითი პროცესია და ყველა პროგრამისტი „თავისუფალია“ თავის არსებობაში, რასაკვირველია აქ ნაგულისხმებია, რომ მისი „თავისუფლება“ გარკვეულად მაინც „შეზღუდულია“ უშუალოდ ალგორითმული ენის სინტაქსითა და სემანტიკით. დიდ როლს თამაშობს ასევე თვით პროგრამისტის პროფესიული დონეც.

პ/ს-ს თუ დაუსახავთ მიზანს, მონახოს ც/ბ-ში მოთავსებული ყველა „ძველი ღვინო“:

შედეგების გამოტანის **Output** ქვეფანჯარაში მივიღებთ შემდეგ პასუხს:

tsinandali
gurdjaani
napareuli
cicka
colikauri
rkaciteli
sviri
teliani
napareuli
mukuzani
saperavi
No.

დაინტერესების შემთხვევაში მკითხველმა შესაძლებელია მიმართოს დანართი 1-ს.

ამოცანა 2. გარკვეის ჩამკონტა ლიკა ფაბბურტში.

- კლუბის დასახელება;
- ქვეყნის დასახელება, საიდანაც არის კლუბი;
- კლუბის ღაარსების წელი;
- მოგებული თასების რაოდენობა შესაბამისი წლების ჩვენებით;
- კლუბის ბიუჯეტი;
- მოცემულ ტურნირზე გატანილი გოლების რაოდენობა;

- ტურნირზე გაშვებული გოლების რაოდენობა;
- ტურნირზე ფრედ დასრულებული თამაშები;
- ტურნირზე მოგებული თამაშების რაოდენობა;
- ტურნირზე წაგებული თამაშების რაოდენობა;
- ტურნირზე კლუბის მიერ მოპოვებული ჯამური თანხა;
- რომელ ეტაპზე გამოეთიშა კლუბი ტურნირს.

ტურნირზე ანაღაურება წარმოებს შევიცარული ფრანკით. ამასთან 2000 წელს ეს თანხები ნაწილდებოდა შემდეგი პრინციპით:

- ჯგუფურ ტურნირში მოგებისათვის კლუბი იღებდა ერთ მილიონ შვეიცარულ ფრანკს, ხოლო ფრეში 0.33 მილიონ შვეიცარულ ფრანკს
- პირველი ჯგუფიდან გასვლაში კლუბი იღებდა 2 მილიონ ფრანკს;
- მეორე ჯგუფიდან გასვლაში 4 მილიონ ფრანკს;
- მეოთხედ ფინალში გასვლაში 8 მილიონ ფრანკს;
- ნახევარ ფინალში გასვლაში 15 მილიონ ფრანკს;
- ფინალში გასვლაში კლუბი იღებდა 20 მილიონ შვეიცარულ ფრანკს;
- თასის მოპოვების შემთხვევაში კი დამატებით კიდევ 25 მილიონ შვეიცარულ ფრანკს.

ამოცანის ც/ბ-ის დაპროექტების შემდეგ, მომხმარებელს ეძლევა საშუალება დაუსახოს პროლოგ-სისტემას სხედასხვა სახის მიზნები (დაუსვას კითხვები დიალოგურ რეჟიმში).

პირველი შეკითხვა ასე ჩამოვყალიბებო: ჩამოთვალეთ 1999-2000 წლების ევროპის ჩემპიონთა ლიგაში მონაწილე ყველა კლუბი. პროლოგ-პროგრამაში ეს შეკითხვა ასე ჩაიწერება. იხ. ნახ. 2.6. :

```
?-write(" CHEMPIONTA LIGASHI MONACILE CLUBEBIA: "),nl,  
write("*****"),  
nl,nl,club(Name,_),write(Name),  
nl,fail.
```

Ex. 2.6.

პასუხი ამ შეკითხვაზე გაროდის ინფორმაციის გამოტანის **Output** ქვეფანჯარაში:
იხ. ნახ. 2.7.

CHEMPIONTA LIGASHI MONACILE CLUBEBIA:

arsenal
aik
barcelona
bayern
bordeaux
boavista

valencia
 villem2
 galatasaray
 herta
 dortmund
 dynamo
 croatia
 chelsea
 lazio
 leverkusen
 manchester
 milan
 molde
 maribor
 olimpiakos
 olympik
 porto
 psv
 rangers
 real
 rosenborge
 spartak
 sparta
 sturm
 fiorentina
 feenoord
 No.

ნახ. 2.7.

შემდეგი მიზანი ასე ჩამოყალიბით: ამ ტურნირში მონაწილე კლუბთაგან, რომელ კლუბს აქვს მოგებული ევროპული ტურნირების ისტორიაში ყველაზე მეტჯერ ესა თუ ის თასი. პროლოგზე ეს მიზანი ასე ჩაიწერება:

```
.....
?-write(" XX-SAUKUNESHI KLUBEBIS MIER"),nl,
write(" CHEMPIONTA LIGASHI "),nl,
write(" MOPOVEBULI TASEBIS RAODENOBAA:"),nl,
write("*****"),
nl,nl,club(Name,_,Tasi,Year,_,_),Tasi>0,
write(Name),write(" mopovebuli aqvs "),
write(Tasi),write(" tasi. "),write(Year),write(" cels."),nl,nl,fail.
```

პასუხი შეკითხვაზე გამოვიყენოთ  ელში. იხ. ნახ. 2.8. :

```
.....
XX-SAUKUNESHI KLUBEBIS MIER
CHEMPIONTA LIGASHI
MOPOVEBULI TASEBIS RAODENOBAA:
*****
```

barcelona mopovebuli aqvs 1 tasi. 1991-92 cels.
 bayern mopovebuli aqvs 3 tasi. 1973-74@74-75@75-76 cels.

dortmund mopovebuli aqvs 1 tasi. 1996-97 cels.
 manchester mopovebuli aqvs 2 tasi. 1967-68@98-99 cels.
 milan mopovebuli aqvs 5 tasi. 1962-63@68-69@88-89@89-90@93-94 cels.
 olympik mopovebuli aqvs 1 tasi. 1992-93 cels.
 porto mopovebuli aqvs 1 tasi. 1986-87 cels.
 psv mopovebuli aqvs 1 tasi. 1987-88 cels.
 real mopovebuli aqvs 8 tasi. 1955-56@56-57@57-58@58-59@59-60@65-66@97-98@99-2000 cels.
 feenoord mopovebuli aqvs 1 tasi. 1969-70 cels.

No.

ნახ. 2.8.

პ/ს-ს დაეუსახოთ მიზანი: ამ ტურნირში მონაწილე ყველაზე მდიდარი კლუბები:

```
?-write(" KVELASE MDIDARI KLUBEBIA "),nl,
write("*****"),nl,
club(Name,_,_,Biudjeti,_,_,_,Biudjeti>=240,write(Name),
write(" Biudjeti "),write(Biudjeti),write(" mln."),nl,nl,fail.
```

 **Output**

საინფორმაციო ეელში მივიღებთ შემდეგ პასუხს. იხ. ნახ. 2.9.

KVELASE MDIDARI KLUBEBIA :

manchester Biudjeti 450 mln.

milan Biudjeti 250 mln.

real Biudjeti 320 mln.

No.

ნახ. 2.9.

შემდეგი მიზანი ასე ჩამოვყალიბოთ: რომელი ქვეყნის კლუბები მონაწილეობდნენ ამ ტურნირში.

```
?-write(" TESIS GATAMASHEBASHI MONACILEOBDNEN "),nl,
write(" SHEMDEGI QVECNEBIS KLUBEBI:"),nl,
write("*****"),nl,
club(Name,Country,_,_,_,_,_,_,write(Name),
write("-----"),write(Country),nl,nl,fail.
```

 **Output**

ეელში. იხ. ნახ. 2.10.

TESIS GATAMASHEBASHI MONACILEOBDNEN

SHEMDEGI QVECNEBIS KLUBEBI:

bordeaux-----	1911 cels.
boavista-----	1910 cels.
valencia-----	1903 cels.
villem2-----	1906 cels.
galatasaray-----	1905 cels.
herta-----	1901 cels.
dortmund-----	1901 cels.
dynamo-----	1905 cels.
croatia-----	1909 cels.
chelsea-----	1900 cels.
lazio-----	1900 cels.
leverkusen-----	1911 cels.
manchester-----	1899 cels.
milan-----	1899 cels.
molde-----	1999 cels.
maribor-----	1912 cels.
olimpiakos-----	1889 cels.
olympik-----	1902 cels.
porto-----	1903 cels.
psv-----	1903 cels.
rangers-----	1873 cels.
real-----	1890 cels.
rosenborge-----	1901 cels.
spartak-----	1903 cels.
sparta-----	1912 cels.
sturm-----	1918 cels.
fiorentina-----	1901 cels.
feenoord-----	1904 cels.

ნახ. 2.11.

წესების ფორმირებისას ამოცანაში „ქართული მარანი“, პროგრამისტი თავის „ეერისტიკებს“, ფორმალისებულს სხვადასხვა ლოგიკური პირობების სახით, ათავსებდა უშუალოდ წესის „ტანში“.

მიზნების პროგრამირება ინტელექტუალური სისტემების დაპროექტების ერთერთი მნიშვნელოვანი ეტაპია. შესაძლებელია პროგრამისტის „ეერისტიკები“, არა წესში ჩაიწეროს, არამედ უშუალოდ კონკრეტულ მიზანში ჩაირთოს. მოვიყვანოთ მიზნების ფორმირების ამ ტექნოლოგიის რამოდენიმე ნიმუში განხილული ამოცანის ბაზაზე.

პროგრამისტის ასრით, „ახალგაზრდა“ კლუბებს მან მიაკუთვნა ის კლუბები, რომლებიც ჩამოყალიბდნენ 1911 წლის შემდეგ. ეს პირობა მან უშუალოდ მიზნის ტანში მოათავსა და არ ჩაწერა ის წესის სახით.

```
?- write( "      YOUNG CLUB  "),
nl,write("*****"),nl,nl,
club( Name, _Year, _ _ _ _ _ _ _ _ _ _ , Year >= 1911,
write( Name ),write( "  "), write( Year ), nl, nl, fail.
```

YOUNG CLUB

aik 1914
bordeaux 1911
leverkusen 1911
molde 1999
maribor 1912
sparta 1912
sturm 1918

No.

ნახ. 2.12.

დასახულ ახალ მიზანში: რომელმა კლუბმა მოიპოვა ამ ტურნირზე ყველაზე მცირე თანხა, პროგრამისტმა მიზანშივე მოათავსა თავისი სუბიექტური შეხედულება (ევრისტიკა) ამ საკითხზე პირობის სახით, რომელიც მიზნის ტექსტშია ასახული:

```
?- write( "      LEAST GET MONEY CLUB
"),nl,nl,write("*****"),nl,nl,
club(Name,Money, Money<3.5,
write(Name),write(" "),write(Money),write(" mln."),nl,nl,fail.
```

დასახულ მიზანზე პასუხი შემდეგია. იხ. ნახ. 2.13.

LEAST GET MONEY CLUB

aik 2.33 mln.
villem2 2.66 mln.
galatasaray 3.33 mln.
molde 3 mln.
maribor 3.33 mln.
psv 3.33 mln.
No.

ნახ. 2.13.

აღნიშნული ამოცანის გარშემო, დაინტერესებულ მკითხველს აქვს შესაძლებლობა მიიღოს პასუხი მისთვის საინტერესო მრავალ კითხვაზე, თუ მიმართავს დანართი 1-ში მოცემული პროგრამის სრულ ტექსტს და შეასრულებს ამოცანას კომპიუტერზე.

ამოცანა 3. ავტომანქანების კატალოგი.

ამოცანაში მოყვანილია თანამედროვე ავტომანქანების ცოდნის ბაზა (ც/ბ), რომელშიც ავტომობილები დახასიათებულია გარკვეული რაოდენობის პარამეტრებით:

- ავტომობილის მარკა;
- ავტომობილის მოდელი;
- ძრავის ტიპი (ბენზინი, დიზელი);

და ა. შ. დაწვრილებით იხილეთ დანართი 1-ში.

პროგრამისტს შემოტანილი აქვს ავტომანქანის კლასის ცნება. ფორმირებული აქვს 3 წესი, ავტომობილების კლასების შესახებ. წესებში გერისტოკული პირობების ჩასმით იგი განასხევენ ავტომობილებს მათი ღირებულების მიხედვით. ასე მაგ. I კლასის ავტომობილად ის თვლის ისეთ ავტომობილს, რომლის ღირებულებაც მოთავსებულია გარკვეულ დიაპაზონში: 80000 \$ დან 170000 \$-მდე. რასაკვირველია, ეს ფასები რეალური ფასებია, აღებულია შესაბამისი კატალოგებიდან და მოთავსებულია ამოცანის ც/ბ-ში.

mashini_1_clasa(Marka,Model) :-
mashina(Marka, Model, _ , _ , _ , Cena), 80000 < Cena, Cena =< 170000.

თუ მომხმარებელს სურს მოძებნოს ამ კლასის ავტომობილები ც/ბ-ში მან პ/ს-ს უნდა დაუსახოს შემდეგი მიზანი:

```
?- write( "Mashini_1_clasa:" ),nl,nl,  
   mashini_1_clasa( Marka, Model ), write( Marka ), write( " " ),  
   write( Model ), write( " " ),nl,nl, fail.
```

პ/ს გასცემს ადექვატურ პასუხს  **Output** გულში: იხ. ნახ. 2.14.

```
Mashini_1_clasa:  
*****  
audi a8 4.2 quattro  
audi s8 4.2 quattro  
bmw 540 a touring  
bmw e38-750 il  
bmw e31-840 ci  
chrysler viper  
land_rover range rover 4.6  
mercedes_benz e430  
mercedes_benz s600(w-140)  
mercedes_benz g320  
porsche 911 carrera  
porsche 911 carrera cabrio  
jaguar sovereign 4.0 lwb  
jaguar executive 3.2  
jaguar xk-8 coupe  
jaguar xkr
```

ნახ. 2.14.

პროგრამისტს ასევე შემოტანილი აქვს ძლიერი და სუსტი ავტომატური ცნება. ამ პარამეტრით იგი აფასებს ავტომატურად ცხენის ძალის მიხედვით. შემოაქვს ევრისტიკული შეფასება ამ პარამეტრის საფუძველზე. ასე მაგ. „ძლიერი“ მანქანას იგი უყენებს შემდეგ ევრისტიკულ პირობას: „ძლიერი“ ავტომატური სიდიდის ც.ძ.-ის სიდიდე უნდა იყოს წესში მოცემულ საზღვრებში:

mashini_moshnie(Marka, Model) :-
 mashina(Marka, Model, _, _, X, _), 180 < X, X =< 455.

შესაბამის შეკითხვას აქვს სახე:

?- write("Moshnie mashini:"), nl, nl,
 mashini_moshnie(Marka, Model), write(Marka), write(" "),
 write(Model), write(" "), nl, nl, fail.

პასუხი შეკითხვაზე გაიცემა  ელში: იხ. ნახ. 2.15.

Moshnie mashini:

 audi a8 4.2 quattro
 audi s8 4.2 quattro
 audi tt 1.8 turbo
 bmw m3
 bmw 328i cabrio
 bmw z3 m s50
 bmw 540 a touring
 bmw e38-750 il
 bmw e31-840 ci
 volvo s90
 volvo s70 და ა.შ.

ნახ. 2.15.

ასევე შესაძლებელია სხვა სახის მიზნების დასმაც: მაგ. რომელი ავტომობილების ძრავები მუშაობენ დიზელზე ან ბენზინზე, რა ტიპის ძარა აქვს ამა თუ იმ ავტომობილს და ა.შ. დაწერილებით ყველა ამ შეკითხვაზე პასუხის მიღების სურვილის შემთხვევაში შესაძლებელია მიემართოთ დანართი 1-ს და ამოცანა გაეუშვათ შესრულებაზე.

ამოცანა 4. მსოფლიოს კურორტები.

ამოცანა ეხება მსოფლიოს კურორტებს. შედგენილ ცოდნის ბაზაში (ც/ბ) შეტანილია საკურორტო ადგილების მახასიათებლები:

- კურორტის ადგილმდებარეობა;
- ტურისტის განსათავსებელი ადგილი: სასტუმრო, კოტეჯი;
- სასტუმროს სახელწოდება;

როგორც წინა ამოცანებში, აქაც მომხმარებელს აქვს შესაძლებლობა დასვას სხვადასხვა შეკითხვები და კი-ის მიერ მიიღოს პასუხი. დაწერილებით იხილეთ დანართი 1-ში.

ამოცანა 5. მსოფლიოს ქვეყნები

ამოცანის მიზანია მსოფლიოს ქვეყნების მიმოხილვა. პროგრამისტის მიერ ფორმირებულია ცოდნის ბაზა (კ/ბ). იგი ახასიათებს მსოფლიოს სხვადასხვა კონტინენტის ქვეყნებს შემდეგი მახასიათებლებით:

- ქვეყნის დასახელება;
 - კონტინენტი, რომელზეც მდებარეობს ესა თუ ის ქვეყანა;
 - ქვეყნის სახელმწიფო მოწყობის ფორმა (რესპუბლიკა, კონსტიტუციური მონარქია, სამეფო, საგრაფო და ა. შ.);
 - ქვეყნის დედაქალაქი;
 - ქვეყნის მოსახლეობის რაოდენობა;
 - ქვეყნის ტერიტორია (1000 კვ.კმ);
- და ა. შ.

ასევე პროგრამისტის მიერ შემოტანილია ორი „ვერისტიკული“ წესი, რომელიც სხვადასხვა ქვეყნებს ახასიათებს ისეთი „კომბინირებული“ მახასიათებლით, როგორიცაა: „დიდი ქვეყანა“ (big_country). ან „პატარა ქვეყანა“ (small_country). მოვიყვანოთ ეს წესები: იხ. ნახ. 220.

***** C E S E B I *****

big_country(A) :-
country(A, _, _, M, D, _), M >= 30.0, D >= 200.0.

small_country(C) :-
country(C, _, _, S, F, _), S <= 10.0, F <= 100.0.

ნახ. 220.

ვერისტიკულ წესებში ქვეყნის „კომბინირებული“ მახასიათებელი აღწერილია ორი პარამეტრით: მოსახლეობის რაოდენობა და ქვეყნის ტერიტორია. პროლოგ-პროგრამის შემდგენლის აზრით, „დიდი ქვეყანა“ ეს ისეთი ქვეყანაა, რომლის მოსახლეობაც მეტია ან ტოლია ოცდაათ მილიონ მცხოვრებზე და რომლის ტერიტორიაც მეტია ან ტოლია 200 ათას კვ. კმ-ზე. ქვეყანა პატარაა, თუ მოსახლეობის რაოდენობა ნაკლებია ან ტოლია 10 მილიონ მცხოვრებზე და ტერიტორია ნაკლებია ან ტოლია 100 ათას კვ. კმ-ზე. წესების ბაზაზე ფორმირებულია მიზნები. იხ. ნახ. 221.

?- write(" DIDI GVEKNEBIA: "), nl,
 write("@@@@@@@@@@@@@@@@@@@@"),nl,nl,
 big_country(A), write(A), nl, nl, fail.

ნახ. 221.

3/-ს ამ შეკითხვასე  Output ქვეყანაში გამოაქვს შედეგი. იხ. ნახ. 222.

DIDI GVEKNEBIA:
 @@@@@@@@@@@@@@

england
 france
 italy
 espana
 polska
 vietnami
 indoeti
 irani
 chineti
 tailandi
 turketi
 iaponia
 kolumbia
 brazilia
 argentina
 usa
 kanada
 mexsika
 No.

ნახ. 222

ხოლო შეკითხვასე, იხ. ნახ. 223.

?- write(" PATARA CVECNEBIA: "), nl,
 write("*****"),nl,nl,
 small_country(F), write(F), nl, nl, fail.

ნახ. 223.

პასუხი იქნება შემდეგი. იხ. ნახ. 224.

PATARA QVEKNEBIA:

danial
 luksemburg
 vatikan
 avstria
 belgia
 malta
 monaco

- სასტუმროს დონე (ვარსკვლავების რაოდენობა);
- მომსახურების დონის სხვადასხვა მაჩვენებელი;
- სასტუმროში ცხოვრების ერთი დღის ღირებულება;
- სასტუმროს ადგილმდებარეობა (ქვეყნის ჩვენებით) და ა.შ.

პროგრამისტის მიერ შემოტანილია ორი ევრისტიკული წესი: იხ. ნახ. 2.16.

%*****Rules:*****

```
magali_momsaxurebis_sastumro(A):-
kurorti(Qalaci,_,Name,Star,_,Pasi,_)
Star >= 3, Pasi >= 40,write(Qalaci),
write(" "),write(Name),write(Star),
write(" *"),write(" "),write(Pasi),
write(" lari."),nl,nl.
```

```
dabali_momsaxurebis_sastumro(D):-
kurorti(Qalaci,_,Name,Star,_,Pasi,_)
Star <3,Pasi <60,write(Qalaci),
write(" "),write(Name),write(Star),
write(" *"),write(" "),write(Pasi),
write(" lari."),nl,nl.
```

ნახ. 2.16.

პროგრამისტის შეხედულებით მაღალი მომსახურების სასტუმროდ შესაძლებელია წაითვალოს სასტუმრო, რომლის ვარსკვლავიანობა მეტია ან ტოლია 3-ის, სასტუმროს ერთი დღის ღირებულება მეტია ან ტოლია 40 ლარის და ა. შ.

ევრისტიკულ წესებზე დაყრდნობით შესაძლებელია შემდეგი მიზნის ფორმირება:

```
?-write("MAGALI MOMSAXUREBIS SASTUMRO"),nl,
write("*****"),
nl,nl,
magali_momsaxurebis_sastumro(S),fail.
```

ამ შეკითხვაზე კს პასუხობს  Output ელში შემდეგნაირად: იხ. ნახ. 2.17.

MAGALI MOMSAXUREBIS SASTUMROEBI:

gudaure: marko_polo 4*. 80 lari.

batumi: L.bakuri 5*. 70 lari.

batumi: tanamgzavri 4*. 60 lari.

trapisoni: besti3* 40 lari.

opatia: vila de almasadori 4*. 70 lari.

opatia: hotel and vila 3*. 60 lari.

No.

ნახ. 2.17.

შემდეგი მიზანი ასე ჩამოყალიბდით: სადა მდებარეობს ესა თუ ის კურორტი.

```
?-write("KURORTIS ADGILMDEBAREOBA"),nl,
write("*****"),nl,nl,
kurorti(Adgili,_____),
write(Adgili),nl,fail.
```

დასახულ მიზანზე კ/ს  Output ეელში გამოაქვს შემდეგი ინფორმაცია: იხ. ნახ. 2.18.

KURORTIS ADGILMDEBAREOBA:

gudauri
bakuriani
bakuriani
bakuriani
bordgomi
batumi
batumi
batumi
shovi
shovi
maltakva
kvareli
tsxvarithamia
trapisoni
opatia
opatia
No.

ნახ. 2.18..

შეკითხვაზე: რომელი სასტუმროებია მოთავსებულია ქ. ბათუმში:

```
?-write("BATUMIS SASTUMROEBI"),nl,
write("*****"),nl,nl,
kurorti(batumi,_____,Dasaxeleba,_____),
write(" "),
write(Dasaxeleba),nl,nl,fail.
```

პროლოგ-სისტემა პასუხობს  Output ქვეყანაჯარაში შემდეგს: იხ. ნახ. 2.19.

BATUMIS SASTUMROEBI

L.bakuri;

tanamgzavri;

inturisti და ა.შ.

ნახ. 2.19.

shveicaria
sakartvelo
azerbaijani
somxeti
izraeli
kuveiti
singapuri
salvadori
kostarika
panama
No.

ნახ. 224

პროლოგ-სისტემას (პ/ს), დიალოგურ რეჟიმში, შესაძლებელია ამოცანის შესაბამისი საპრობლემო არიდან დაეუსეთ სხედასხვა სახის შეკითხვები. მაგალითად, როგორია ამა თუ იმ ქვეყნის სახელმწიფო მოწყობის ფორმა. იხ. ნახ. 225.

```
?- write( " EVROPIS SAXELMCIPOEBIS MOCKOBIS FORMA:" ),
nl,write("*****"),
nl,nl,country( Name, europa, Porma, _ _ _ ),
write( Name ), write( "——" ), write( Porma),
write( "." ), nl, nl, fail.
```

ნახ. 225.

ამ შეკითხვაზე პ/ს გზ Output - ელში გამოიტანს შედეგს. იხ. ნახ. 226.

EVROPIS SAXELMCIPOEBIS MOCKOBIS FORMA:

england——gaertianebuli samepo.
france——republika.
italy——republika.
espana——samepo.
polska——republika.
dania——samepo.
luksemburg——sagercogo.
vatican——calaci saxelmcipo.
avstria——republika.
belgia——samepo.
malta——republika.
monaco——satavado.
norvegia——samepo.
shveicaria——confederacia.
shvecia——samepo.
sakartvelo——republika.
No.

ნახ. 226

ნამოკადალიბოთ შემდეგი მისანი (შეითხვა). გამოეიტანოთ ქვეყანჯარაშო ეუროპის სახელმწიფოების დელაქალაქები. იხ. ნახ. 2.27.

```
?- write(" EVROPIS
DEDAQALAEIBIA:"),nl,write("@@@@@@@@@@@@@@"),nl,nl,
country( Name, europa, _, Capital, _, _ ), write( Name ), write( "
"), write( Capital ),
nl, nl, fail.
```

ნახ. 2.27.

ამ შეითხვაზე პასუხი ასეოია. იხ. ნახ. 2.28.

EVROPIS DEDAQALAEIBIA:

```
england london
france paris
italy rom
espana madrid
polska varшава
dania kopenhagen
luksemburg luksemburg
vatikan vatikan
avstria vena
belgia briuseli
malta valleta
monaco monaco
norvegia oslo
shveicaria berni
shvecia stogholmi
sakartvelo tbilisi
No.
```

ნახ. 2.28.

შეითხვაზე: აზიის ქვეყნების მოსახლეობის რაოდენობა? ა/ს პასუხოზს. იხ. ნახ. 2.29.

ASIIS QVEKNEBIS MOSAXLEOBIS RAODENOBAA:

```
azerbaijani 7.4 ml.
somxeti 3.412 ml.
vietnami 70.9 ml.
izraeli 5.5 ml.
indoeti 897 ml.
irani 61 ml.
chineti 1179 ml.
kuveiti 1.4 ml.
nepali 19.3 ml.
singapuri 2.88 ml.
siria 13.4 ml.
```

tailandi	57.8 ml.
turketi	59.9 ml.
iaponia	127 ml.
No.	

ნახ. 2.29.

შემდეგი კითხვა ასე ჩამოვყალიბოთ: სამხრეთ ამერიკის კონტინენტის ქვეყნების ტერიტორიებია? იხ. ნახ. 2.30. პასუხი ამ შეკითხვაზე გამოდის  ქვეყანაჯარაში.

SAMXRET AMERICA: TERITORY:

kolumbia 1139 1000 kv.km.

venesuela 912 1000 kv.km.

ekvadori 284 1000 kv.km.

peru 1285 1000 kv.km.

brazilia 8512 1000 kv.km.

bolivia 1099 1000 kv.km.

paragvai 407 1000 kv.km.

cile 757 1000 kv.km.

argentina 2777 1000 kv.km.

urugvai 178 1000 kv.km.

No.

ნახ. 2.30.

მომხმარებელს აქვს შესაძლებლობა ა/ს-ს დაუსახოს სხვა მიზნებიც, რომლებზეც იგი მიიღებს ადეკვატურ პასუხს. დაწერილებით შესაძლებელია იხილოთ დანართი 1-ში.

ანალოგიური ტექნოლოგიებით მომხმარებელს (სტუდენტი, მაგისტრი, ხო-ის საკითხებით დაინტერესებული პირი) აქვს შესაძლებლობა დაისახოს გადასაწყვეტი პრობლემა (ნებისმიერი საპრობლემო არისათვის), ასახოს იგი წარმოდგენის რომელიმე ხელოვნური ენის (პრედიკატების თეორია, გრაფების თეორია, სემანტიკური ქსელები და ა. შ.) საშუალებით, აირჩიოს პროგრამირების ალგორითმული ენა, დააპროექტოს ცოდნის ბაზა (ც/ბ), შეადგინოს პროლოგ-პროგრამა და გადაწყვიტოს პრობლემა.

თავი 3. ალტერნატივის ბაღარჩევის მართვა. გადაწყვეტილების მიღების პროცესის ავტომატიზაცია

გადაწყვეტილების მიღები პირი (გპ) მმართველი გადაწყვეტილებების ფორმირებისას ეფუძნება ობიექტურ და სუბიექტურ ფაქტორებს. მათ შესახებ ჩვენ ერცლად ეისაუბრეთ პირელ თაეში და [7] ნაშრომში. გპ გადაწყვეტილების მიღებისას, ასეეე იეენებს თაეის ცოდნას, გამოცდილებასა და ინტუიციას. ამ საში კომპონენტიდან ცოდნისა და გამოცდილების (დროის გარკეულ შუალედში დაგროვილი ცოდნა) ე.წ. „გაინჟინრება“ შესაძლებელია ცოდნის ბაზების (ე/ბ) სახით. ამ პროცესს აწარმოებს სპეციალისტი, რომელსაც „ცოდნის ინჟინერი“ ეწოდება. იგი კონკრეტული „საპრობლემო არის“ ექსპერტთან ერთად აპროექტებს ექსპერტულ სისტემას. ექსპერტული სისტემა (ეს) ცოდნაზე დაფუძნებული პროგრამული პროდუქტია. [7] ნაშრომში დაწერილებით იქნა განხილული გრაფების თეორია. გრაფი, ხელოვნური ინტელექტის სისტემებში (ხი), ამონახსნთა სიერცეში ამოცანის შედეგის ძებნის ერთერთი ყველაზე ეფექტური და შედეგაიანი ინსტრუმენტი. გრაფში ძებნის მეთოდების: ძებნა დაბრუნებებით (backtrack), ძებნა სიერეში, ძებნა სიღრმეში, ეერისტიკული ძებნა, თეორიული და პრაქტიკული ასპექტები განხილულია [7] ნაშრომში. კონკრეტულ ამოცანებსა და პროლოგ-პროგრამებს ამ მეთოდების შესახებ ნაშრომის მეორე ნაწილში განვიხილავთ.

პროლოგ-იმიტატორს „Strawberry Prolog“ ჩაშენებული აქვს სპეციალური მექანიზმი, რომელიც აადეილებს ალტერნატიული ამონახსნების სიერცეში „ოპტიმალური“ ვარიანტის ძებნას. მას ჩატრის (სუტ) მექანიზმს უწოდებენ. გამოიეყენოთ ეს მექანიზმი (ოპერატორი) ე.წ. „არითმეტიკული ამოცანების“ ამოხსნის გასამარტივებლად.

დასახული მიზნის (Goal) მიღწევის გზაზე პროლოგ-სისტემა აწარმოებს ვარიანტების ავტომატურ გადარჩევას. მიზნის ვერ მიღწევის შემთხვევაში იმიტატორი აკეთებს უკან დაბრუნებას და განაგრძობს ძებნას. ამასთან ახდენს ძეელი ბილიკის (path) მოეეათას (სუტ), რაც „შეღკოშში აადეილებს ძებნის პროცესს. ასეთი გადარჩევის მექანიზმის იმიტატორში ჩაშენება ამარტივებს პროლოგზე პროგრამირების პროცესს. ჩატრის მექანიზმი ანთაეისუფლებს პროგრამისტს იმისგან, რომ მან თეით დააპროგრამოს ეს ყრარემენტი. სხეა კუთხით თუ შეეხედაეთ ამ საკითხს, შეუზღუდაეი, არაკონტროლირებადი გადარჩევა შესაძლებელია იქცეს პროგრამის არაეფექტური მუშაობის წყაროდ (ხშირ შემთხვევებში შესაძლებელია ადგილი კონდეს პროგრამის ჩაციკლას).

3.1. ჩატრის (სუტ) მექანიზმის გამოყენების მაგალითი.

განვიხილოთ პროლოგის „არითმეტიკული ნაწილის“ რამდენიმე მაგალითი:

1. ორ რიცხვს შორის max-ის პოვნა, პროლოგ-პროგრამას აქვს ქვემოთ მოყვანილი სახე (იხ. ნახ. 3.1):

*****M A X-is gamotvla(cut-it.)*****

MAX-is gamotvla shesadzlebelia procedurit!

max(X,Y,X):-X>=Y.

max(X,Y,Y):-X<Y.

ეს ორი ცესი ურტიერტ-გამომრიცხავი ცესებია!!!

C U T-is gamokenebit procedura martivdeba!

***** G O A L!!!*****

%?-max(8,4,W),write("W="),write(W),nl,nl.

%?-max(51000,27000,R),write("R="),write(R),nl.

?-max(27000,51000,T),write("T="),write(T),nl.

max(X,Y,X):-X>=Y,!.
max(X,Y,Y).

ნახ. 3.1.

დავისახოთ მარტივი მიზანი. ორ რიცხვს: 27000 და 51000-ს შორის ეიპოვოთ max. პროცედურას აქვს ნახ. 3.2-ზე მოცემული სახე:

max(X,Y,X):-X>=Y,!.
max(X,Y,Y).

ნახ. 3.2.

ამოცანის ამოხსნის შედეგი გამოდის  ქვეყანჯარაში:

```
Compiling the file:  
C:\Program Files\Strawberry Prolog\beso\kuku pro  
0 errors, 0 warnings.
```

T=51000

Yes.

მაქსიმალური შედეგი: $\max = 51000$.

ნახ. 3.1-ზე მოყვანილი პროლოგ-პროგრამის $\max$ -ის მოძებნის „სტანდარტული პროცედურა“:

$$\max(X, Y, X) :- X \geq Y. (1)$$

$$\max(X, Y, Y) :- X < Y. (2)$$

პროცედურის ეს ორი წესი ურთიერთგამომრიცხავი წესებია. თუ შესრულდა პირველი წესი, მეორე აუცილებლად განიცდის მარცხს და პირიქით. ამის გამო, შედარებით „ეკონომიური“ ფორმულირებისთვის, ჩვენ გამოვიყენეთ ცნება: „თუ არა და“. მისი პროგრამული რეალიზაციისათვის კი ჩამოჭრის „cut“ მექანიზმი.

2. სიისადმი ელემენტის მიკუთვნების პროცედურა (member), cut-ის გამოყენებით

პროლოგის „არითმეტიკულ“ ნაწილს მიეძღვნა [7] ნაშრომის მეხუთე თავი, რომელშიც მოყვანილია მოცემული ამოცანების ამოხსნის „სტანდარტული“ პროცედურები. ჩვენ ამ ქვეთავში განვიხილავთ „არითმეტიკული“ ამოცანების გარკვეულ რაოდენობას. მათი პროცედურები უფრო კომპაქტურია „სტანდარტულ“ პროცედურებთან შედარებით, cut მექანიზმის გამოყენების გამო. ამოცანის ტექსტი მოყვანილია ნახ. 3.3-ზე:

*****SIISADMI ELEMENTIS MIKUTVNEBIS (MEMBER)*****

***** PROCEDURA (CUT-is gamokenebit.)*****

*****G O A L!!!!*****

%?-member(a,[a,b,c,d,a,g]).

?-member(2,[9,8,7,6,2,2,3]).

member(X,[X|_]):-!.

member(X,[_Y]):-

member(X,Y).

%*****

ნახ. 3.3.

პროლოგ სისტემას (პ/ს) დავესახოთ მისანი: ეკუთვნის თუ არა რიცხვი 2 [9,8,7,6,2,2,3] სიას?

ამოცანის შესრულების შედეგი გამოდის



ეკლში იხ. ნახ. 3.4.

Compiling the file:

C:\Program Files\Strawberry Prolog\beso\lolo.pro
0 errors, 0 warnings.

Yes.

ნახ. 34.

წაკეთის „cut“ მექანიზმის მოქმედების უკეთ გასაგებად შევასრულეთ იგივე ამოცანა ე.წ. „ბიჯობრივ“ რეჟიმში, რომელიც დაწერილებით ნაშრომის პირველ თავში განვიხილეთ. იხ. ნახ. 35.

Proof Tree - step 7.

```
?-  
member(2, [9, 8, 7, 6, 2, 2, 2, 3]).  
member(2, [8, 7, 6, 2, 2, 2, 3]).  
member(2, [7, 6, 2, 2, 2, 3]).  
member(2, [6, 2, 2, 2, 3]).  
member(2, [2, 2, 2, 3]).  
!.
```

ნახ. 35.

3. სიაში ახალი ელემენტის ჩამატების (Insert) პროცედურა
ამოცანის ტექსტი მოყვანილია ნახ. 36.

SIASHI ELEMENTIS CHAMATEBA (INSERT)
CUT(!)-is gamokenebit.

*****KOMENTARI*****

predicati(damokidebuleba) INSERT-i carmo-
vidginot, rogorc 3 argumentiani predikati:
insert(X,L,L1),sada
X-chasamatebeli elementia;
L-siaa,romelshic unda chavamatot X;
L1-axali siaa.
tu X aris L siis cevri,mashin L1=L.
else, L1 L siaa X axali elementit.

*****G O A L!!!*****

```
/*?-insert(d,[a,b,c],R),write("R="),write(R),  
nl,nl.*/
```

```
/*?-insert(1,[2,3,4,5],W),write("W="),write(W),
```

-----PROCEDURA-----

insert(X,L,L1):-member(X,L),!.

insert(X,L,[X|L]).

X-s vamatebt siis tavshi.

member(X,[X|_]):-!.

member(X,[_Y]):-member(X,Y).

ნახ. 3.6.

4. ობიექტების კლასიფიკაციის ამოცანა

ამოცანის არსი შემდეგში მდგომარეობს. ვთქვათ მიმდინარეობს საჩოგბურთო ტურნირი. ჩვენ დაეპროექტეთ ცოდნის ბაზა (ც/ბ), რომელიც შეიცავს ინფორმაციას ნათამაშები პარტიების შესახებ. მოთამაშე წყვილების შერჩევა რაიმე სისტემით არ ხდებოდა, უბრალოდ თვითოეული მოთამაშე ხდებოდა რამოდენიმე მოწინააღმდეგეს. შეხვედრების შედეგები ც/ბ-ში მოთავსებულია ფაქტების სახით, ჩვენი მიზანია განესაზღვროთ დამოკიდებულება (პრედიკატი): კლასი (class), რომელიც ახდენს ჩოგბურთელების რანჟირებას კატეგორიების მიხედვით:

კლასი(ჩოგბურთელი, კატეგორია).

შემოვიტანოთ სამი კატეგორია:

1. გამარჯვებული. ნებისმიერი ჩოგბურთელი, რომელსაც არც ერთი თამაში არ წაუგია.
 2. მებრძოლი. ნებისმიერი ჩოგბურთელი, რომელსაც აქვს როგორც მოგებული, ასევე წაგებული თამაშებიც.
 3. სპორტსმენი. ნებისმიერი ჩოგბურთელი, რომელმაც ყველა მატჩი წააგო.
- სამეფო კატეგორიისათვის შემოვიტანოთ შემდეგი წესები:

X – მებრძოლია, თუ არსებობს რომელიღაც Y, ისეთი, რომ X-ს მოგებული აქვს Y-თან,
და

არსებობს რომელიღაც Z, ისეთი, რომ Z-ს მოგებული აქვს X-თან.

X – გამარჯვებულია, თუ

X მოუგო რომელიღაც Y-ს და

X არაგისთან ჯერ წაუგია.

წესების ეს ფორმულირებები შეიცავენ ლოგიკურ უარყოფას „არა“-ს. პროლოგში ლოგიკური ოპერაცია უარყოფისათვის შემოღებულია ჩაშენებული პრედიკატი „not“. მისი გამოყენებით, საბოლოოდ ჩვენი პროლოგ-პროგრამა მიიღებს სახეს, რომელიც ნახ. 3.7-ზეა მოცემული:

*****OBIEQTEBIS KLASIFIKACIIS AMOCANA*****

*****KOMENTARI:*****

*****TENISIS KLUBI*****

Davushvat chven gvaqvs tenisis clubis turniris monacemebi faqtebis saxit:
 gamarjvebuli (dato, givi); gamarjvebuli (zura, dato); da ase shemdeg.
 Shemovitanot damokidebuleba:

class(motamashe,kategoria).

Chvens magalitshi avigot 3 kategoria :

1. gamarjvebuli;
2. mebrdzoli;
3. sportmeni.

gamarjvebulia,romelmac moigo kvela matchi.
 mebrdzolma zogi moigo da zogi caago,xolo
 sportmenma kvela tamashi caago!!!

gamovikenot .N O T. operatori

shedegi „Yes“-ის დროს vnaot „Proof Tree“-ში.

*****MIZNEBI:*****

%?-class(besiki,sportmeni).

%?-class(besiki,mebrdzoli).

?-class(besiki,gamarjvebuli).

%?-class(gela,sportmeni).

%?-class(bacho,gamarjvebuli).

%?-class(bacho,sportmeni).

%?-class(bacho,mebrdzoli).

%?-class(lasha,sportmeni).

```
/*-write( "BESIKIM MOUGO:"),nl,nl,
moigo(besiki,W),
write(W),point,nl,nl,fail.*
```

```
/*?-write(" BACHOM CAAGO:"),nl,
write("*****"),nl,nl,
moigo(R,bacho),write(R),point,nl,nl,fail.*/

/*?-write(" CAGEBULI CHOGBURTELEBI:"),nl,
write("*****"),
nl,nl.moigo(_,T),write(T),point,nl,nl,fail.*/
```

*****CODNIS BAZA :*****

*****FAQTEBI: *****

```
moigo(besiki,gela).
moigo(gela,bacho).
moigo(besiki,bacho).
moigo(bacho,lasha).
moigo(besiki,dato).
moigo(bacho,misha).
moigo(besiki,misha).
moigo(bacho,dato).
moigo(besiki,lasha).
```

*****CESEBI:*****

```
class(X,mebrdzoli):-
moigo(X,_),
moigo(_,X).
```

```
class(X,gamarjvebuli):-
moigo(X,_),
not(moigo(_,X)).
```

```
class(X,sportmeni):-
not(moigo(X,_)).
```

***** PROGRAMIS DASASRULI *****

ნახ. 3.7.

ამოცანას დაეუსახოთ მიზნები:
ბესიკი გამარჯვებულია.

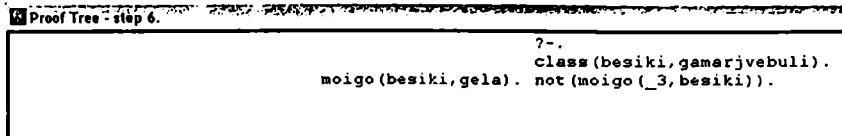
?-class(besiki, gamarjvebuli). შედეგების  ელში მივიღებთ პასუხს: „Yes“.

იხ. ნახ. 3.8.

```
Compiling the file:
C:\Program Files\Strawberry Prolog\beso\tenisi.pro
0 errors, 0 warnings.
Yes.
```

ნახ. 3.8.

მართლაც, თუ გადავხედავთ ც/ბ-ს, ბესიკს არც ერთი მატჩი არ წაუგია, დამამტკიცებელი ხის ! ნიშნაკზე მიმთითებლის დაწკაპუნებით გამოდის ფანჯარა, რომელიც ნახ. 3.9-ზეა ასახული:



ნახ. 3.9

„დამამტკიცებელი ხე“ იძლევა შემდეგ ინფორმაციას: ამოცანამ შედეგის მიღწევამდე გაიარა 6 ბიჯი (step 6). გამოიყენა რა „გამარჯვებულის“ წესი „not“ ლოგიკური კავშირით. ამ წესის მიხედვით არ არსებობს არც ერთი სპორტსმენი, რომელსაც ბესიკისთვის მოეგოს. not (moigo(_ X)).

პროლოგ-სისტემას (პ/ს) დავეუსახოთ შედარებით „რთული“ მიზანი: „ვის მოუგო ბესიკი?“.

```
?-write( "BESIKIM MOUGO:"),nl,nl,
moigo(besiki,W),
write(W),point,nl,nl,fail.
```

„Proof Tree“ ფანჯარაში გამოდის შემდეგი ინფორმაცია. იხ. ნახ. 3.10.

Compiling the file:
C:\Program Files\Strawberry Prolog\beso\tenisi.pro
0 errors, 0 warnings.

BESIKIM MOUGO:

gela
bacho
dato
misha
lasha
No.

ნახ. 3.10

ასევე შესაძლებელია სხვა მიზნების დასახვა, რომლებიც მოთავესებულია მიზნების ჩამონათვალში.

გადაწყვეტილების მიღების პროცესი საფუძვლად უდევს ადამიანის ნებისმიერი სახის მოქმედებას.

გადაწყვეტილების მიღები პირი (გმპ), ეს ყველა ინდივიდია, რომელიც იღებს როგორც საყოფაცხოვრებო, ასევე სხვა სახის გადაწყვეტილებებს. როგორც ზემოთ აღინიშნა, გმპ-ს გადაწყვეტილებებზე მოქმედებს ობიექტური და სუბიექტური ფაქტორები. გადაწყვეტილების მიღების პროცესი არჩევის პროცესია. გმპ ამონახსნთა სიერცეში, ალტერნატიული ვარიანტებიდან გარკვეულ მოსაზრებებზე (ვერისტიკებზე) დაყრდნობით, ირჩევს მოცემული მომენტისათვის მისი აზრით საუკეთესო ამონახსნის ვარიანტს. არჩეული შესაძლებელია არ აღმოჩნდეს საუკეთესო. არჩევის გასაკეთებლად საჭიროა მინიმუმ ორი ალტერნატივა მაინც. ერთი ალტერნატივის შემთხვევაში არჩევის პროცესი არ მიმდინარეობს. შესაბამისად არ ხდება გადაწყვეტილების მიღება, თუმცა „არაფრის“ კეთება ასევე გადაწყვეტილებაა.

ქვეთავში მოყვანილია ამოცანები, რომლებიც სხვადასხვა „საპრობლემო“ არეს შეეხება.

ამოცანა 1. ფირმაში ეკანტური თანამდებობის შერჩევა

ამოცანის ძირითადი არსი შემდეგში მდგომარეობს. ეთქვამთ, რომელიმე აბსტრაქტულ, მეცნიერული მიმართულების ფირმაში არის ეკანტური თანამდებობები, რომლებზეც ცხადდება კონკურსი. კონკურსის შედეგებს ანალიზებს დირექტორთა საბჭო და ფირმის ხელმძღვანელი და იღებს მმართველ გადაწყვეტილებებს. ჩვენს შემთხვევაში ეკანტური თანამდებობების დაკავების შესახებ.

სამუშაოს მაძიებლის საკონკურსო შერჩევა ხორციელდება რამოდენიმე ეტაპად. პირველ ეტაპზე ც/ბ-ში ფირმის თანამშრომლის (ოპერატორის) მიერ შეიტანება სამუშაოს მაძიებლის პირადი მონაცემები ე.წ. „სიეი“ (ვინაობა, დაბადების წელი, განათლების ცენზი, საცხოვრებელი ადგილი, ენების ცოდნა, კომპიუტერის ცოდნის დამატატურებელი სერთიფიკატები და ა.შ.).

ფირმაში გამოცხადებულია შემდეგი სამი ეკანსია:

- მეცნიერი-მუშაკი;
- ინჟინერ-კონსტრუქტორი;
- ექსპლუატაციის ინჟინერი;

ეკანტური თანამდებობების ჩამონათვალი დალაგებულია პრიორიტეტების მიხედვით. რადგანაც ფირმა მეცნიერული მიმართულებისაა, ამიტომ მეცნიერ-მუშაკის თანამდებობა ყველაზე მაღალ პრიორიტეტულია.

საკონკურსო შერჩევის შემდგომ ეტაპზე წარმოებს მაძიებლის შეფასება შემდეგი პარამეტრებით:

- სამეცნიერო ხარისხის არსებობა (Degree);
- სამეცნიერო აღმოჩენების არსებობა (Discovery);
- ინსტიტუტის დიპლომის საშუალო ნიშანი (Grade);
- მოცემული სპეციალობით მუშაობის საერთო სტაჟი (Experience).

პროლოგ-პროგრამის პირველი ეტაპის ფრაგმენტს აქვს ნახ. 3.11. წარმოდგენილი სახე:

AMOCANIS AMOXSNIS I ETAPI.

SAMUSHAOS MADZIEBLIS MONACEMEBIS SHETANA.

?-monacemi.

monacemi:-

```
write(" MOGVACODET TKVENI MONACEMEBI:"),
```

```
nl,nl,
```

```
read(Monacemi),write(Monacemi),nl,nl,
```

```
monacemi.
```

ნახ. 3.11.

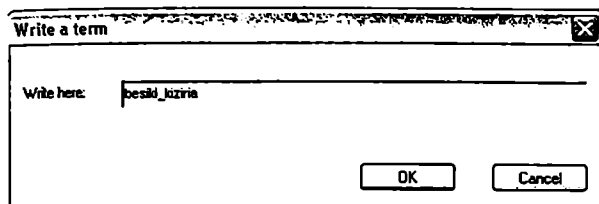
ამოცანის I ეტაპზე ხდება „სამუშაოს მაძიებლის“ სივს შეტანა ც/ბ-ში. ამ ეტაპზე გამოყენებულია პროლოგის იტერაციული პროცესების ორგანიზების მექანიზმი „რეკურსია“. ფრაგმენტის ბოლო სრულდება ატომით „monacemi“ (ამოცანის ნახ. 3.11.-ზე წარმოდგენილი ფრაგმენტი წააგავს ქვეპროგრამის გამოძახების მექანიზმს, მისი სახელით).

ინფორმაციის გამოტანის  ფანჯარაში გამოდის შეტყობინება:

```
MOGVACODET TKVENI MONACEMEBI:
```

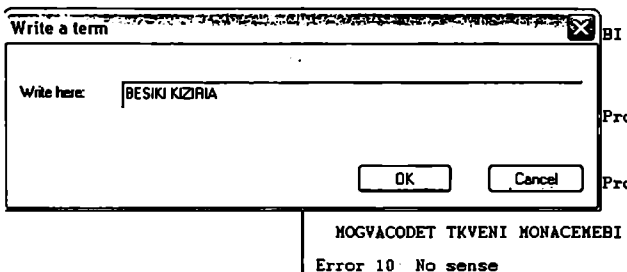
ამის შემდეგ რეკურსიის მექანიზმი იძახებს ინფორმაციის შეტანის ქვეფანჯარას იხ. ნახ. 3.12. ქვეფანჯარაში შევითანთ „სამუშაოს მაძიებლის“ პარამეტრს, მაგ. სახელსა და გვარს. მიმთითებლის დაწკაპუნებით კლავიშზე „OK“ ინფორმაცია იგზავნება ც/ბ-ში და გამოდის იდენტური ცარიელი ქვეფანჯარა ახალი ინფორმაციის შესატანად. ინფორმაციის ამოწურვის შემთხვევაში (მაძიებლის სივს პარამეტრების დასრულებისას), დაწკაპუნებთ „Cancel“-ზე. პროცესი სრულდება. ამ და სხვა ამოცანებშიც მომხმარებელმა უურადღებდა უნდა მიაქციოს ერთ გარემოებას. ინფორმაციის შეტანის ქვეფანჯარაში შესატანი ტექსტი

უნდა აიკრიფოს ლათინური პატარა ასოებით ნიშნებით, როგორც თეორიული კურსიდან ვიცით [7] . პროლოგში მხოლოდ ცვლადები ჩაიწერება დიდი ასოებით ნიშნებით.



ნახ. 3.12.

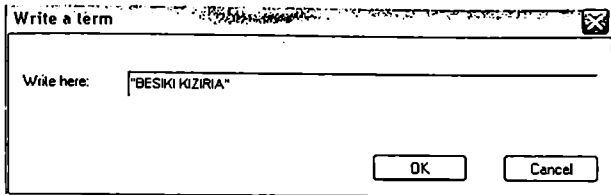
ამასთან, ტექსტის ცალკეული სიტყვები ერთმანეთისგან აუცილებლად უნდა გამოეყოფოდეს ნიშნით: „ -“, წინააღმდეგ შემთხვევაში იმიტატორი მოგვცემს შეტყობინებას სინტაქსური შეცდომის შესახებ. იხ. ნახ. 3.13.



ნახ. 3.13.

თუ აუცილებლობას წარმოადგენს დიდი ასოებით ნიშნებით (ან რომელიმე სხვა შრიფტით) ტექსტის შეტანისა ფანჯარაში, მაშინ უნდა გამოვიყენოთ აპოსტროფები „'“. ამ შემთხვევაში შეტანილი ტექსტი იმიტატორის მიერ აღიქმება, როგორც „String“ და შესაბამისად სინტაქსური შეცდომაც აღმოიფხვრება. იხ. ნახ. 3.14.

ამით ამოცანის პირველი ეტაპი დასრულებულად შეიძლება მივიჩნიოთ.



```

Compiling the file
C:\Program Files\Strawberry Prolog\beso\vakansia.pro
0 errors. 0 warnings.

```

```

MOGVACODET TKVENI MONACEMEBI:

```

```

BESIKI KIZIRIA

```

```

MOGVACODET TKVENI MONACEMEBI:

```

ნახ. 3.14.

ამოცანა შედგენილია ბლოკური პრინციპით, რაც იძლევა საშუალებას ამოცანა საჭიროებისამებრ გავეშვათ ცალკეული ეტაპების მიხედვით. მოვიყენათ პროლოგ-პროგრამის ერთერთი ეტაპის ფრაგმენტის ამონაბეჭდი და აეხსნათ მისი მუშაობის პრინციპი იხ. ნახ. 3.15.

?-position1.

Position-Tanamdeboba.
Posicion1:mecnieri-mushaki.

position1 :-

```

write(" GAQVT SAMECNIERO XARISXI?"),nl,nl,
read(Degree),write("DEGREE="),write(Degree),
nl,nl,write(" GAQVT MECNIERULI AGMOCHENEBI?"),
nl,nl,read(Discovery),write("DISCOVERY="),
write(Discovery),nl,nl,

```

Degree = yes, Discovery = yes ->

Position = mecnieri_mushaki,

```
write(" MIVIGOT PIRMASHI MECNIER-MUSHAKAD."),nl,
```

```
write("*****"),
```

```
nl,nl,write(" POSITION="),write(Position),nl,
```

```
write("*****"),nl,nl
```

```
else Position = no,
```

```
write(" PIRMASHI MIGEBAZE UARI ETQVAS!"),nl,
```

```
write("*****"),
```

```
nl,nl,write(" Position="),write(Position),point,
```

```
nl,write("*****"),nl,nl.
```

```
*****
```

ნახ. 3.15.

ამოცანის ამოხსნის ამ ეტაპზე აღწერილია ვაკანსიაზე: მეცნიერ-მუშაკი „მაძიებლის“ შერჩევის ალგორითმი. ამ თანამდებობის დასაკავებლად „სამუშაოს მაძიებელს“ სხვა პარამეტრებთან ერთად აუცილებლად უნდა გააჩნდეს სამეცნიერო ხარისხი (Degree) და მეცნიერული აღმოჩენები (Discovery). ამოცანის გაშვებისას, ინფორმაციის გამოტანის

 Output

ველში გამოდის შეკითხვა იხ. ნახ. 3.16.:

```
Compiling the file:
C:\Program Files\Strawberry Prolog\beso\vakansia
0 errors. 0 warnings.
```

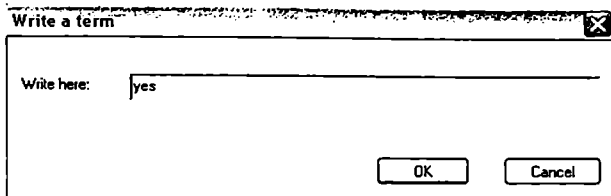
```
GAQVT SAHECNIERO XARISXI?
```

ნახ. 3.16.

ხოლო ეკრანზე გამოდის ქვეყანჯარა იხ. ნახ. 3.17. ეთქვათ, მოცემულ შემთხვევაში „სამუშაოს მაძიებელს“ გააჩნია სამეცნიერო ხარისხი. ინფორმაციის შეტანის ქვეყანჯარაში შეგუქვს „Yes“ და ვადასტურებთ „OK-ით. პროგრამა გადადის შემდეგ

ბიჯზე.  Output

ველში გამოდის შეტყობინება:



ნახ. 3.17.

ქვეყანაში „Yes” დადასტურების შემთხვევაში  ველში გამოდის შემდეგი შეტყობინება: იხ. ნახ. 3.18.

```
GAQVT SAMECNIERO XARISXI?
DEGREE=yes
GAQVT MECNIERULI AGMOCHENEBI?
DISCOVERY=yes
MIVIGOT PIRMASHI MECNIER-MUSHAKAD.
*****
POSITION=mecnieri_mushaki
*****
Yes.
```

ნახ. 3.18.

მივიღეთ მმართველი გადაწყვეტილება: „სამუშაოს მაძიებელს” თავისი პარამეტრების შესაბამისად შეეთავსეთ „მეცნიერი-მუშაკის” თანამდებობა.

ამოცანაში გამოყენებულია დასკვნის მიღების როგორც ინდუქციური (იმპლიკაცია →), ასევე დედუქციური მეთოდები.

სხვა თანამდებობების (ვაკანსიების) გადანაწილების ეტაპებს არ განვიხილავთ. ისინი თითქმის იდენტურია ზემოთ განხილული შემთხვევისა. ამოცანის დაწერილობით გაცნობის მსურველს შესაძლებლობა აქვს გაეცნოს ამოცანის სრულ ტექსტს დანართი 2-ში.

დანართი 2-ში ასევე მოყვანილია განხილული ამოცანის ანალოგიური ამოცანა, რომელშიც აღწერილია სარეკლამო სააგენტოს შესაბამისი ვაკანსიები:

კომპიუტერული გრაფიკის დიზაინერი;
დამკაბადონებელი;

ინგლისური ენის მასწავლებელი;

საოფისე პროგრამების მასწავლებელი.

ასევე მოცემულია ეაკისიების დაკავების პირობები. მომხმარებელს (სტუდენტს ან ლაბორატორიული სამუშაოს პედაგოგს) აქვს ალტერნატიული არჩევანის საშუალება. განიხილავს ამ ორი ამოცანიდან ერთერთს. ნაშრომის ავტორთა აზრით სასურველია ზემოთ განხილული ამოცანა ან მისი ანალოგიური ამოცანა ამოიხსნას ლაბორატორიულ მეცადინეობაზე პირველივე სემესტრში.

ამოცანა 2 ფინანსური მრჩეველი

[7] ნაშრომის მეორე თავში მოყვანილია ამოცანა „ფინანსური მრჩეველი“. ამოცანა აღწერილია „პრედიკატების“ თეორიის ბაზაზე. ხი-ის წარმოდგენის ამ ხელოვნური ენის საშუალებით ჩაწერილია ამოცანის, როგორც საწყისი პირობები (ფაქტები), ასევე პროდუქციული წესებიც. თავში მოცემულია ამოცანის სრული აღწერა, ამიტომ ჩვენ აქ ამ საკითხს დაწვრილებით აღარ შევხებით. როგორც აღვნიშნეთ, „პრედიკატების“ თეორია საფუძვლად დაედო ლოგიკური პროგრამირების ალგორითმულ ენას პროლოგს და მის მრავალ ეერსიას, მათ შორის „Strawberry Prolog“-ს.

ჩვენი აზრით, თეორიულ კურსში (ლექციებზე) „პრედიკატების“ თეორიის ამ ამოცანის ახსნის შემდეგ, აუცილებელია ლაბორატორიულ მეცადინეობაზე ამოვხსნათ შესაბამისი პროლოგ-პროგრამა, რომლის სრული ტექსტიც მოყვანილია დანართი 2-ში.

მოიყვანოთ ამოცანის მოკლე აღწერა:

ამოცანის დანიშნულებაა: „ფინანსური მრჩეველი“. თავისი საექსპერტო ცოდნით, დაეხმაროს ინვესტორს სწორი საფინანსო გადაწყვეტილების მიღებაში.

ფინანსურმა მრჩეველმა ძირითადად უნდა იხელმძღვანელოს შემდეგი დებულებებით:

1. ინვესტორმა, რომელსაც არა აქვს „საკმარისი“ დანაზოგი საბანკო ანგარიშზე, უნდა იფიქროს მის გაზრდაზე, მიუხედავად შემოსავლებისა.

2. ინვესტორმა, რომელსაც აქვს „საკმარისი“ დანაზოგი საბანკო ანგარიშზე და გააჩნია „კარგი“ მიმდინარე შემოსავალი, უნდა იფიქროს „სარიისკო“ გადაწყვეტილებებზე: ნაწილი თანხით შეიძინოს აქციები, ხოლო დარჩენილი თანხით შეიყვანოს საბანკო ანგარიშში.

იგულისხმება, რომ ინვესტორებს აქვთ თავის კმაყოფაზე მყოფი პირები. ამ პირებისათვის შემოგვაქვს ორი დამატებითი პირობა:

ა) თვითოეულ კმაყოფაზე მყოფ პირზე ინვესტორს ანგარიშზე აუცილებელია ედოს 5000 ლარი მაინც;

ბ) საკმარის, სტაბილურ შემოსავლად ჩაეთვალოთ წელიწადში 15 000 ლარი,

ამასთან, დამატებით 4000 ლარი თვითოეულ კმაყოფაზე მყოფ პირზე.

ამ პირობების გათვალისწინებით ფინანსური მრჩეველი შემუშავებს კონკრეტული ინვესტორისათვის კონკრეტულ საფინანსო სტრატეგიას !!!

განვიხილოთ ამოცანის ამოხსნის სოგიერთი საინტერესო მომენტი.

ამოცანის არსიდან გამომდინარე, შესაძლებელია 3 ძირითადი მიზნის ჩამოყალიბება:

***** MIZNEBI *****

?-investicia(tanxa_bankshi). (1)

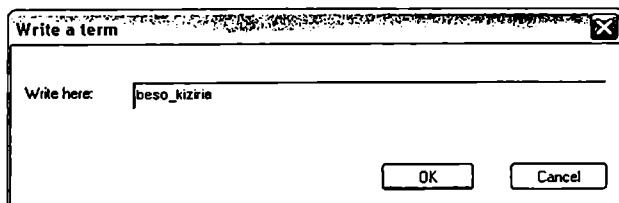
?-investicia(akcia). (2)

?-investicia(combinacia). (3)

პროლოგ-პროგრამას დაეუსახოთ პირველი მიზანი. მიზნის გააქტიურების შემდეგ,

შედეგების  **Output** ელში გამოდის შეტყობინება და ინფორმაციის შეტანის „Read“ ფანჯარა იხ. ნახ. 3.19.

GTXOVT, GVACNOBOT INVESTORIS VINAQBA !!!



ნახ. 3.19.

ინვესტორის ვინაობის შეტანის შემდეგ  **Output** ელში გამოდის შემდეგი შეტყობინება.

Ramdeni piria tqvens kmakofaze?

ეკრანზე გამოსულ საინფორმაციო ქეფანჯარაში შევიტანოთ ციფრი 3, რაც იმას ნიშნავს, რომ ინვესტორს ჰქაევს კმაყოფაზე მეოფი 3 პირი. თეორიული კურსიდან გაეისხნოთ, რომ იმიტატორი „Strawberry Prolog“-ი ინტერაქტიული იმიტატორია. ამოცანის ამოხსნა საჭიროების შემთხვევაში შესაძლებელია დიალოგურ რეჟიმში. ამ ამოცანაში სწორედ ამ შემთხვევასთან გვაქვს ადგილი.

Ra tanxa gagachniat bankshi?

 **Output** ელში გამოსულ ამ შეკითხვაზე ვპასუხობთ „Read“ ფანჯარაში თანხის ჩაწერით. მკითხველი რასაკვირველია მიხედება, რომ განსახილველი ამოცანა გამოგონილი ამოცანაა. ჩვენ ეახდენთ გარკვეული ფინანსური პროცესის მოდელირებას. ამოცანა

აღებულია განვითარებული ქვეყნის მაგალითზე. ამის გამო აქ აღწერილი პროცესი და მოყვანილი თანხები ჩვენთვის შესაძლებელია ცოტათი „გაუგებარიც“ კი იყოს.

პროლოგ-პროგრამაში „ფინანსური მრჩეველი“, ამოცანის შინაარსიდან გამომდინარე, ფორმირებულია 3 ძირითადი და 4 დამხმარე წესი, ამასთან მიზანი (მისი ტექსტი) შეგნებულად „ემთხვევა“ შესაბამისი წესის თავს. ასე მაგალითად, ჩვენს მიერ დასახული მიზანი: ?-investicia(tanxa_bankshi). „ემთხვევა“ პირველი წესის თავს:

%+++++++ CESI 1 ++++++

investicia(tanxa_bankshi) :- წესის თავია.

write(" GTXOVT, GVACNOBOT INVESTORIS VINAoba !!!"),

```
read(Investori),nl,nl, write(" I N V E S T O R I A "),
write(Investori),nl, nl,
write("Ramdeni piria tqvens kmakofaze?"),read(G_Y),nl,
write("-----"),nl,
write("Tqvens kmakofaze: "),write(G_Y),write(" piri"),nl,nl,
%-----
```

tanxa_bankshi(arasakmarisi)->

```
%-----
write("*****"),nl,
write("Tqven tanxa unda sheinaxot bankshi"),nl

else write("*****"),nl,
write("Tqven unda ganaxorcielot sxva saxis investicia"),nl,fail.
```

პ/ს „Read“ ფანჯარაში გამოყავს დიალოგის მომდევნო შეკითხვა:

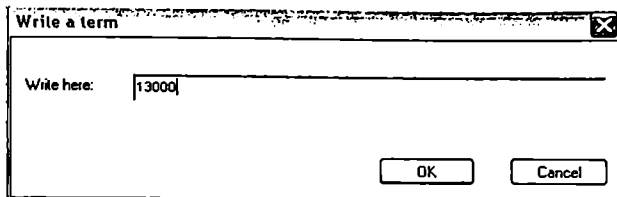
Ra tanxa gagachniat bankshi?

ამის შემდეგ პ/ს გადადის დამხმარე წესებზე. პროგრამაში გამოყენებულია როგორც ინტუქციური, ასევე დედუქციური პროდუქციული წესები. როგორც ზემოთ აღვნიშნეთ, ინტუქციურ წესებს საფუძვლად უდევს ლოგიკური ოპერაცია იმპლიკაცია (→). იმის შესაბამისად, თუ რა თანხას შეეიტანთ „Read“ ფანჯარაში, პროგრამაც წესებზე დაკრძნობით, „მიიღებს“ შესაბამის გადაწყვეტილებას და რეკომენდაციის სახით

Output ფანჯარაში მიაწვდის ინვესტორს.

თუ ინვესტორს არ გააჩნია საკმარისი თანხა ანგარიშზე, შესაბამისად:

$X < G_Y * 5000$, სადაც X ბანკის ანგარიშზე არსებული თანხაა, ხოლო G_Y კმაყოფაზე მყოფ პირთა რაოდენობა და ქვეფანჯარა „Read“-ში შევეიტანეთ თანხა 13 000 ლარი მივიღებთ: იხ. ნახ. 320.



ნახ. 320.

ძირითად და დამხმარე წესებზე დაყრდნობით ა/ს დასახული მიზნისათვის მიიღებს შემდეგ რეკომენდაციას იხ. ნახ. 321. :

```
GTXOVT, GVACNOBOT INVESTORIS VINAoba !!!
```

```
I N V E S T O R I A   besiki_kiziria
```

```
Ramdeni piria tqvens kmakofaze?
```

```
Tqvens kmakofazea: 3 piri
```

```
Ra tanka gagachniat bankshi?
```

```
Tqvens angarishzea:13000 lari
```

```
*****
Tqven tanxa unda sheinaxot bankshi
Yes.
```

ნახ. 321.

„ფინანსური მრჩეველის “ რეკომენდაცია ასე გამოიყურება:

Tqven tanxa unda sheinaxot bankshi.

პროლოგ-სისტემისათვის (ა/ს) სხვა მიზნის დასახვისას, პროგრამის მიერ გაიცემა შესაბამისი რეკომენდაცია.

მაგალითისათვის მოგვეყავს მელრე მიზანი:

?-investicia(akcia). (2)

Output ელში წარმოდგენილი ამოცანის პასუხი იქნება: იხ. ნახ. 322.

Compiling the file:

C:\Program Files\Strawberry Prolog\beso\FINANSURI MRCHEVELI.pro
0 errors, 0 warnings.

GTXOVT, GVACNOBOT INVESTORIS VINAoba !!!

Ramdeni piria Tqvens kmayofaze?

Tqvens kmayofazea: 3 piri.

Ra tanxaa tqvens angarishze?

Tqvens angarishzea: 30000 lari.

Tqveni shemosavali stabiluria ? (Y/N). YES.

Tqveni shemosavali stabiluria.

Tqveni mimdinare shemosavali ?

Tqveni mimdinare shemosavalia: 40000 lari.

Tqven shegidzliat tanxis dabandeba akciebshi !!!

Yes

ნახ. 3.22.

ანალოგიური წესით პასუხობს პ/ს ჩვენს მიერ დასახულ ყველა მიზანს. უფრო დაწვრილებით იხილეთ დანართი 2.

ამოცანა 3. საქართველოს რკინიგზა

ხელოვნური ინტელექტის (ხი) სისტემები გამოიყენება ადამიანის მოღვაწეობის მრავალ სფეროში. განსახილველი ამოცანა ეხება საქართველოს რკინიგზას. მეურნეობის ეს დარგი საკმაოდ რთულია და ჩვენი ამოცანა არ არის მისი სრული განხილვა. ეს არც შედის ჩვენს მიზნებში. ამოცანა ეხმარება პოტენციურ მგზავრს, გარკვეული ინფორმაციის ბაზაზე. მიიღოს სწორი გადაწყვეტილება მატარებლით მგზავრობის დაგეგმვის შესახებ. პროლოგ-პროგრამაში აღწერილია სამგზავრო მატარებლების მოძრაობის გრაფიკი, მატარებლის და ადგილის კატეგორია. მგზავრს, დაპროექტებული ცოდნის ბაზაზე (ცბ) დაყრდნობით, შესაძლებლობა აქვს ამოირჩიოს დანიშნულების სადგურამდე მატარებლის და ადგილის კატეგორია სასურველი თანხის ფარგლებში. ც/ბ-ში მოცემულია დასაველო საქართველოს ოთხი ძირითადი დანიშნულების სამგზავრო სადგური: ბათუმი, თბილისი, ფოთი და ზუგდიდი. საწყის პუნქტად შერჩეულია თბილისი. სამგზავრო მატარებლების კატეგორიები ამ წლის მონაცემებით იყოფა ოთხ ჯგუფად: სამგზავრო, საფირმო, დიზელ და ჩქარი ადგილების კატეგორიები შემდეგია: საერთო, პლაცკარტი, კუპე და რბილი.

მგზავრს შეუძლია მიიღოს საჭირო ინფორმაცია თუ მიზანში მიუთითებს:

- დანიშნულების სადგურის მითითებით იგი მიიღებს ინფორმაციას ველში, ყველა შესაძლო ვარიანტის სახით;
- მხოლოდ მატარებლის კატეგორიას;
- მხოლოდ ადგილის კატეგორიას;
- მხოლოდ სასურველ თანხას.

მგზავრს ასევე შეუძლია შეიტანოს რამოდენიმე მოთხოვნა ერთდროულად (მაგ. დანიშნულების სადგური და სასურველი თანხა).

პროგრამაში გამოყენებულია პროლოგის იმიტატორის ე.წ. გრაფიკული ოპერატორები (პრედიკატები). მათი მეშვეობით პროლოგ-პროგრამა ქმნის სამუშაო ქვეფანჯარას, რომელთანაც მუშაობს მგზავრი. ასეთი ფანჯრების შექმნა აადვილებს პროგრამასთან მომხმარებლის დიალოგს. ამასთან, ინტერფეისული კავშირის „მომხმარებელი-კომპიუტერი“ ეიზუალური მხარეც საკმაოდ მაღალი დონისაა. ჩვენ პროლოგის იმიტატორის ამ შესაძლებლობებს დაწერილებით ნაშრომის მეოთხე თავში განვიხილავთ. მრავალი მაგალითის ბაზაზე (იხ. თავი IV).

მოიყვანოთ ამოცანის ამოხსნის რამოდენიმე ფრაგმენტი. სრულად ამოცანის ტექსტს დაინტერესებული მკითხველი იხილავს დანართი 2-ში.

პროგრამულად, გრაფიკული პრედიკატებით შექმნილ სამუშაო ქვეფანჯარას აქვს სახე. იხ. ნახ. 3.23.

ნახ. 3.23.

მუშა ქვეფანჯრის (ნახ. 3.23.) მენიუს სტრუქტურაში, მენიუს კომპონენტზე: „Biletis Archeva“ მიმითებლის დაწკაპუნებით ჩამოიშლება ქვემენიუ, რომელიც შეიცავს შემდეგ პარამეტრებს. იხ. ნახ. 3.24.

Matareblebis Ganrigi

Gamosvla Bilebis Archivi Programis Shesaxe

Danishi Danishnulebis Adgili
 Danishnulebis adgili da Pasit
 Danishnulebis Adgili, Matareblebis Tipik da Pasit

Mati Yvela Probit

Adgilis Kategorია:

Fasi:

Damaxsovreba

ნახ. 324.

ქვემოთხსენიებულ კომპონენტებში: დანიშნულების ადგილით, დანიშნულების ადგილით და ფასით, დანიშნულების ადგილით, მატარებლის ტიპით და ფასით, ყველა პირობით.

ამოცანას დაეხმარება შემდეგი მიზანი: პოტენციურმა მგზავრმა მიუთითოს მისთვის სასურველი დანიშნულების ადგილი (მგზავრობის ბოლო პუნქტი). ამისათვის იგი მიმითებლით დააწკაპუნებს ამ პარამეტრზე. ეკრანზე გამოდის ინფორმაციის შეტანის „Read“ ფანჯარა შესაბამისი სათაურით. მგზავრი მასში აკრიფავს მგზავრობის ბოლო

პუნქტს, მაგ. ბათუმს. დადასტურების შემთხვევაში „OK“ ღილაკზე დაჭერით **Output** ელში გამოდის ბათუმისკენ მიმავალი ყველა მატარებლის ჩამონათვალი. იხ. ნახ. 325.

Shetyvanet Danishnulebis Adgili

Write here:

OK **Cancel**

Danishnulebis Adgili:	batumi
Matareblis Tipi:	dizeli
Adgilis Kategorია:	saerto
Sapasuri:	12.5
Danishnulebis Adgili:	batumi
Matareblis Tipi:	sapirmo
Adgilis Kategorია:	saerto
Sapasuri:	3.7
Danishnulebis Adgili:	batumi
Matareblis Tipi:	sapirmo
Adgilis Kategorია:	plackarti
Sapasuri:	8
Danishnulebis Adgili:	batumi
Matareblis Tipi:	sapirmo
Adgilis Kategorია:	kupe
Sapasuri:	13
Danishnulebis Adgili:	batumi
Matareblis Tipi:	sapirmo
Adgilis Kategorია:	rbili
Sapasuri:	20
Danishnulebis Adgili:	batumi
Matareblis Tipi:	sangzavro
Adgilis Kategorია:	saerto
Sapasuri:	3.4
Danishnulebis Adgili:	batumi
Matareblis Tipi:	sangzavro
Adgilis Kategorია:	plackarti
Sapasuri:	5.1
Danishnulebis Adgili:	batumi
Matareblis Tipi:	sangzavro
Adgilis Kategorია:	kupe
Sapasuri:	10.8
Danishnulebis Adgili:	batumi
Matareblis Tipi:	sangzavro
Adgilis Kategorია:	rbili
Sapasuri:	15

ნახ. 325.

ქვემოთ მეთრე პარამეტრსე დაცაუნებოთ პროლოგ-პროგრამას ესახათ შიზანს: დანიშნულების ადგილი ისეე ბათუში, ამასთან მგზავრი დამატებით იძლევა ინფორმაციას ბილეთის მისთვის სასურველი (მისაწვდომი) ღირებულების შესახებ. შედეგი იხილეთ ნახ. 326.-ზე.

Sheiyvanet Danishnulebis Adgili

Write here:
batumi

OK
Cancel

Sheiyvanet Tanxa

Write here: 10

OK Cancel

```

+++++
+ PASUXI +
+++++

Danishnulebis Adgili: batumi
Matareblis Tipi: sapirao
Adgilis Kategorია: saerto
Sapasuri: 3.7

Danishnulebis Adgili: batumi
Matareblis Tipi: sapirao
Adgilis Kategorია: plackarti
Sapasuri: 8

Danishnulebis Adgili: batumi
Matareblis Tipi: sangzavro
Adgilis Kategorია: saerto
Sapasuri: 3.4

Danishnulebis Adgili: batumi
Matareblis Tipi: sangzavro
Adgilis Kategorია: plackarti
Sapasuri: 5.1

```

ნახ. 326.

პროლოგ-პროგრამას დაეუსახოთ „ყველაზე“ რთული მიზანი. მხედველობაში მიიღოს მგზავრის მიერ შეთავაზებული ყველა პარამეტრის მნიშვნელობა და მის ბაზაზე „მიიღოს“ მმართველი გადაწყვეტილება. ეს პროცესი ასახულია ნახ. 327-ზე.

Sheiyvanet Danishnulebis Adgili

Write here: batumi

OK Cancel

Sheiyvanet Matareblis Tipi

Write here: sapirao

OK Cancel

Sheiyvanet Adgilis Kategorია

Write here: rbili

OK Cancel

Sheiyvanet Tanxa

Write here: 20

OK Cancel

```

+++++
+          PASUXI          +
+++++

Danishnulebis Adgili: batumi
Matareblis Tipi: sapirao
Adgilis Kategorია: rbili
Sapasuri: 20

```

ნახ. 327.

ანალოგიური ტექნოლოგიით შესაძლებელია მივიღოთ პასუხი მგზავრისათვის აუცილებელ ყველა შეკითხვაზე. ამასთან ერთად გაეითვალისწინოთ როგორც სუბიექტური ფაქტორები (მგზავრის მიერ შერჩეული საბოლოო დანიშნულების პუნქტი, მატარებლის

ტ.პ.ი. გეზაურობის ღირებულება და სხვა). ასევე **ობიექტური ფაქტორები** (საქართველოს რკინიგზის მატარებლების პარკი, დღეისათვის არსებული ბილეთის ღირებულება და სხვა).

ამოცანაში ასევე შესაძლებელია ც/ბ-ში ახალი ინფორმაციის შეტანა, ძირითადი მუშა ქვეფარჯის (ნახ. 3.23.) მეშვეობით. ფანჯრის სპეციალურ ტესტ-ბოქსებში შეგვაქვს მონაცემები ახალი მარშრუტების დანიშვნის შესახებ, ფასების რეგულირების შესახებ და ა.შ. შემდეგ ბოჯზე ვაჭერთ ღილაკს „დამახსოვრება“. ახალი ინფორმაცია ჩაიწერება ც/ბ-ში.

ამოცანა 4. ქ. თბილისის სამარშრუტო ტაქსები

განრიგებისა და მარშრუტების შედგენის მათემატიკური მეთოდების შემუშავებას დიდი უურადლება ექცევა ხი-ის ისეთი მიმართულების ამოცანებში, როგორიცაა „რობოტები და მანიპულიატორები“. ჩვენ ამ საკითხებს ნაშრომის მეორე ტომში შევეხებით. ახლა კი განვიხილოთ „შედარებით“ მარტივი ამოცანა, რომელიც თავისი შინაარსით მეტად აქტუალურია.

ამოცანის ტანში ჩაშენებულ რელაციურ ცოდნის ბაზაში მოთავსებულია ინფორმაცია ქ. თბილისში მოძრავი სამარშრუტო მიკროავტობუსების შესახებ: საწყისი და საბოლოო პუნქტები, ამ პუნქტებს შორის მოძრავი მიკროავტობუსების სელაგეზის ნომერი და სხვა. ამოცანის დანიშნულებაა გეზაურს მიაწოდოს ინფორმაცია ყველა იმ სელაგეზის ნომრის შესახებ, რომელიც მოძრაობს გარკვეულ პუნქტებს შორის. ინფორმაციის ძებნისა და დაშვებების საკითხები ეფუძნება პროლოგ-პროგრამაში მოცემულ ოთხ პროდუქციულ წესს, რომელშიც გათვალისწინებულია თითქმის ყველა შესაძლო ვარიანტი საძიებელი მარშრუტის სელაგეზის ნომრების შესახებ. აღნიშნული წესები მოყვანილია ნახ. 3.28-ზე:

*****C E S E B I*****

marshuti(A, B, N) :-
 marshutka(N, A, _),
 marshutka(N, B, _).

marshuti(A, B, N) :-
 marshutka(N, _ , A),
 marshutka(N, _ , B).

marshuti(A, B, N) :-
 marshutka(N, A, _),
 marshutka(N, _ , B).

marshuti(A, B, N) :-
 marshutka(N, _ , A),
 marshutka(N, B, _).

ნახ. 3.28.

ამოცანის სრული ტექსტი მოცემულია დანართი 2-ში.

დაეუსახოთ პროლოგ-პროგრამას შემდეგი მიზანი: ეიპოვოთ ორ პუნქტს „თავისუფლების მოედანი“ და „ევა ფშაველას გამზირი“ შორის მოძრავი სამარშრუტო მიკროავტობუსების სელაგეზების ნომრები. მიზანი იხილეთ ნახ. 329.

```
?-write("*****"),nl,fail.
```

```
?- A1="tavisuflebis moedani", A2="vaja pshavelas gamz.", write(A1),  
write("-"),write(A2),nl,write("dadis marshutkebi:"),  
nl,marshuti(A1,A2,A),write("N:"),write(A),nl,fail.
```

```
?-write("====="),nl,fail.
```

ნახ. 329.

პასუხი დასახულ მისანზე გამოდის  ველში. იხ. ნახ. 330.

Compiling the file:

C:\Program Files\Strawberry Prolog\beso\marshrutka.pro

0 errors, 0 warnings.

tavisuflebis moedani-vaja pshavelas gamz.

dadis marshutkebi:

N:2

N:2a

No.

ნახ. 330.

მოცემული ამოცანის ც/ბ-ში შესაძლებელია ქთბილისის მიკროავტობუსების ყველა სელაგეზის შესახებ ინფორმაციის შეტანა. მოცემულ მომენტში ეს ჩვენს მიზანს არ წარმოადგენს. განხილული ამოცანა სასწავლო ამოცანაა და მისი დანიშნულებაც ესაა.

მარშრუტებისა და განრიგების შედგენის გარდა პროლოგში ასევე შესაძლებელია მომხმარებლისათვის სხვადასხვა სახის მენიუების შედგენა. განვიხილოთ ამოცანა, რომელიც შედგენილია ჩვენი სპეციალობის სტუდენტის მიერ და ავტორთა ჯგუფმა მისანშეწონილად ჩათვალა მათი ორგანიზაციის გამო ნაშრომში მისი ჩართვა.

ამოცანა 5 კომპიუტერის კონფიგურაციისა და ღირებულების შერჩევა მყიდველის მოთხოვნათა შესაბამისად

ამოცანა დაიწერა რამდენიმე წლის წინათ, რის გამოც პროგრამის ც/ბ-ში მოთავსებული კომპიუტერის კომპონენტები და შესაბამისად მათი ფასებიც ცოტათი მოჰყვებოდა. საჭიროების შემთხვევაში მათი განახლება არაერთად სიმძნელეს არ წარმოადგენს. მაგრამ ეს ფაქტორი ჩვენი ასრით გადამწყვეტი არ არის. მთავარი თვისება ამოცანაში ჩადებული იდეაა.

პროგრამის ც/ბ-ში მოთავსებულია კომპიუტერის ძირითადი მოწყობილობების ნაწილის: პროცესორის, დედა პლატის, მონიტორისა და ეინფესტერის სხვადასხვა სახეობების ჩამონათვალი შესაბამისი ფასის ჩვენებით. ც/ბ-ში ჩადებული ინფორმაცია რეალურია და შესაბამისად პროგრამის დაწერის დროს. ასე მაგალითად პროცესორები. იხ. ნახ. 331.:

-----C P U-----PRACE(S)-----

cpu("pentium233 MMX", 111).

cpu("pentiumII 400 MMX", 712).

cpu("pentiumII 350 MMX", 523).

cpu("pentiumII 266 MMX", 201).

cpu("pentium 233 MMX box", 112).

cpu("pentiumII 333 MMX ", 413).

ნახ. 331.

ამოცანის შინაარსი ასეთია. რომელიმე კომპიუტერების ამწყობ ფირმაში (ასეთი ფირმები დღეს თბილისში მრავლადაა, რომლებიც აწყობენ და ჰყიდნი კომპიუტერებს) მოდის მყიდველი, რომელსაც გააჩნია გარკვეული თანხა და სურვილი აქვს შეიძინოს მისთვის სასურველი კონფიგურაციის კომპიუტერი. პროლოგ-პროგრამა მას სთავაზობს კომპიუტერის სხვადასხვა კონფიგურაციებს. ამასთან მყიდველს აქვს შესაძლებლობა წინასწარ გაეცნოს ფირმაში არსებული კომპიუტერის ცალკეული კომპონენტების ჩამონათვალს და მათ პარამეტრებს. ასე მაგალითად, ნახ. 331.-ზე წარმოდგენილია ფირმაში არსებული პროცესორების ტიპები და მათი ფასები.

პროლოგ-პროგრამას მყიდველმა შესაძლებელია დაუსახოს შემდეგი მიზანი: ინფორმაციის მიწოდება ფირმაში არსებული ყველა მონიტორების შესახებ. მიზანი პროგრამულად შესაძლებელია ასე ჩამოყალიბდეს. იხ. ნახ. 332.:

?-write(" MONITOR "),nl,

```
write("*****"),
```

```
nl,nl,monitor(Name,_),write(Name),nl,nl,fail.
```

ნახ. 332.

 Output

ამ შეკითხვას (მიზანზე) პროლოგ-სისტემა (პ/ს) ექვლში გამოიტანს შემდეგ შეტყობინებას. იხ. ნახ. 333.

Compiling the file:

C:\Program Files\Strawberry Prolog\beso\computer.pro

0 errors, 0 warnings.

MONITOR

panasonic 15

panasonic 17

panasonic 17pf

view sonic 17

No.

ნახ. 333

პ/ს-ს ასევე შესაძლებელია დაეუხახოთ „შედარებით რთული“ მიზანი. ეთქვას, მკიდეგლს გააჩნია 510 დოლარი.

?- computer(510, F), fail.

 Output

ამ შეკითხვას ფანჯარაში გამოდის შემდეგი ინფორმაცია. იხ. ნახ. 334.

COMPUTERIS PASI=502 \$

+++++

C P U pentium233 MMX price=111 \$

H.D. fujitsu 2.1 Gb price=111 \$

M.B. aristo am price=60 \$

MONITOR panasonic 15 price=220 \$

COMPUTERIS PASI=503 \$

+++++

C P U pentium 233 MMX box price=112 \$

H.D. fujitsu 2.1 Gb price=111 \$

M.B. aristo am price=60 \$

MONITOR panasonic 15 price=220 \$

No.

ნახ. 334.

პ.ს-მ 510 დოლარის ღირებულებით მიიღო კომპიუტერის კომპლექტის ორი ვარიანტი. მიიღეული ამ ორი ვარიანტიდან აირჩევს ერთერთს და შეისყიდის მისთვის სასურველ კომპიუტერს.

აღეული მისახედეია, რომ რაც მეტია თანხა, მით მეტია ასარჩევი ვარიანტების რაოდენობა. ასე მაგალითად, 550 დოლარის შემთხვევაში:

?- computer(550, T), fail.

ვარიანტების რაოდენობა გაცილებით მეტი იქნება, იხ. ნახ. 335.

COMPUTERIS PASI=502 \$

+++++

C P U pentium233 MMX price=111 \$

H.D. fujitsu 2.1 Gb price=111 \$

M.B. aristo am price=60 \$

MONITOR panasonic 15 price=220 \$

COMPUTERIS PASI=532 \$

+++++

C P U pentium233 MMX price=111 \$

H.D. fujitsu 2.1 Gb price=111 \$

M.B. mycomp a15 price=90 \$

MONITOR panasonic 15 price=220 \$

COMPUTERIS PASI=511 \$

+++++

C P U pentium233 MMX price=111 \$

H.D. fujitsu 3.2 Gb price=120 \$

M.B. aristo am price=60 \$

MONITOR panasonic 15 price=220 \$

COMPUTERIS PASI=541 \$

+++++

C P U pentium233 MMX price=111 \$

H.D. fujitsu 3.2 Gb price=120 \$

M.B. mycomp a15 price=90 \$

MONITOR panasonic 15 price=220 \$

COMPUTERIS PASI=534 \$

+++++

C P U pentium233 MMX price=111 \$

H.D. quantum 3.2 Gb price=143 \$

M.B. aristo am price=60 \$

MONITOR panasonic 15 price=220 \$ და ა.შ.

.....

ნახ. 3.35.

პროლოგ-სისტემა ამოცანის ამოხსნისას ასრულებს „საკმაოდ დიდ“ სამუშაოს. იგი ეფუძნება ხი-ის სისტემების ბევრი ამოცანის ამოხსნისას მიღებულ, ე.წ. „ვარიანტების სრული გადარჩევის“ მეთოდს. ძეგნის პროცესი მიმდინარეობს ზევიდან ქვემოთ (ძეგნა სიღრმეში) და მარცხნიდან მარჯვნივ.

ამ მეთოდის რეალიზაციისათვის პროგრამაში შემუშავებულია „პროდუქციული წესი“, რომელიც მოქმედებს ნახ. 3.36-ზე.

*****Rules*****

computer(X, Y) :-

cpu(A, B),

h_d(C, D),

m_b(G, H),

monitor(I, J),

$X > B + D + H + J, Y = B + D + H + J,$

```

write( "  COMPUTERIS PASI=" ),
write( Y ), write( " $" ), nl,
write( "+++++++" ), nl, nl,
write( "C P U" ), write( "  " ), write( A ),
write( " price=" ), write( B ), write( " $" ), nl, nl,
write( "H.D." ), write( "  " ), write( C ),
write( " price=" ), write( D ), write( " $" ), nl, nl,
write( "M.B." ), write( "  " ), write( G ), *+
write( " price=" ), write( H ), write( " $" ), nl, nl,
write( "MONITOR" ), write( "  " ), write( I ),
write( " price=" ), write( J ), write( " $" ), nl,
write( "*****" ),
nl, nl, nl.

```

ნახ. 336.

წესის თავში პრედიკატი „computer“ ბინარული (ორ არგუმენტიანი) პრედიკატია, რომლის პირველი არგუმენტი X მყიდველის მიერ შემოთავაზებული თანხაა, ხოლო მეორე არგუმენტი Y კომპიუტერის მიერ შერჩეული ერთერთი კომფიგურაციის ფასია. წესში გამოყენებულია მარტივი ლოგიკური პირობა, რომელიც უზრუნველყოფს ძებნის პროცესის სწორი მიმართულებით წარმართვას. იღვამარტივია, მყიდველის ხელზე არსებული თანხა X მეტი უნდა იყოს კომპიუტერის კომპონენტების ღირებულების ჯამზე.

$X > B + D + H + J$. ვარიანტების გადარჩევისას პ/ს ასეთი ვარიანტის „მოძებნისას“, Y ცელაღში აფორმირებს კომპიუტერის ამ კონფიგურაციის ჯამურ ფასს და სხვა

მონაცემებთან ერთად გამოაქვს  **Output** ველში, მაგ-დ იხ. ნახ 335.

COMPUTERIS PASI=511 \$

+++++

ვარიანტების ძებნის „სრული გადარჩევის მეთოდი“ კომპიუტერისთვისაც კი „შრომატევადი“ მეთოდია, განსაკუთრებით როცა გადასარჩევი ვარიანტების რიცხვი დიდია. ხოის ამოცანებში გადარჩევის პროცესის „გასამარტივებლად“ გამოიყენება ვ.წ. „ვერისტიკული მეთოდი“, რომელიც გმპ-ის ვერისტიკებს ეფუძნება. ადამიანი გადაწყვეტილებების მიღებისას ეფუძნება „ბევრ ვერისტიკას“, რომელიც თავის მხრივ ინდივიდის ცოდნაზე, გამოცდილებაზე, ინტუიციაზე და ადამიანის აზროვნების სხვა მრავალ ფაქტორზეა დამოკიდებული. რაიმე პრობლემის გადაწყვეტისას, გმპ თავის

ინდივიდუალურ ფაქტორებს გარდაქმნის სხვადასხვა ფორმალისებში. ამის მარტივი, მაგრამ ნათელი მაგალითია მოცემულ ამოცანაში პროგრამისტის მიერ გადასარჩევი ვარიანტების შემცირების მიზნით, ზემოთ აღწერილი ლოგიკური პირობის შემოტანა ($X > B + D + H + J$). ამით ძეგლის პროცესი შედარებით „იოლღება“.

დანართი 2-ში ასევე მოყვანილია „მენიუს“ შედგენის მეორე ნიშნულში, პროლოგ-პროგრამის სრული ტექსტის სახით. იგი შეეხება „სტუდენტური ვახშამის“ მენიუმის შედგენას და ჩვენი აზრით საინტერესო ამოცანაა. დაინტერესებულ მკითხველს შესაძლებლობა აქვს გაეცნოს მას და გამოიყენოს იგი ლაბორატორიული სამუშაოსათვის.

4.1. პროლოგ-პროგრამების შედგენა ე.წ. „კლასიკური“ და გრაფიკული ოპერატორების გაყოფნებით. ამ მეთოდების შედარებითი ანალიზი.

„Strawberry Prolog“ იმიტატორს სხვა მრავალ შესაძლებლობებთან ერთად, რომლებიც წინა თავებში განვიხილეთ, გააჩნია უნარი სპეციალური გრაფიკული ოპერატორების გამოყენებით შექმნას სხვადასხვა დანიშნულებისა და შინაარსის ფანჯრები. მათი მეშვეობით მომხმარებელს ეძლევა საშუალება მოათავსოს ამ ფანჯრებში ამოცანის ამონახსნის შედეგები: რელაციური (მაშენებელი) ბაზების კომპონენტები, დაურთოს ამოცანის შედეგებს ფოტო და ხმოვანი ინფორმაცია და ა.შ. ყოველივე ეს ასახულია ქვემოთ განხილულ ამოცანებში. ამასთან დაცულია [7] სახელმძღვანელოში მოყვანილი პრინციპი: ამოცანათა კრებულში შესულია ხი-ის ე.წ. „კლასიკური“ ამოცანები, ნაშრომის ავტორთა ჯგუფის მიერ დასმული და გადაწყვეტილი ამოცანები, ასევე მრავალი წლის განმავლობაში ჩვენი სპეციალობის სტუდენტებისა და მაგისტრების მიერ შედგენილი საუკეთესო ამოცანების ნიმუშები.

გრაფიკული ოპერატორების მოქმედების არსის უკეთ გასაგებად, განვიხილოთ მეორე თავში წარმოდგენილი ამოცანა ავტომობილების ცოდნის ბაზის შესახებ, რომელიც შესაძლებელია გამოყენებული იქნეს პირველი სემესტრის N2 ლაბორატორიულ სამუშაოდ. მოვიყვანოთ ამოცანის პროლოგ-პროგრამის ამონაბეჭდი.

%***** PROGRAMMA-KATALOG C A R.*****

%***** G O A L !!! *****

```
?-write(" AVTOMANCANIS MODELEBI:"),nl,
write("*****"),nl,
car(K,_,_,_),write(K),nl,fail.
```

```
/*?-write(" 1989 clis automobilebia: "),nl,
write("*****"),nl,nl,
car(Name,1989,_,_,_),write(Name),nl,fail.*
```

```
/*?-write(" 1990 clis automobilebia: "),nl,
write("*****"),nl,nl,
car(Name,1990,_,_,_),write(Name),nl,fail.*
```

```
%?-car(pontiac,1989,_,_,_).
```

```
%?-car(opel,1989,_,_,_).
```

```
/*?-write(" G E R M A N Y:"),nl,
write("*****"),nl,nl,
car(Z,_,_,germany,_,_),write(Z),nl,nl,fail.*
```

```

/*?-write("  U S A:"),nl,
write("*****"),nl,nl,
car(Z,__,usa,_),write(Z),nl,nl,fail.*/

/*?-write("  J A P A N:"),nl,
write("*****"),nl,nl,
car(Z,__,japan,_),write(Z),nl,nl,fail.*/

/*?-write("  C O U N T R Y :"),nl,
write("*****"),nl,nl,
car(__,_,V,_),write(V),nl,nl,fail.*/

/*?-write("  C I L.=6: "),nl,
write("@@@@@@@@@@@@@@@@@@@@@@@@@@"),
nl,nl,
car(A,_6,_),write(A),nl,nl,fail.*/

/*?-write("  C I L.=8: "),nl,
write("*****"),nl,nl,
car(B,_8,_),write(B),nl,nl,fail.*/

%?-car(audi,_8,_).

%?-car(pontiac,__,usa,_).

%?-car(pontiac,__,georgia,_).

/*?-car(pontiac,__,_,H),
write(" P O N T I A C-(km/h.)"),
nl,nl,write(H),nl,nl.*/

/*?-car(bmw,__,_,J,_),
write(" B M W-(HP.)"),
nl,nl,write(J),nl,nl.*/

/*?-car(Name,__,_,W,_),write(Name),write("  "),
write(W),write(" HP."),nl,nl,fail.*/

/*?-car(Name,__,_,R),write(Name),write("  "),
write(R),write(" km/h."),nl,nl,fail.*/

/*?-write("  G O O D C A R: "),nl,
write("*****"),nl,nl,
good_car(V),write(V),nl,nl,fail.*/

/*?-write("  B A D C A R: "),nl,
write("*****"),nl,nl,
bad_car(W),write(W),nl,nl,fail.*/

%@@@@@@@@ CODNIC BAZA @@@@@@@@@@

%***** F A K T E B I: *****

```

```

car(pontiac,1989,6,usa,208,215).
car(opel,1987,6,germany,177,225).
car(audi,1988,8,germany,250,235).
car(volvo,1988,4,sweden,120,200).
car(bmw,1986,6,germany,211,230).
car(ford,1988,4,usa,147,200).
car(cadillac,1989,8,usa,160,185).
car(honda,1990,6,japan,280,270).
car(fiat,1988,4,italy,83,170).
car(toyota,1990,4,japan,156,220).
car(mercedes_benz,1989,8,germany,326,250).
car(porsche,1989,8,germany,330,275).
car(ferrari,1989,8,italy,300,255).
car(lincoln,1989,6,usa,182,185).
car(mitsubishi,1988,4,japan,95,180).
car(chrysler,1989,6,usa,143,195).
car(jaguar,1990,6,britain,507,322).
car(alfa_romeo,1989,6,italy,190,230).
car(jeep,1989,6,usa,180,175).

%***** C E S E B I: *****

good_car(X):-car(X,_Y,_,S),Y>=6,S>250.

bad_car(A):-car(A,_B,_,F),B<6,F=<200.

%***** 26 aprili 2004 celi.*****

```

მოკლედ გაეიხსენოთ მეორე თავში მოყვანილი ამ ამოცანის აღწერა. ამოცანის არსი შემდეგია. მოცემულია ავტომობილების რელაციური ცოდნის ბაზაში (ც/ბ) შემოტანილია „რედიკატი „car“ (ავტომობილი). თვითოეული ობიექტი (ავტომობილი) აღწერილია 6

პარამეტრით: ავტომობილის ტიპი, გამოშვების წელი, ძრავში ცილინდრების რაოდენობა, დამრთავი ქვეყანა, ავტომობილის სიმძლავრე ც/ძ-ში და მისი მაქსიმალური სიჩქარე. ხი-ს თოვრიული კურსიდან ვიცით, რომ ობიექტების აღწერა შესაძლებელია ნებისმიერი რაოდენობის კომპონენტით. ობიექტი (ავტომობილი) შესაძლებელი იყო აგვეწერა ვთქვათ 20 პარამეტრით და ც/ძ-ში შეგვეტანა ინფორმაცია 100 ავტომობილის შესახებ. ყველაფერი ეს დამოკიდებულია ამ პროგრამის მომხმარებლის მიზანზე და მისი ინფორმაციულობის ხარისხზე.

განსახილველი ამოცანის ცოდნის ბაზა ასევე შეიცავს 2 წესს. ც/ბ-ზე დაყრდნობით მომხმარებელი აყალიბებს მიზნებს (შეკითხვებს) პროლოგ-სისტემისადმი (პ/ს). ამოცანის ტექსტში მიზნები მოყვანილია პროგრამის თავში (იხ. ტექსტის ამონაბეჭდი). მიზნების ჩამორთვალი ასევე შესაძლებელი იყო მოგვეთავსებინა პროგრამის ბოლოში. ამით არაფერი არ შეიცვლებოდა. (ეს მომენტი გასათვალისწინებელია ნებისმიერ პროლოგ-პროგრამის შედგენისას).

როგორც ზემოთ აღინიშნა, მომხმარებელი თავის შეკითხვებს (მიზნებს) წინასწარ აყალიბებს და ათავსებს მათ სტრიქონულ ან აბზაცურ კომენტარში. გამომდინარე დასახული მიზნიდან, მომხმარებელი ხსნის კომენტარის ნიშნებს (% ან /* */) მიზნებზე. ამით მიზანი (შეკითხვა) აქტიურდება და იგი გადადის პროლოგ-პროგრამის ტანში. პ/ს იწყებს პასუხის ძებნას. მოყვანილი პროგრამის ამონაბეჭდში გააქტიურებულია პირველივე მიზანი (ასევე შესაძლებელია ნებისმიერი სხვა მიზნის გააქტიურებაც). კითხვის მიზანია

გამოვითანოთ შედეგების Output ქვეფანჯარაში ც/ბ-ში მოთავსებული ყველა ავტომობილის დასახელება. მოვიყვანოთ ამ კითხაზე პასუხის ამონაბეჭდი.

```
Compiling the file:
C:\Program Files\Strawberry Prolog\beso\car.pro
0 errors, 0 warnings.
```

```
AVTOMANCANIS MODELEBI:
*****
pontiac
opel
audi
volvo
bmw
ford
cadillac
honda
fiat
toyota
mercedes_benz
porsche
ferrari
lincoln
mitsubishi
chrysler
jaguar
alfa_romeo
jeep
No.
```

შედგენს ამონაბეჭდის მიხედვით, კომპილატორმა მოახდინა C დისკზე მოთავსებული „Program Files“ დირექტორიაში მოთავსებული ქვედირექტორიის „Strawberry Prolog“ –ში ფაილის „car.pro“-ს კომპილაცია. სინტაქსური შეცდომები და გაფრთხილებები არ არის: (0 errors , 0 warnings). ც/ბ-ში მოიძებნა ჩამოთვლილი ავტომობილები. ჩამონათვალის ბოლოს „No“ ნიშნავს, რომ ც/ბ-ში მეტი ავტომობილი არ არის.

განვიხილოთ ამავე ამოცანის ნებისმიერი სხვა მიზანი (კითხვა). ამისათვის უპირველეს ყოვლისა უნდა მოვახდინოთ წინა შეკითხვის კომენტარში რეჟიმში გადაყვანა (ჩავესათ შეკითხვა კომენტარის სპეც. სიმბოლოებში). ახალ შეკითხვას აქვს სახე:

```
/*?-write(" U S A:"),nl,
write("*****"),nl,nl,
car(Z,_usa_,),write(Z),nl,nl,fail.*
```

პ/ს-ს ეპილეტ დაეძახება: მოძებნოს ც/ბ-ში ამერიკული წარმოების ყველა ავტომობილი. კითხვაზე პასუხის ამონაბეჭდს აქვს სახე:

```
U S A:
*****
pontiac
ford
cadillac
lincoln
chrysler
jeep
No
|
```

ასევე შესაძლებელია მოყვანილ პროგრამაში ყველა შეკითხვაზე პასუხის მიღება იმავე ტექნოლოგიით.

დავისახოთ შემდეგი მიზანი: გამოვეყენოთ პროლოგ-იმიტატორის გრაფიკული შესაძლებლობები და ჩავეწეროთ ზემოთ განხილული ამოცანა განსხვავებული ფორმით.

ამოცანის პროლოგ-პროგრამის ამონაბეჭდს (გრაფიკული ოპერატორების ბაზაზე) აქვს შემდეგი სახე:

```
?-
window(,_,win_func(), "base of car",200, 100, 400, 400).
```

```
win_func(init):-
    menu(pop_up,_, menu_window(),"file"),
    menu(pop_up,_, menu_go(),"Car"),
    menu(normal,_, menu_about(),"About"),
```

```
edit(G_E1,_, edit_func(), "" ,150, 10, 100, 25),
edit(G_E2,_, edit_func(), "" ,150, 50, 100, 25),
```

```

edit(G_E3,_, edit_func(), "", 150, 100,100, 25),
edit(G_E4,_, edit_func(), "", 150, 150,100, 25),
edit(G_E5,_, edit_func(), "", 150, 200,100, 25),
edit(G_E6,_, edit_func(), "", 150, 250,100, 25),

```

```

static(_,_, fa(), "Name:", 20, 10,100, 20),
static(_,_, fa(), "Year:", 20, 50,100, 20),
static(_,_, fa(), "Cilindr:", 20, 100,100, 20),
static(_,_, fa(), "Country:", 20, 150,100, 20),
static(_,_, fa(), "Hp:", 20, 200,100, 20),
static(_,_, fa(), "Km/h:", 20, 250,100, 20),

```

```

button(_,_,bfunc(),"Record",150,300,70,25).

```

```

win_func(close) :-not(yes_no("Exit","Confirm Exit?", !)).

```

```

menu_exit(press) :- close_window(_).

```

```

menu_window(init):-menu(normal,_,_,menu_exit(),"Exit").

```

```

menu_about(press):-message("About", "This program is made by Georgi Zakalashvili |22
November 2004|",!).

```

```

menu_go(init):-
    menu(normal,_,_,s1(),"Name"),
    menu(normal,_,_,s2(),"Year"),
    menu(normal,_,_,s3(),"Cilindr"),
    menu(normal,_,_,s4(),"Country"),
    menu(normal,_,_,s5(),"Hp"),
    menu(normal,_,_,s6(),"Km/h").

```

```

s1(press):-
    read(X, "Car Name",s),nl,
write("NAME"),write("  "),
write("YEAR"),write("  "),
write("CILINDR"),write("  "),
write("COUNTRY"),write("  "),
write("HP"),write("  "),
write("KM/H"),nl,nl,
car(X,Y,Z,A,B,C),
write(X),write("  "),
write(Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),
write(C),nl,fail.

```

```

s2(press):-
    read(Y, "Car Year",s),nl,
write("NAME"),write("  "),
write("YEAR"),write("  "),
write("CILINDR"),write("  "),

```

```

write("COUNTRY"),write("  "),
write("HP"),write("  "),
write("KM/H"),nl,nl,
car(X,Y,Z,A,B,C),
write(X),write("  "),
write(Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),
write(C),nl,fail.

```

s3(press):-

```

read(Z, "Car Cilindr",s),nl,
write("NAME"),write("  "),
write("YEAR"),write("  "),
write("CILINDR"),write("  "),
write("COUNTRY"),write("  "),
write("HP"),write("  "),
write("KM/H"),nl,nl,
car(X,Y,Z,A,B,C),
write(X),write("  "),
write(Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),
write(C),nl,fail.

```

s4(press):-

```

read(A, "Car Country",s),nl,
write("NAME"),write("  "),
write("YEAR"),write("  "),
write("CILINDR"),write("  "),
write("COUNTRY"),write("  "),
write("HP"),write("  "),
write("KM/H"),nl,nl,
car(X,Y,Z,A,B,C),
write(X),write("  "),
write(Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),
write(C),nl,fail.

```

s5(press):-

```

read(B, "Car HP",s),nl,
write("NAME"),write("  "),
write("YEAR"),write("  "),
write("CILINDR"),write("  "),
write("COUNTRY"),write("  "),
write("HP"),write("  "),
write("KM/H"),nl,nl,
car(X,Y,Z,A,B,C),

```

```

write(X),write("  "),
write(Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),
write(C),nl,fail.

```

s6(press):-

```

read(C, "Car Kilometr/hour",s),nl,
write("NAME"),write("  "),
write("YEAR"),write("  "),
write("CILINDR"),write("  "),
write("COUNTRY"),write("  "),
write("HP"),write("  "),
write("KM/H"),nl,nl,
car(X,Y,Z,A,B,C),
write(X),write("  "),
write(Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),
write(C),nl,fail.

```

bfunc(press):-

```

X is get_text(G_E1),
Y is get_text(G_E2),
Z is get_text(G_E3),
B is get_text(G_E4),
C is get_text(G_E5),
D is get_text(G_E6),
assert_in(car(X,Y,Z,B,C,D)),
message("Thank", "Thank you for record our Data Base!",),

```

```

set_text("",G_E1),
set_text("",G_E2),
set_text("",G_E3),
set_text("",G_E4),
set_text("",G_E5),
set_text("",G_E6).

```

%*****P A K T E B I*****

```

car("pontiac","1989","6","usa","208","215").
car("opel","1988","6","germany","177","225").
car("audi","1988","8","germany","250","200").
car("volvo","1988","4","sweden","120","200").
car("bmw","1986","6","germany","211","230").
car("ford","1988","4","usa","147","200").
car("cadillac","1989","8","usa","160","185").
car("honda","1990","6","japan","280","270").
car("fiat","1988","italy","83","170").

```

```

car("toyota", "1990", "4", "japan", "156", "220").
car("mersedes_benz", "1989", "8", "germany", "326", "250").
car("porsche", "1989", "8", "germany", "330", "275").
car("ferreri", "1989", "8", "italy", "300", "255").
car("lincoln", "1989", "6", "usa", "182", "185").
car("mitsubishi", "1988", "4", "japan", "95", "180").
car("chreysler", "1989", "6", "usa", "143", "195").
car("jaguar", "1990", "6", "britan", "507", "322").
car("alfa_romeo", "1989", "6", "italy", "190", "230").
car("jeep", "1989", "6", "usa", "180", "175").
car("opel", "1989", "6", "germany", "177", "225").
car("lincoln", "1992", "8", "usa", "200", "220").
car("ford", "1993", "8", "usa", "150", "220").
car("jeep", "1993", "6", "usa", "150", "200").
car("bmw", "1995", "8", "germany", "300", "280").
car("jaguar", "1995", "8", "britan", "550", "350").
car("toyota", "1995", "6", "japan", "254", "300").
car("pontiac", "1991", "8", "usa", "220", "220").
car("alfaromeo", "1995", "8", "italy", "250", "300").
car("linkoln", "2000", "12", "usa", "225", "300").

```

გაეცნოთ „Strawberry Prolog“ იმიტატორის ახალ ფუნქციებს–გრაფიკულ ოპერატორებს, მოცემული მაგალითის ბაზაზე.

პროგრამა იწყება შემდეგი მიხნით:

?- window(_ , win_func(_), "Base of car", 200, 100, 400, 400).

window გრაფიკული ოპერატორია (იმიტატორში ჩაშენებული პრედიკატია), რომელიც გამოიყენება მოცემული ზომისა და პარამეტრების ფანჯრის შესაქმნელად და ეკრანზე გამოსატანად. ზოგადად მას აქვს შემდეგი სახე:

window(Hendle, Parent, Win_func(_), Title, X, Y, Width, Height).

Hendle ფანჯრის სახელია, რომლის მეშვეობითაც შემდგომში შესაძლებელია ამ ფანჯარასთან მუშაობა. ძირითადად გამოიყენება გლობალური ცვლადები, მაგრამ ჩვენს შემთხვევაში იგი არაა საჭირო და ეიყენებთ ფიქტიურ ცვლადს: (_).

Parent არის ე.წ. „მშობელი“ ფანჯრის სახელი. ჩვენ აქაც ეიყენებთ ფიქტიურ პარამეტრს.

Title არის ფანჯრის სათაური. ჩვენს შემთხვევაში „ base of car “.

X,Y წარმოადგენენ ფანჯრის მარცხენა ზედა კუთხის კოორდინატებს. ხოლო Width და Height შესაბამისად ფანჯრის სიგრძე და სიმაღლეა პიქსელებში.

Win_func() არის მოცემული ფანჯრის ფუნქციის სახელი. სახელი შესაძლებელია იყოს ნებისმიერი. ჩვენს შემთხვევაში იგი წარმოადგინებთ იმავე სახელით, როგორც ეს ზოგად ფორმაშია: „Win_func(_)“. მისი მეშვეობით შესაძლებელია ფანჯარაზე სხვა და სხვა სახის მანიპულაციების განხორციელება. პრედიკატ Win_func()-ით ფანჯარას ეგზავნება

სხედასხვა ტიპის გზავნილი (messages). ქვემოთ მოყვანილია ამ ფუნქციის რეალიზაცია განსახილველ პროგრამაში:

```
win_func(init) :-
    menu(pop_up, _, menu_window(), "file"),
    menu(pop_up, _, menu_go(), "Car"),
    menu(normal, _, menu_about(), "About"),

    edit(G_E1, _, edit_func(), "", 150, 10, 100, 25),
    edit(G_E2, _, edit_func(), "", 150, 50, 100, 25),
    edit(G_E3, _, edit_func(), "", 150, 100, 100, 25),
    edit(G_E4, _, edit_func(), "", 150, 150, 100, 25),
    edit(G_E5, _, edit_func(), "", 150, 200, 100, 25),
    edit(G_E6, _, edit_func(), "", 150, 250, 100, 25),

    static(_, _, fa(), "Name:", 20, 10, 100, 20),
    static(_, _, fa(), "Year:", 20, 50, 100, 20),
    static(_, _, fa(), "Cilndr20, 200, 100, 20),
    static(_, _, fa(), "Km/h:", 20, 250, 100, 20),
    static(_, _, fa(), "Country:", 20, 150, 100, 20),
    static(_, _, fa(), "Hp:",
        button(_, bfunc(), "Record", 150, 300, 70, 25).
```

messages (გზავნილი) შესაძლებელია იყოს სხედასხვა ტიპის. ჩვენ ამ შემთხვევაში „win_func“ პრედიკატს გავუგზავნეთ შეტყობინება „init“-ს, რომლის დანიშნულებაცაა ფანჯრის კომპონენტების ფორმირება. პრედიკატი „win_func“ მოცემულია პროლოგის წესის სახით. შექმნილ ქვეფანჯარას აქვს სახე:

ნახ. 4.1. პროლოგ-პროგრამის მიერ ფორმირებული ფანჯარა

ახალშექმნილი ფანჯრის სათაურის სტრიქონში მოთავსებულია ფანჯრის სახელი, რომელიც მოთავსებულია ამოცანის მიზანში (კითხვაში) „base of car“.

მენიუს სტრიქონს ვაფორმირებთ იმიტატორში ჩაშენებული პრედიკატით: მენიუ (menu). პრედიკატის ზოგადი სახეა:

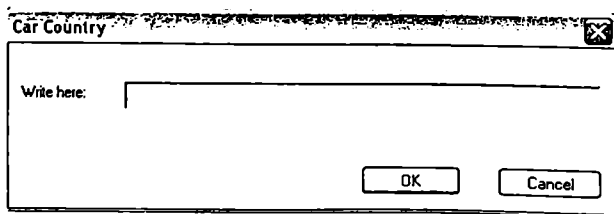
`menu( Type, Handle, Parent, Menu_func(), Text ).`

Handle, Parent კომპონენტებზე ზემოთ უკვე ვისაუბრეთ. მოცემულ შემთხვევაში ეს პარამეტრები არ გვჭირდება, აჩივთ მათ ადგილას ვათავსებთ დეფისის („-“) სიმბოლოს.

Type განსაზღვრავს მენიუს ტიპს. პროლოგი-იმიტატორში გამოიყენება რამოდენიმე სახის მენიუს ტიპი. ამ პროგრამაში ვიყენებთ ორი ტიპის მენიუს: **pop_up** და **normal**-ს. პირველი მათგანი გამოიყენება ე.წ. ჩამოშლადი მენიუების (ქვემენიუების) შესაქმნელად, ხოლო მეორე კი ცალკეული მენიუების შესაქმნელად. „Text“ პარამეტრი განსაზღვრავს მენიუს სახელს, მაგალითად ჩვენს შემთხვევაში შერჩეული იქნა მენიუს შემდეგი სახელები: **file**, **car**, **about**. ამასთან პირველი ორი ჩამოშლადი (**pop_up**) ტიპისაა, ხოლო მესამე „**normal**“ ტიპისა (იხილეთ პროგრამის ნაწყვეტი):

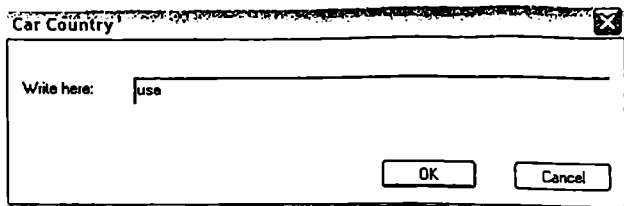
```
menu(pop_up, -, menu_window(),"file"),
menu(pop_up, -, menu_go(),"Car"),
menu(normal, -, menu_about(),"About"),
```

„Car“ მენიუ შეიცავს ც/ბ-ში მოთავსებული ავტომობილების პარამეტრების ჩამონათვალს: **Name**, **Year**, **Cilindr** და ა.შ. ამ კომპონენტებზე მიმთითებლის დაწკაპუნებით გამოდის შესაბამისი „Read“ ფანჯარა. მაგ. ქვემენიუ „Country“-ზე დაწკაპუნებით გამოდის ფანჯარა:



ნახ. 4.2.

მომხმარებელმა სატექსტო ველში (ტექსტ-ბოქსში) უნდა აკრიფოს იმ ქვეყნის დასახელება, რომლის ავტომობილებზეც სურს ინფორმაციის მიღება (ნახ. 4.3):



ნახ. 43.

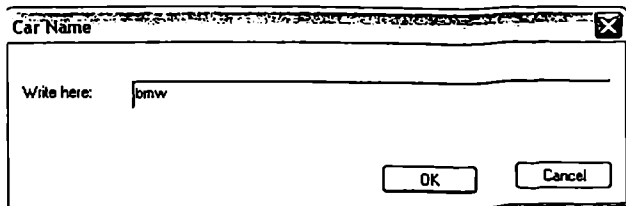
დადასტურების შემთხვევაში („OK“ კლავიშზე მიმოითებლის დაწკაპუნებით), **Output** ველში გამოდის ინფორმაცია ც/ბ-ში ამერიკული წარმოების ავტომობილების შესახებ.

Compiling the file:
C:\Program Files\Strawberry Prolog\beso\Car_finish.pro
0 errors, 0 warnings.

NAME	YEAR	CILINDR	COUNTRY	HP	KM/H
pontiac	1989	6	usa	208	215
ford	1988	4	usa	147	200
cadillac	1989	8	usa	160	185
lincoln	1989	6	usa	182	185
chreysler	1989	6	usa	143	195
jeep	1989	6	usa	180	175
lincoln	1992	8	usa	200	220
ford	1993	8	usa	150	220
jeep	1993	6	usa	150	200
pontiac	1991	8	usa	220	220
linkoln	2000	12	usa	225	300

No.

თუ აღნიშნულ ქვემენიუში მიმოითებელს დაეაწკაპუნებთ პარამეტრზე „Name“ და „READ“ ფანჯრის სატექსტო ველში აეკრეფთ საძიებელი ავტომობილის სახელს, ეთქვათ „BMW“-ს (ნახ. 4.4.), **Output** შედეგების ქვეფანჯარაში გამოვა ინფორმაცია ც/ბ-ში მოთაესებული ამ სახელის მქონე ყველა ავტომობილების შესახებ. იხილეთ შედეგის ამონაბეჭდი ნახ. 4.5.



ნახ. 44.

Compiling the file
C:\Program Files\Strawberry Prolog\beso\Car_finish.pro
0 errors. 0 warnings.

NAME	YEAR	CILINDR	COUNTRY	HP	KM/H
bmw	1986	6	germany	211	230
bmw	1995	8	germany	300	280

No
!

ნახ. 4.5.

„win_func” პრედიკატის შემდეგი კომპონენტია „edit” (რედაქტირება) პრედიკატი, რომელიც ფანჯარაში შესატანი ინფორმაციის ველებს ქმნის და ათავსებს მათ პროლოგ-პროგრამის მიერ შექმნილ ფანჯარაში. განსახილველ პროგრამაში „edit” პრედიკატის პირველი კომპონენტები: G_E1, G_E2, G_E3, ... , G_E6, ე.წ. გლობალური ცვლადებია. მათი არსი მოყვანილია [7] ნაშრომში. ამ შემთხვევაში მათი მნიშვნელობები ემთხვევა ავტომობილის ზემოთ ჩამოთვლილ პარამეტრებს: G_E1 „Name”-ს, G_E2, „Year”-ს, და ა.შ. „edit” პრედიკატის საშუალებით ახალ ფანჯარაში იქმნება ე.წ. ტექსტ-ბოქსები, რომლებსაც ამოცანის ამოხსნის შემდგომ ეტაპზე (ც/ბ-ში ახალი ინფორმაციის ჩამატებისას) ეიყენებთ.

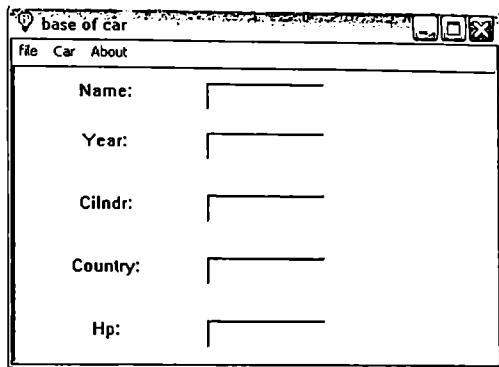
```
edit(G_E1,_,edit_func(), "", 150,10,100,25),
edit(G_E2,_,edit_func(), "", 150,50,100,25),
edit(G_E3,_,edit_func(), "", 150,100,100,25),
edit(G_E4,_,edit_func(), "", 150,150,100,25),
edit(G_E5,_,edit_func(), "", 150,200,100,25),
edit(G_E6,_,edit_func(), "", 150,250,100,25),
```

ციფრები 150 და 10 (50, 100, 150, 200, 250) ტექსტ-ბოქსის მარცხენა ზედა კუთხის კოორდინატებია პიქსელებში, ხოლო 100 და 25 ამ ქვეფანჯრის სიგრძე და სიგანეა.

ნახ. 4.1. მოყვანილი ფანჯრის სტატიკური ტექსტების (ისინი ფანჯრის მარცხნივ მდებარეობენ) შესატანად გამოიყენება პრედიკატი „static” (სტატიკური,უძრავი). მოცემულ ამოცანაში პროგრამის ტექსტში ისინი ჩაიწერება შემდეგი გრაფიკული პრედიკატებით:

```
static(_,_,fa(), "Name:", 20,10,100,20),
static(_,_,fa(), "Year:", 20,50,100,20),
static(_,_,fa(), "Cilindr20,200,100,20),
static(_,_,fa(), "Km/h:", 20,250,100,20),
static(_,_,fa(), "Country:", 20,150,100,20),
static(_,_,fa(), "Hp:",
button(,_,bfunc(),"Record",150,300,70,25).
```

იხილეთ ნახ. 4.6.



ნახ. 4.6.

ფანჯრებში სამართავეი ღილაკების შესაქმნელად გამოიყენება პრედიკატი „button“.

`button(„_“,bfunc(„Record“,150,300,70,25)).`

ზემოთ მოყვანილი გრაფიკული ოპერატორით ჩვენს ფანჯარაში შექმნილია „Record“ ღილაკი (ნახ. 4.1), რომლის მეშვეობითაც წარმოებს ც/ბ-ში ახალი ინფორმაციის ჩამატება (ამ ამოცანაში ახალი ავტომობილის შესაბამისი მონაცემების შეტანა. ამასთან აუცილებელია ინფორმაცია შევითანოთ იმავე ფორმატით).

`car("pontiac","1989","6","usa","208","215").`

იმიტატორის გრაფიკული ოპერატორების საშუალებით ც/ბ-ის პროექტირებისას უნდა გამოვიყენოთ მონაცემთა წარმოდგენის სტრიქონული ტიპი „string“, ამის გამო მონაცემები ც/ბ-ში აუცილებელია ჩავსვათ აპოსტროფებში: „ “ .

ქვემოთ მოყვანილია ც/ბ-ში ახალი ინფორმაციის ჩამატების პროგრამული ფრაგმენტი:

```
bfunc(press):-
X is get_text(G_E1),
Y is get_text(G_E2),
Z is get_text(G_E3),
B is get_text(G_E4),
C is get_text(G_E5),
D is get_text(G_E6),
assert_in(car(X,Y,Z,B,C,D)),
mmessage("Thank", "Thank you for record our Data Base!",!),
set_text(„“,G_E1),
set_text(„“,G_E2),
set_text(„“,G_E3),
set_text(„“,G_E4),
set_text(„“,G_E5),
set_text(„“,G_E6).
```

ნახ. 4.7.

„bfunc”(press), button გრაფიკული ოპერატორით შექმნილი ღილაკის ფუნქციის აღწერი პრედიკატია, რომელსაც გსაენილით ეგსაენება მოცემულ მომენტში ღილაკის მიერ შესასრულებელი დავალება. ამ შემთხვევაში „press” (დატერა) „Record” ღილაკსე ნიშნავს ც/ბ-ში ახალი ინფორმაციის ჩამატებას. ამისათვის ეიქცეეით შემდეგნაირად:

1. შეეიტანოთ საინფორმაციო ეელეებში (რომლებიც პროლოგ-პროგრამაში აღწერილია edit პრედიკატით) (იხ. ნახ. 4.1) შესაბამისი ინფორმაცია „get_text” გრაფიკული ოპერატორით (მონაცემები ახალი ავტომობილის შესახებ).
2. „get_text” პრედიკატით, ც/ბ-ის car(X, Y, Z, A, B, C) ფუნქტორის ცვლადებს ენიჭებათ G_E1, G_E2, ..., G_E6 გლობალური ცვლადების მნიშვნელობები.
3. ღილაკსე „Record” დატერით, ც/ბ-ის მართვის ოპერატორი „assert_in” ახდენს ახალი ინფორმაციის შეტანას ც/ბ-ის ბოლოში (ც/ბ-ში ინფორმაციის ბოლოში ჩამატება განპირობებულია მართვის „assert_in” ოპერატორის არსიდან). პროლოგის იმიტატორში ასევე გამოიყენება ც/ბ-ის მართვის სხვა ოპერატორებიც, რომელთაც ჩვენ შემდგომში საჭიროებისამებრ გამოიყენებთ.
4. „set_text” პრედიკატით სუფთადება (ნულდება) საინფორმაციო ეელეები ახალი ინფორმაციის მისაღებად და გამოდის შეტყობინება ქეფანჯრის სახით:



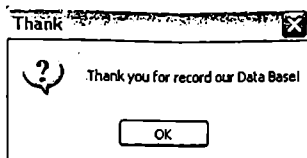
ნახ. 4.8.

ქეფანჯარა „Thank”-ს მასში ჩაწერილი ტექსტი მიეწოდება ნახ.4.7. მოცემული ამოცანის ფრაგმენტში ჩაწერილი message-ით.

ალსანიშნავია ერთი გარემოება. ნახ.4.8.-ზე სამკუთხედში მოთავსებული ძახილის ნიშანი (!) ფორმირდება პროგრამულად, გსაენილის მესამე პარამეტრის „ ! ”-ის ნიშნის მეშეობით.

`message("Thank", "Thank you for record our Data Base!", !),`

სხვა სახის გამაფრთხილებელი შეტყობინების საჭიროების შემთხვევაში შესაძლებელია გამოიყენოთ, ეთქვას „?”-ის ნიშანი. ფანჯარა მიიღებს სახეს (იხ. ნახ.4.9). თუ გამაფრთხილებელი შეტყობინების ადგილას გამოიყენებთ „!” სიმბოლოს, მაშინ ქეფანჯარაში გამაფრთხილებელი ნიშანი სამკუთხედი არ გამოვა.

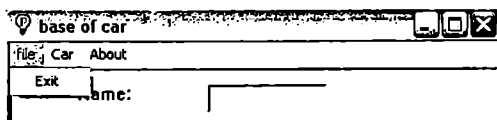


ნახ. 4.9.

ხოლო პროგრამაში შესაბამის შეტყობინებას ექნება შემდეგი სახე:

message("Thank", "Thank you for record our Data Base!", ?).

პროგრამულად შექმნილი ნახ. 4.1. წარმოდგენილი ქვეფანჯრის დასასურად ვიყენებთ ჩამოშლად მენიუ „File“-ის ქვემენიუს ამ შემთხვევაში ერთადერთ „Exit“ პარამეტრს. იხ. ნახ. 4.10.



ნახ. 4.10.

პროგრამულად ეს ასე ჩაიწერება:

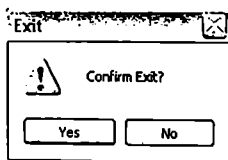
menu_window(init):-menu(normal, __,menu_exit(),"Exit").

პროგრამიდან გამოსასვლელად ვიყენებთ პროგრამის ფრაგმენტს:

win_func(close) :-not(yes_no("Exit", "Confirm Exit?", !)).

menu_exit(press) :- close_window(_).

ანუ პროგრამის ძირითადი მიზნის ფუნქციას „win_func“-ს ეუფაენით გზავნილს close. იმიტატორის რეაქცია ამოცანიდან „გამოსვლა-არგამოსვლის“ (yes_no) შესახებ აისახება ქვეფანჯრით. იხ. ნახ. 4.11.



ნახ. 4.11.

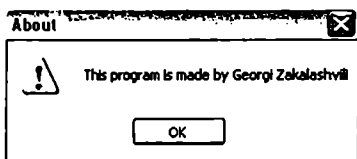
მომხმარებელი მიმოიხივების დაწკაპუნებით დაადასტურებს („yes“) ან არ დაადასტურებს („no“) ფანჯრის დახურვის სურვილს.

„normal“ ტიპის (არანამოშლადი) მენიუ „about“-ის ფუნქციურ პრედიკატს „menu_about“ მიეწოდება შეტყობინება:

menu_about(press):-message("About", "This program is made by Georgi Zakalashvili [22 November 2004]",!).

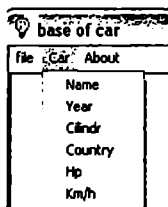
გზავნილი აზუსტებს პროლოგ-პროგრამის შემოღენელის ვინაობას.

შესაბამისად ეკრანზე გამოდის ქვეფანჯარა და დადასტურების შემთხვევაში („OK“) ქვეფანჯარა იხურება.



ნახ. 4.12.

„pop_up“ ტიპის ნამოშლადი მენიუს „Car“-ის ფუნქციურ პრედიკატს: „menu_გ“-ს მიეწოდება გზავნილი, რომელიც შეიცავს ქვემენიუს პუნქტებს ჩამონათვალის სახით. იხილეთ ნახ. 4.13.



ნახ. 4.13.

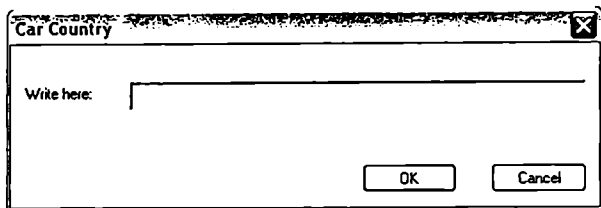
პრედიკატები s1, s2, . . . , s6 აღწერენ პროლოგ-სისტემის მოქმედებას „Car“ მენიუს რომელიმე პუნქტის არჩევისას. პროლოგ-პროგრამის შესაბამის ფრაგმენტს აქვს სახე:

s4(press):-

```
read(A, "Car Country",s),nl,
write("NAME"),write(" "),
write("YEAR"),write(" "),
write("CILINDR"),write(" "),
write("COUNTRY"),write(" "),
write("HP"),write(" "),
write("KM/H"),nl,nl,
car(X,Y,Z,A,B,C),
write(X),write(" "),
write(Y),write(" "),
write(Z),write(" "),
write(A),write(" "),
write(B),write(" "),
write(C),nl,fail.
```

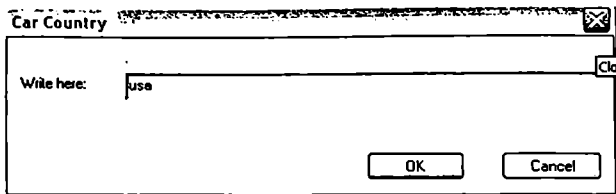
ნახ. 4.14.

მოვიყვანოთ ნახ. 4.14. მოცემული ფრაგმენტის აღწერა. პრედიკატ s4-ს ეგზავნება გზავნილი `press` (დაჭერა). `s4-ს` შეესაბამება ც/ბ-ში მოთავსებული პრედიკატ „car“-ის კომპონენტი: „Country“. მას ასევე შეესაბამება მენიუ „Car“-ის ერთერთი ქვეპუნქტიც. მენიუში ამ პუნქტზე მიმითვებლის დაწკაპუნებით (დაჭერით) ეკრანზე გამოდის ქვეფანჯარა, რომელიც მოყვანილია ნახ. 4.15. ამ შემთხვევაში მომხმარებელმა დააჭირა პუნქტს „Country“ (ავტომატურად დაამზადებული ქვეყანა).



ნახ. 4.15.

მომხმარებელი შეიტანს საინფორმაციო ველში ეოქვათ: `usa-ს`. შესაბამის ქვეფანჯარას ეგნება სახე:



ნახ. 4.16.

მიმოიხილეთ ქვეყანების „OK“ კლავიშზე დაჭერით იმიტატორი ც/ბ-დან ამოიღებს ამერიკული წარმოების ავტომობილების ყველა ექვს კომპონენტს. დასმულ კითხვაზე პასუხები გამოიტანება ამოცანის ფანჯრის მარჯვენა ქვეყანაში **Output**.

ნახ. 4.14. ამოცანის ფრაგმენტში გამოყენებული ოპერატორით: `read( Term, String, Flag )`, Term-ს შეუთავსდება ნახ. 4.16. ქვეყანაში მოყვანილი თერმი: „usa“. String შეუთავსდება ფანჯრის დასახელება „Car Country“, ხოლო „Flag“-ს შეუთავსდება სიმბოლო „s“.

ამრიგად, ამოცანა „ავტომობილების ც/ბ-ის შესახებ“ ამოეხსენით ორი მეთოდით. მეორე მეთოდს, რომელშიც გამოყენებულია გრაფიკული ოპერატორები გააჩნია ეიზუალური ეფექტი. ეს ეფექტი უფრო ნათლად ჩანს ჩვენს მიერ მოყვანილ ამოცანებში, რომელთაც ქვემოთ განვიხილავთ.

ზემოთ განხილული ამოცანა, ნაშრომის ავტორთა აზრით, შესაძლებელია გამოყენებული იქნეს აღნიშნული კურსის შესწავლისას, I სემესტრის ბოლოს ლაბორატორიული სამუშაოს სახით.

4.2. პროექტის იმიტატორის ბრაუზერული ოპერატორების ზოგადი მიმოხილვა

გარდა 4.1-ში მოყვანილი გრაფიკული ოპერატორებისა, პროლოგის მოცემული ვერსიის იმიტატორს ასევე გააჩნია სხვადასხვა დანიშნულების გრაფიკული პრედიკატები. შეიძლება ამებრ მოვახდინოთ მათი მიმოხილვა შესაბამისი მაგალითების დართვით.

მოყვანილ ჩვენი სპეციალობის მაგისტრის გიორგი ბასილიას მიერ შედგენილი პროლოგ-პროგრამა, რომელიც მოქმედებაში წარმოგიდგენს ფანჯრებთან მუშაობის ტექნიკას. პროგრამას თან ახლავს ინგლისურენოვანი კომენტარები.

?-
`window( , win_func( ), "Course Project", 100, 100, 400, 200).` % Show current window
 with given size

`win_func(close) :-`
`not( yes_no( "", " Are You Sure?", ?)).`

% Function Y/N
 % Are You sure?

```

win_func(init) :-                                % Start of Function
edit_n(G, MainEdit, _, edit_func, "This is a Demonstration of Functions With Menus &
Windows in Prolog...", 0, 0, 500, 100, 2, _, _),

menu(normal, _, _, menu_window(_), "&New window"),      % New Window
menu(normal, _, _, menu_go(_), "&Go !"),                % GO - Expand !
menu(normal, _, _, menu_name(_), "&Change The Title"),    % Change Name of Window
menu(normal, _, _, menu_menu(_), "&Extend The Menu"),    % Add Menu Item
menu(normal, _, _, menu_about(_), "&About Program"),     % About Project
menu(normal, _, _, menu_exit(_), "E&xit Program").      % Exit Function

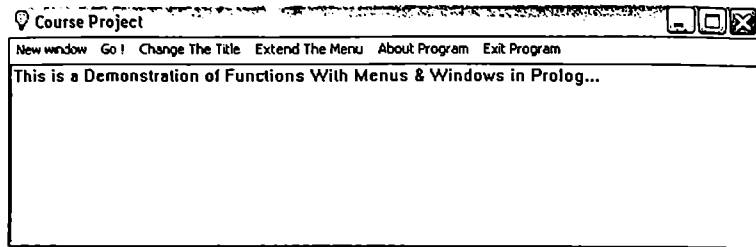
menu_window(press) :-                               % After a Click on New Window
position(_, X, Y),                                   % Position of Window - X,Y
window(_, 0, win_func(_), "Window", X+120, Y+200, 300, 200). % Show New window on New
                                                                position
menu_go(press) :-                                    % After a Click on
                                                                menu item - GO
size(_, X, Y),                                       % Resize Window
size(_, X+1, Y),                                     % Resize To Right
X<600,                                                % Until X value is reached
menu_go(press).
menu_name(press) :-                                  % After A click on "Change Name"
Menu item
read(X, "Give a new name to the window", s),        % Show Input Text Box with a Title
set_text(X, _).                                     % Set Window Title
menu_menu(press) :-                                  % After a click on Extend menu item
read(X, "Give a name for a new menu item", s),      % Show input Text Box (X) With a
Title
menu( normal, _, _, menu_go(_), X).                  % Create new menu item from a text
(X) & Act like GO! function
menu_about(press):-
message("About", " Course Project\n Group 108033\n Author: George Basilaia\n 2004
Year\n\nGeorgian\n Technical\n University", i),
write("OK"), nl.
menu_exit(press) :-                                  % On press of Exit
close_window(_).                                     % Close Window

```

ნახ. 4.17.

გაეუშვათ ამოცანა შესრულებაზე. მივიღებთ ფანჯარას, რომელიც მოცემულია ნახ. 4.18. ამოცანაში კარგად არის მოფიქრებული ის მომენტი, რომ პროგრამულად შექმნილი ფანჯრის მენიუს სტრუქტურაში მენიუს კომპონენტებად გვევლინება იმიტატორის გრაფიკული პრედიკატები.

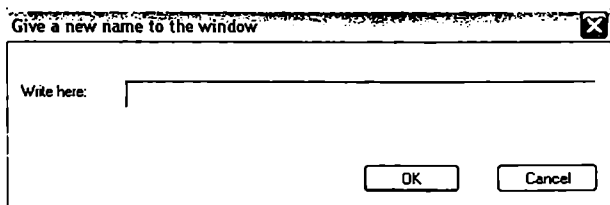
ეს კი იძლევა საშუალებას, ამოცანის შესრულებაზე გაშვებისას, ვიხილოთ ეს პრედიკატები უშუალოდ მათი ფუნქციების შესრულებისას.



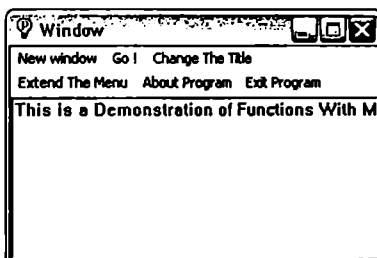
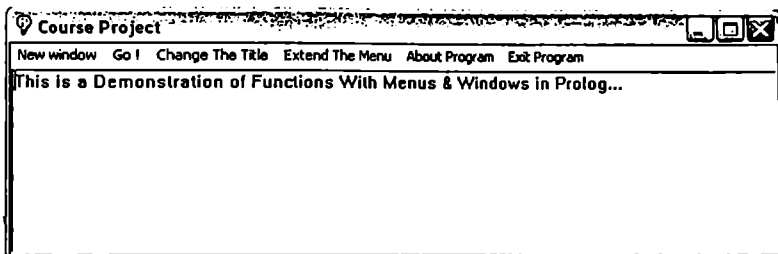
ნახ. 4.18

გრაფიკული პრედიკატი, რომელიც მოთავსებულია მენიუს სტრიქონში „New window“ მასზე მიმოთივლების დაწკაპუნებისას იწვევს ახალი იმავე ფანჯრის გაჩენას. იხ. ნახ. 4.20. პრედიკატ „Go“-ზე დაწკაპუნებით ფანჯარა დაგრძელდება მარჯვნივ პროგრამაში მიმოთივებული პარამეტრებით.

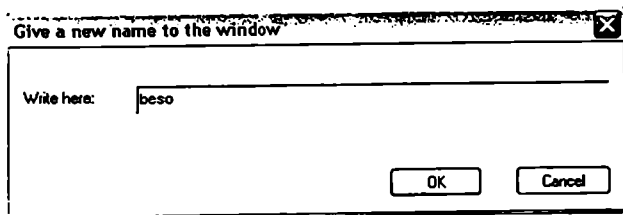
პრედიკატ „Change The Title“-ზე მიმოთივების დაწკაპუნებით შესაძლებელია ფანჯრის სათაურის სტრიქონში ჩაწერილი სათაურის შეცვლა. ფანჯარაში გამოდის ახალი სათაურის შესატანი „Read“-ის ფანჯარა. იხილეთ ნახ. 4.19. ამ ფანჯრის სათაურის სტრიქონში მოთავსებულია ტექსტი: „Give a new name to the window“ საინფორმაციო ველში შევიტანოთ ახალი სათაური, ეთქვას „beso“. დადასტურების შემთხვევაში ფანჯრის სათაური იცვლება ახალი სათაურით. იხ. ნახ. 4.21. და ნახ. 4.22. ასევე შესაძლებელია მენიუს სტრიქონის კომპონენტების დამატება.



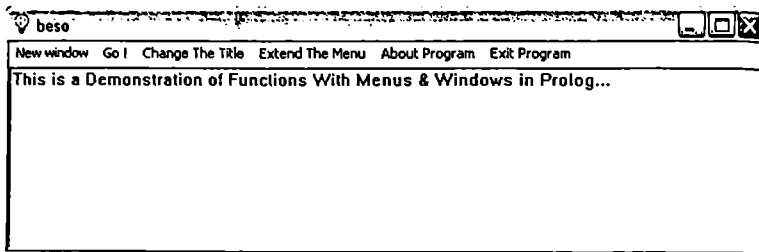
ნახ. 4.19. ფანჯრის სათაურის შეცვლა „Change The Title“ პრედიკატით



ნახ. 4.20. ახალი იდენტური ფანჯრის გაჩენა



ნახ. 4.21.



ნახ. 4.22. სათაურ შეცვლილი ახალი ფანჯარა

ზემოთ აღწერილი გრაფიკული პრედიკატები უმარტივესი ტიპის პრედიკატებია. მიმოვიხილოთ შედარებით „რთული“ პრედიკატები.

პროლოგ-იმიტატორი იძლევა საშუალებას პროგრამულად შექმნილ ფანჯრებზე მოეხდინოთ სხედასხვა სახის მანიპულაციები.

შესაძლებელია ფანჯრის უკანა ფონის (background) შეცვლა, როგორც სხედასხვა შეფერილობით, ასევე უკანა ფონად ნებისმიერი გრაფიკული ფაილის (სურათის) დართვა.

სხედასხვა ფერის მისაღებად გამოიყენება იმიტატორში ჩაშენებული პრედიკატი: `rgb` (Red Green Blue). მაგალითად, `rgb( 225, 225, 0 )` ყვითელი ფერია, `( 125, 225, 0 )` მწვანე ფერი, `rgb( 0, 0, 0 )` შავი ფერი, ხოლო `(225, 225, 225)` ნაცრისფერი.

პრედიკატ `beep`-ის გამოყენებით იმიტატორი გამოსცემს სხედასხვა ტონალობის ხმოვან სიგნალს wav გაფართოებით. პრედიკატში უნდა ჩაიწეროს სიგნალამდე მისასვლელი გზა. ხმოვანი სიგნალის ზოგადი მისამართი ასე ჩაიწერება:

`beep ( " C: \ \ WINDOWS \ \ Media \ \ ...wav" ).`

ფანჯრის უკანა ფონად სურათის ჩასასმელად გამოიყენება პრედიკატი: `bitmap` (გრაფიკული ობიექტის ციფრული სახით წარმოდგენა). ამასთან, გრაფიკული ობიექტების შესაბამისი ფაილები აუცილებლად უნდა იყოს ***. bmp გაფართოებით (გაფართოება „bmp“ სიტყვა `bitmap`-ის შეკეცილი ფორმაა). მაგალითად პრედიკატი:

`bitmap ( _ _ f 9(, X +print ( "cat.bmp" ), 260, 230 )`, ფანჯარაში (260, 230) კოორდინატზე გამოიტანს Window-ის მუდმივ მეხსიერებაში მოთავსებულ სურათს „cat.bmp“. ყველა გრაფიკული ფაილი კომპიუტერში მოთავსებულია „C“ დისკის ერთ ღირეპტორიაში, მისამართზე:

`_ _ C:\Documents and Settings\All Users\Application Data\Microsoft\User Account Pictures\Default Pictures`



ნახ. 4.24.

პრედიკატი: line (X1, Y1, X2, Y2, . . . , Xn, Yn), სადაც X1, Y1 და ა.შ. გასაღები ხაზის კოორდინატებია. მისთვის სისქისა და ფერის მისაცემად ვიყენებთ პრედიკატს:

pen (PenWidth, Color). ამ პრედიკატების გამოყენების მაგალითია:

```
win_func ( paint ) :-  
    pen ( 2, rgb ( 0, 0, 0 ) ),  
    line ( 0, 40, 430, 40 ),
```

გრაფიკულ ფუნქციას win_func-ს ეგზავნება გზავნილი paint (ხატვა).

პრედიკატი color_text_back (Window, Color). ტექსტის ფონის ფერია ფანჯარაში.

პრედიკატი window_brush ფანჯრის შემფერადებელს ნიშნავს. მისი საშუალებით ფანჯარა იღებს rgb პრედიკატში ციფრთა სამეულით აღწერილ ფერს.

პრედიკატი color_text-ით იცვლება უშუალოდ ფანჯარაში მოთავსებული ტექსტის ფერი.

გარდა ზემოთ მოყვანილი გრაფიკული პრედიკატებისა პროლოგ-იმიტატორი ასევე შეიცავს სხვა დანიშნულების გრაფიკულ პრედიკატებსაც. ჩვენ შევეცადეთ მოგვეყვანა მათი უმრავლესობის დანიშნულების აღწერა და გვეჩვენებინა პრედიკატების მოქმედება განხილული მაგალითების საფუძველზე. დინტერესებულ მკითხველს აქვს შესაძლებლობა იმიტატორის „Help“-ის დახმარებით გაეცნოს მათ უფრო დაწვრილებით.

4.3. რელაციური (ჩაშენებული) ბაზების პროექტირების ამოცანებში გრაფიკული პრედიკატების გამოყენება

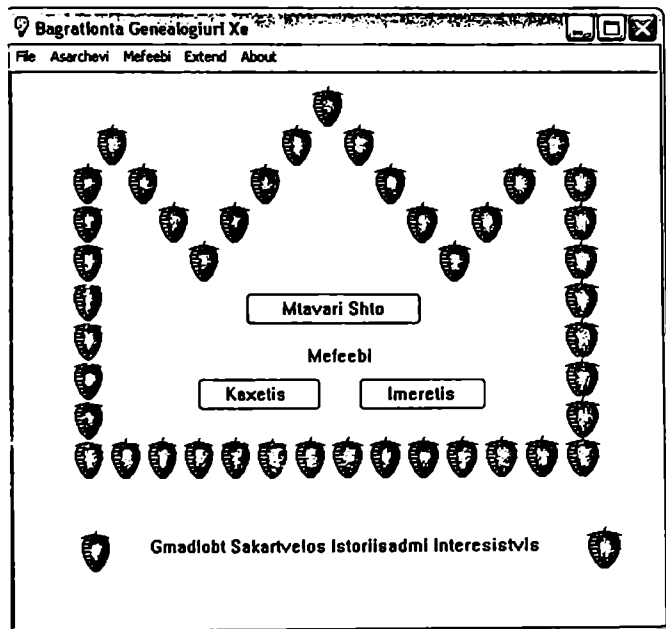
ნაშრომის მეორე თავში ჩვენ განვიხილეთ ამოცანები, რომლებიც დაფუძნებული იყო ე.წ. რელაციურ (პროლოგ-პროგრამის ტანში) „ჩაშენებულ“ ცოდნის ბაზებზე (ც/ბ). ამოცანები შეეხებოდა ადამიანის გონებრივი მოღვაწეობის სხვადასხვა სფეროს. ამ ქვეთავში განვიხილავთ ამოცანებს, რომლებიც შედგენილია უშუალოდ სახელმძღვანელოს ავტორების მიერ, ასევე ჩვენი სპეციალობის სტუდენტებისა და მაგისტრების ნამუშევრებს. ამოცანები შეეხება სხვადასხვა „საპრობლემო არეს“ და ბოლო ხუთ წელიწადში ლაბორატორიულ მეცადინეობებზე შესრულებული პროგრამებისა და საკურსო სამუშაოების ნაკრებს წარმოადგენს. პროლოგ-პროგრამებში რელაციური ც/ბ-ს გარდა, ასევე გამოყენებულია ზემოთ აღწერილი გრაფიკული პრედიკატები. ქვეთავში მოყვანილია ამოცანების ზოგადი დახასიათება. ძირითადად ყურადღება გამახვილებულია ამოცანის მნიშვნელოვან მომენტებზე და ნიუანსებზე. პროლოგ-პროგრამების სრული ამონაბეჭდები და ამოცანების (საკურსო სამუშაოების) აღწერები მოყვანილია ნაშრომის ბოლოს დანართი 3-ში.

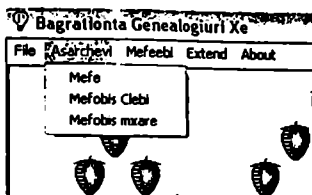
ამოცანის მიზანია ეიზულურად, გრაფიკული ფანჯრების საშუალებით წარმოგვიდგინოს საქართველოს სამეფო დინასტიის გენიალოგიური ხე.

ამოცანა შედგება რელაციური ც/ბ-გან, რომელიც შეიცავს ინფორმაციას საქართველოს სამეფო დინასტიის წარმომადგენლების შესახებ: მათი მოღვაწეობის პერიოდს (წელი, ხაუკუნე), საქართველოს რომელი მხარის (დასავლეთი, აღმოსავლეთი) მეფეები (ან მთავრები) იყვნენ. პროგრამულად შედგენილი გრაფიკული ფანჯრების მეშვეობით გეძლევა საშუალება ყოველივე ეს წარმოვადგინოთ ეიზულურად, ლამაზი ფორმით.

ამ ამოცანის შესრულებაზე გაშვების შემდეგ, ეკრანზე გამოდის გრაფიკული პრედიკატებით შექმნილი ძირითადი ფანჯარა, რომელიც ნახ. 4.25-ზეა წარმოდგენილი. გრაფიკული ფანჯარა შეიცავს სათაურის სტრიქონს, მენიუს სტრიქონს და უშუალოდ გრაფიკულ სურათს. ძირითადი ფანჯრის სათანადო დილაქებზე მიმთითებლის დაწკაპუნებით გამოდის სხვადასხვა სახის ქვეფანჯრები. მათ ჩვენ ქვემოთ განვიხილავთ.

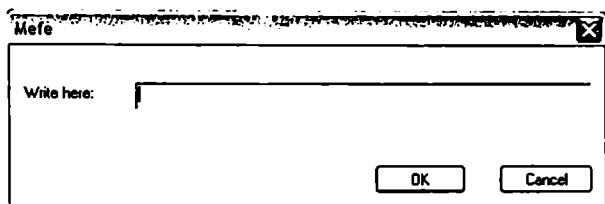
მენიუს სტრიქონის კომპონენტზე „Asarchevi“ (იხ. ნახ. 4.26.) დაწკაპუნებით ეკრანზე გამოდის ამ კომპონენტის შესაბამისი ქვემენიუ, რომელიც შეიცავს შემდეგ პუნქტებს: Mefe, Mefobis Clebi, Mefobis mxare (იხ. ნახ. 4.26).





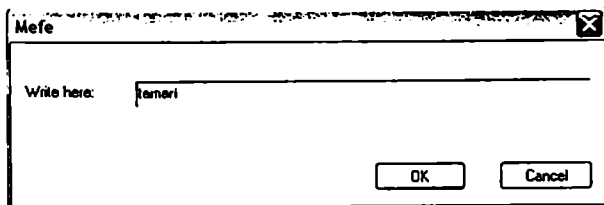
ნახ. 426.

ქვემენიუს პუნქტზე „Mefe“ დაწკაპუნებით გერანზე გამოდის ინფორმაციის შეტანის ოპერატორის „read“-ის შესაბამისი ქვეფანჯარა (იხ. ნახ. 427).



ნახ. 427.

ქვეფანჯრის საინფორმაციო ველში ამოცანის ც/ბ-ში მოთავსებული რომელიმე შეფის, უთქვამთ „tamari“-ის სახელის ჩაწერის შემდეგ (ნახ. 428.), შედეგების ~~Output~~ ველში გამოდის ც/ბ-დან აღებული ინფორმაცია. იხ. ნახ. 429.



ნახ. 428.

Compiling the file:
C:\Program Files\Strawberry Prolog\beso\bagratiunebi.pro
0 errors, 0 warnings.

Mefe	Mefobis Clebi	Mxare
tamari	1184-1213	ertiani sakartvelo

No

ნახ. 429.

თუ მიმთითებულს დაეწაკაუნებთ ნახ. 425.-ზე წარმოდგენილ ქვემენიუსე „Mefobis mxare“, მაშინ **Output** ელში გამოიტანება ინფორმაცია ც/ბ-დან საქართველოს აღნიშნული მხარის მეფეების შესახებ. (იხ. ნახ. 430). ელში გამოტანილია ინფორმაცია კახეთის მეფეების შესახებ.

Compiling the file:
C:\Program Files\Strawberry Prolog\beso\bagratiunebi.pro
0 errors, 0 warnings

Mefe	Mefobis Clebi	Mxare
giorgi 8	1466-1476	kaxeti
aleksandre 1	1476-1511	kaxeti
giorgi 2	1511-1513	kaxeti
levan 1	1518-1574	kaxeti
aleksandre 2	1574-1605	kaxeti
davit 1	1601-1602	kaxeti
konstantine 1	1605	kaxeti
teimuraz 1	1606-1648	kaxeti
teimuraz 1	1625-1633	kaxeti
daviti	16??-16??	kaxeti
erekle 1	1703-1709	kaxeti
konstantine	1722-1732	kaxeti
teimuraz 2	1732-1744	kaxeti
davit 2	1703-1722	kaxeti
erekle 2	1744-1762	kaxeti

No

ნახ. 430.

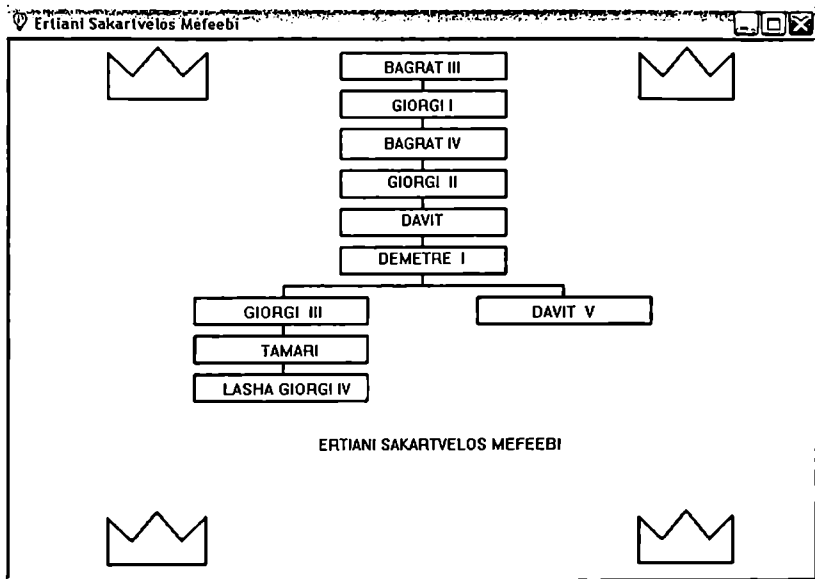
განსახილველი პროლოგ-პროგრამის შინაარსიდან გამომდინარე შესაძლებელია იგივე სახის ინფორმაციის მიღება ქვეფანჯრების სახით. მათში ინფორმაცია წარმოდგენილი იქნება გრაფიკული სახით (გენიალოგიური ხეების მეშვეობით).

ძირითადი ფანჯრის (იხ. ნახ. 425.) ღილაკზე „მთავარი შტო“ დაჭერით გამოდის ქვეფანჯარა გაერთიანებული საქართველოს სამეფო დინასტიის შესახებ იხ. ნახ. 431

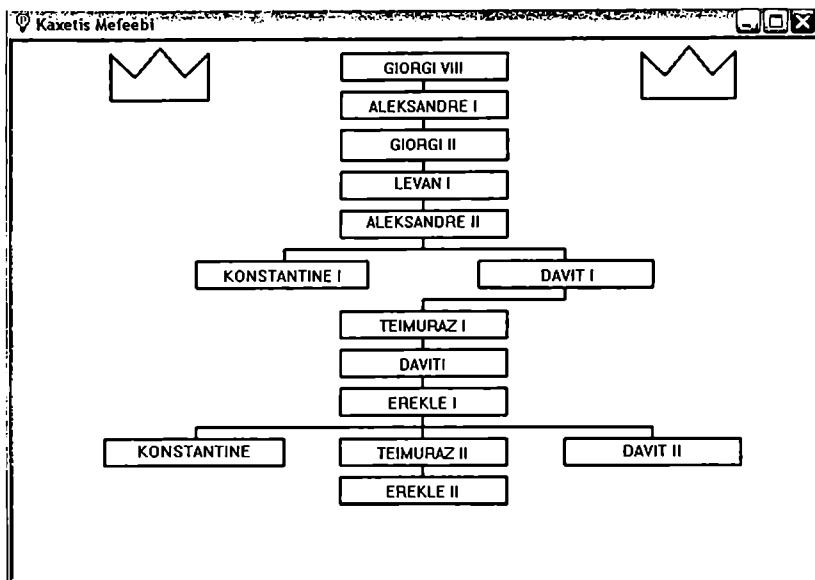
ღილაკზე „Kaxetis“ დაჭერით გამოდის ქვეფანჯარა: კახეთის სამთავროს მეფეების შესახებ გენიალოგიური ხის სახით. იხ. ნახ. 432.

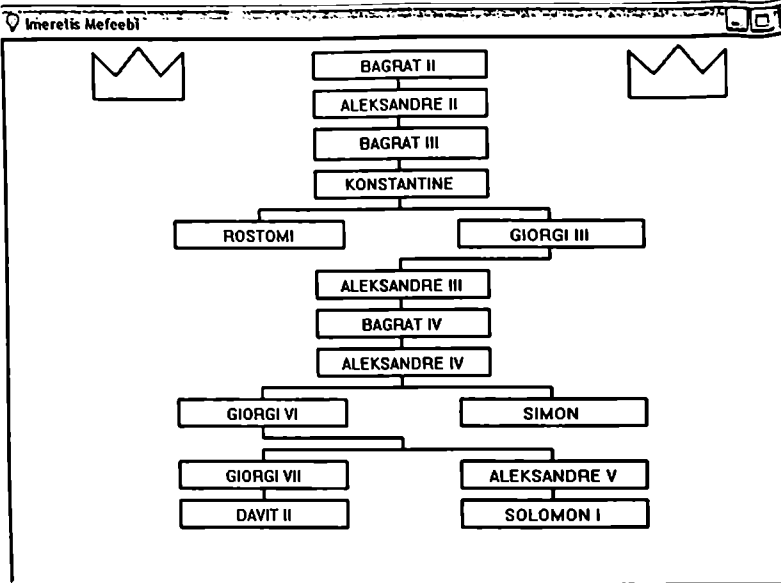
ღილაკზე „Imeretis“ დაჭერით გამოვა ქვეფანჯარა: იმერეთის სამთავროს მეფეების შესახებ. იხ. ნახ. 433.

ამოცანის პროლოგ-პროგრამის სრული ტექსტი მოცემულია დანართი 3-ში.



ნახ. 4.31. გაერთიანებული (ერთიანი) საქართველოს მეფეები





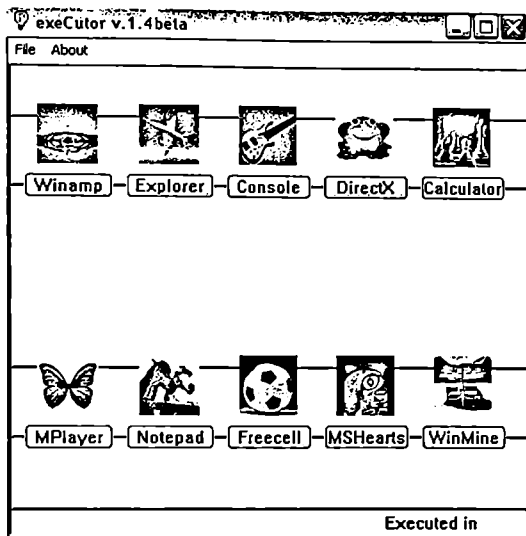
ნახ. 432. და ნახ. 433.

ამოცანა 2. შესრულებული (exeCutor).

ამოცანის დანიშნულებაა გაუშვას შესრულებაზე *.exe გაფართოების მქონე პროგრამები. ამასთან ერთად იგი ც/ბ-ში აფიქსირებს მათ გასაშვებად საჭირო დროს და ამ პროგრამამდე მისასვლელ ბილიკს (path). ამავე პროგრამას აქვს უნარი მიმართოს ფაილებს *.pro გაფართოებით, ე.ი. გამოიძახოს პროლოგ-პროგრამები. განსახილველ პროგრამაში გამოყენებულია გრაფიკული პრედიკატები. გრაფიკული ფანჯარა, რომელიც ასრულებს ძირითად ფუნქციებს მოცემულია ნახ. 434.

ამოცანის პროლოგ-პროგრამის სრული ტექსტი ისევე როგორც ამ ქვეთავში მოყვანილ დანართში პროგრამებისა და საკურსო სამუშაოების ტექსტები მოყვანილია შესაბამის დანართში.

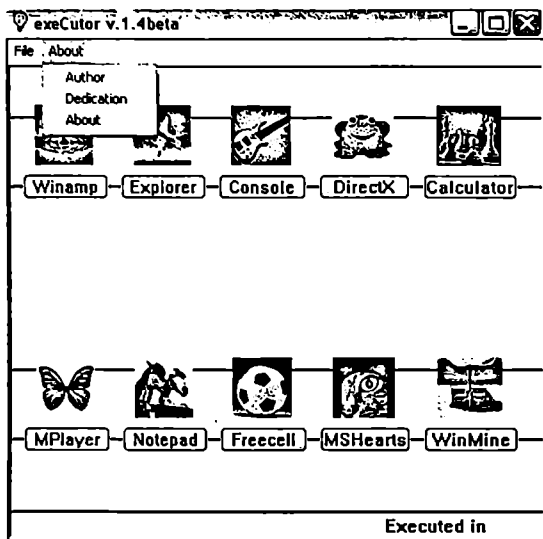
დანართში, დაინტერესებულ მკითხველს (სტუდენტი იქნება ეს, მაგისტრი, თუ ლაბორატორიული სამუშაოების ხელმძღვანელი) საშუალება ეძლევა პასუხი გასცეს მისთვის საინტერესო თითქმის ყველა შეკითხვას.



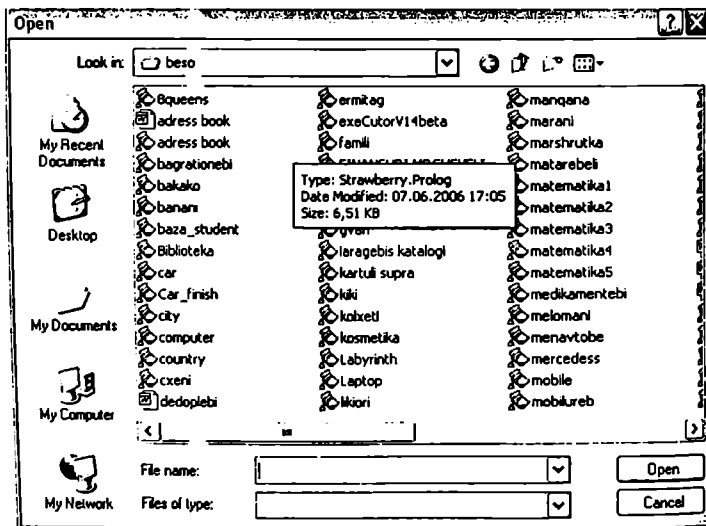
ნახ. 4.34. ძირითადი ფანჯარა

ძირითად ფანჯარაში, რომელსაც აქვს ნაცრისფერი უკანა ფონი (ფონს აფორმირებს პრედიკატი rgb) და ფორმირებულია პროლოგის გრაფიკული პრედიკატების ბაზაზე, მოთავსებულია ყველა იმ პროგრამების სახელები (ასეთი 10 პროგრამაა), რომელთა გამოძახებაც ისურვა პროგრამის შემდგენელმა. იგი ჩვენი სპეციალობის მაგისტრი მაშა ბელმასოვაძე გამოსაძახებელი, Window-ს სტანდარტული პროგრამების (გაფართოებით *.exe ფაილები) სახელები მოთავსებულია ღილაკებზე. ფანჯრის შექმნის ტექნოლოგიას, პროგრამაში გამოყენებული გრაფიკული პრედიკატების დანიშნულებას, ღილაკების შექმნის, ფონისა და გრაფიკული ფაილების (სურათების) ჩასმას და მრავალ სხვა ნიუანსს სვერ სვეით შევეხეთ. მოვიყვანით ამ ამოცანის შესრულების სოფიერითი მომენტი.

მენიუს სტრუქტურის პირველი კომპონენტი: File, რომელიც შეიცავს ქვემენიუს. იხილეთ ნახ. 4.35. ქვემენიუს პირველი კომპონენტი: „Open File ...” იძახებს პროლოგ-მიმტატორის „Beso” ლირექტორიას, რომელშიც განთავსებულია პროლოგ-პროგრამები (ფაილები *.pro გაფართოებით) იხ. ნახ. 4.36. შემდგომ ბიჯზე მომხმარებელს ეძლევა საშუალება გამოიძახოს ამ ლირექტორიაში შემავალი ნებისმიერი პროლოგ-პროგრამა და შეასრულოს იგი. მენიუს პირველი კომპონენტი ასევე შეიცავს ქვემენიუს: „Exit”, რომლის არსიც ჩვენთვის ცნობილია. „File”-ს მეორე კომპონენტით „Delete Record” წარმოებს ც/ბ-ში ნაწერილი ინფორმაციის (პროგრამამდე ბილიკის) ამოშლა.



6sb. 435.

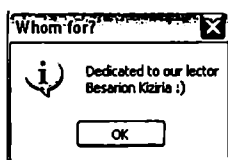


6sb. 436.

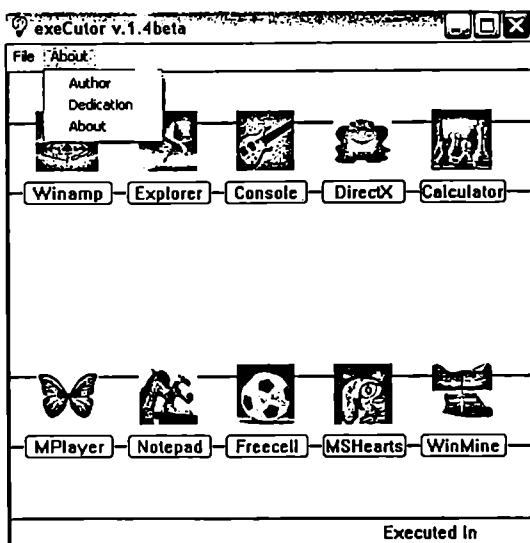
მენიუს სტრიქონის მეორე კომპონენტი ასევე ჩამოშლადი მენიუა და შეიცავს რამოდენიმე კომპონენტს. იხილეთ ნახ. 438.

ქვემენიუს კომპონენტ „Author“-ზე დაწკაპუნებით გამოდის ინფორმაცია პროგრამის ავტორის შესახებ.

ქვემენიუს მეორე კომპონენტ „Dedication“-ზე (მიძღვნა) დაწკაპუნებით ეკრანზე გამოდის ქვეფანჯარა, რომელშიც მოთავსებულია ავტორის მიძღვნის ტექსტი. იხ. ნახ. 437.



ნახ. 437.



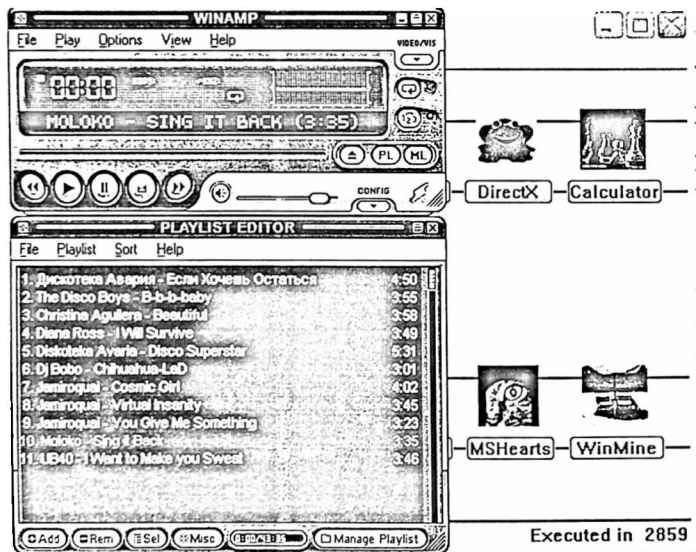
ნახ. 438.

ენახით პროლოგ-პროგრამა მოქმედებაში. ეოქვათ მომხმარებელს აქვს სურვილი გამოიძახოს პროგრამა „Winamp“, რომლის საშუალებითაც შესაძლებელია კომპიუტერზე მუსიკის გაშვება. მომხმარებელი მიმთითებლით აწკაპუნებს შესაბამის ლილავს. გამოდის „Winamp“ ქვეფანჯარა და ძირითადი ფანჯრის (ნახ. 4.34) მარჯვენა დაბალ კუთხეში მცირე ზომის ქვეფანჯარაში გამოდის ტექსტი „Executed in“ და ამ პროგრამის შესრულების დრო მიკროწამებში (2250 მკწ.). ასევე ც/ბ-ში ჩაიწერება ამ პროგრამამდე მისასვლელი ბილიკი (path).

file("C:\\Program Files\\Winamp\\winamp.exe",1453)

ბილიკი პროგრამამდე.

ნახ. ნახ. 4.39.



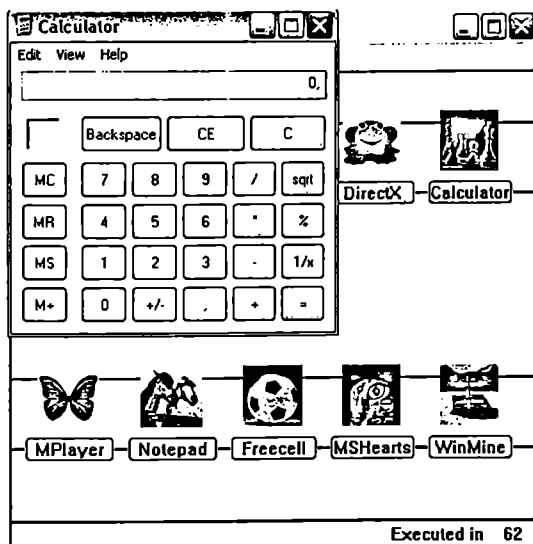
ნახ. 4.39.

კალკულატორის ღილაკზე დაჭერისას ფანჯარაში გამოდის კალკულატორი, ცისფერ სარკმელში აღინიშნება ამ პროგრამის გამოძახების დრო და ც/ბ-ში ჩაიწერება გზა კალკულატორის პროგრამამდე. იხ. ნახ. 4.40.

```
file("C:\\WINDOWS\\system32\\calc.exe",186)
```

ასევე შესაძლებელია ფანჯრის სხვა სტანდარტული პროგრამების გამოძახებაც.

პროგრამის სრული ტექსტი იხილეთ დანართი 3-ში.



ნახ. 4.40

ამოცანა 3. საქართველოს ისტორიული ძეგლები

პროგრამა წარმოადგენს მონაცემთა ბაზას, რომელიც შეიცავს ინფორმაციას საქართველოს ისტორიული ძეგლების შესახებ. პროლოგ-პროგრამაში შესაძლებელია სასურველი ძეგლის ძიება სხვადასხვა პარამეტრის მიხედვით: სახელი, ტიპი (ეკლესია, მონასტერი, სასახლე, ციხესიმაგრე), აშენების თარიღი (წელი), ძეგლის აშენების საუკუნე,

ქალაქი, სადაც იგი მდებარეობს, კუთხე (საქართველოს რომელ კუთხეშია აგებული), ქვეყანა (სადაც ეს ძეგლი მდებარეობს), ძეგლის ამშენებელი, დამაარსებელი (თუ ის ცნობილია).

ამოცანის შესრულებაზე გაშვებისას ც/ბ-დან ელვებულობით ინფორმაციას ამ ძეგლის შესახებ, მის ზოგად აღწერასა და პარამეტრებს. შედეგს თან ახლავს ძეგლის გრაფიკული ფაილი (ფოტოსურათი). ასევე შესაძლებელია ც/ბ-ის შევსება შესაბამისი ინფორმაციით. პროგრამის სრული ტექსტი და საკურსო გეგმარის აღწერილობა მოყვანილია მესამე დანართში.

ამოცანის სამუშაო ძირითად ფანჯარას აქვს ნახ. 4.41. მოცემული სახე.



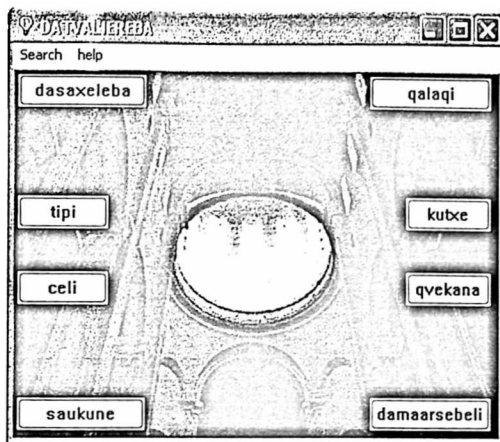
ნახ. 4.41. ამოცანის ძირითადი ფანჯარა

მენიუს სტრიქონი შედგება შემდეგი კომპონენტებისაგან:

„File“-შეიცავს „Exit“ ქვემენიუს, რომლის დანიშნულებაც ცნობილია.

მენიუს მეორე კომპონენტს „Windows“-ს გააჩნია ორი ქვემენიუ:

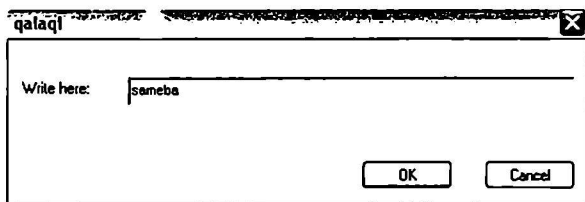
ქვემენიუ „DATVALIEREBA“-ზე მიმთითებლის დაწკაპუნებით ეკრანზე გამოდის ქვეფანჯარა, რომელიც ნახ. 4.42-ზეა გამოსახული



ნახ. 4.42. ცოდნის ბაზაში ინფორმაციის ძებნის ქვეფანჯარა

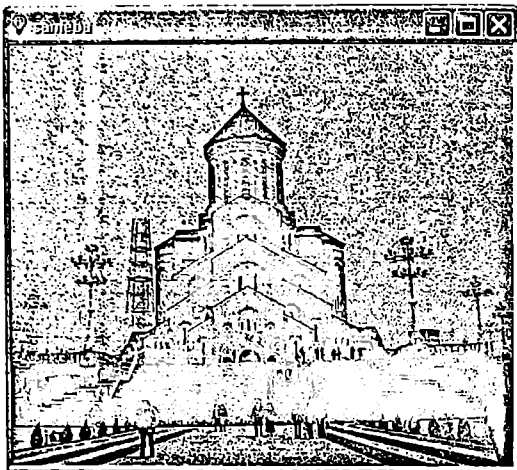
ამ ქვეფანჯრის მენიუს სტრიქონი შეიცავს „Search“ (ძებნა) კომპონენტს. თავის მხრივ მას ქვემენიუს სახით გააჩნია ისტორიული ძეგლის იგივე პარამეტრები, რომლებიც მოთაესებულია 4.42. ფანჯარაში: „Dasaxeleda“, „qalaqi“, „tipi“, „celi“, „saukune“, და ა.შ. ანალოგიური პარამეტრებით არის აღწერილი ისტორიული ძეგლი ასევე ც/ბ-იც.

ეთქვათ, პროგრამის მომხმარებელს აქვს მიზანი მოძებნოს ისტორიული ძეგლი მისი დასახელების მიხედვით. ამისათვის იგი დააწკაპუნებს მიმთითებლით ქვეფანჯრის „dasaxeleda“ ლიდაკზე. გამოდის ინფორმაციის შესატანი „Read“ ქვეფანჯარა. იხ. ნახ. 4.43.



ნახ. 4.43.

მომხმარებელი საინფორმაციო ველში შეიტანს მისთვის სასურველი ისტორიული ძეგლის სახელს, ეთქვათ „sameba“-ს. ეკრანზე გამოდის ძეგლის გრაფიკული ფაილი (ფოტოსურათი). იხილეთ ნახ. 4.44.



ნახ. 4.44.

ამასთან ერთად ინფორმაციის გამოტანის ქვეფანჯარაში გამოდის ინფორმაცია აღნიშნული ძეგლის შესახებ. იხილეთ ნახ. 4.45.

Compiling the file:
C:\istoriuli Zeglebi\istoriuli dzeglebi.PRO
0 errors, 0 warnings.

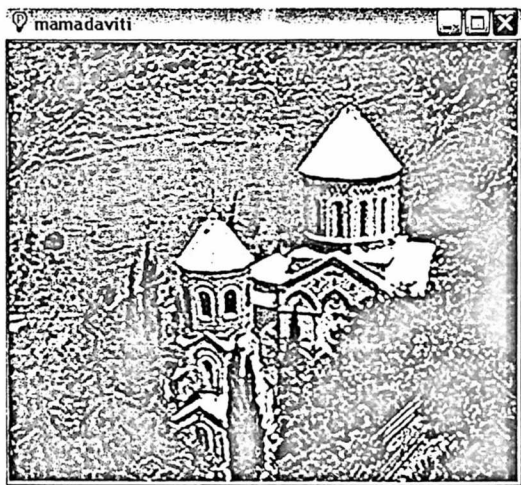
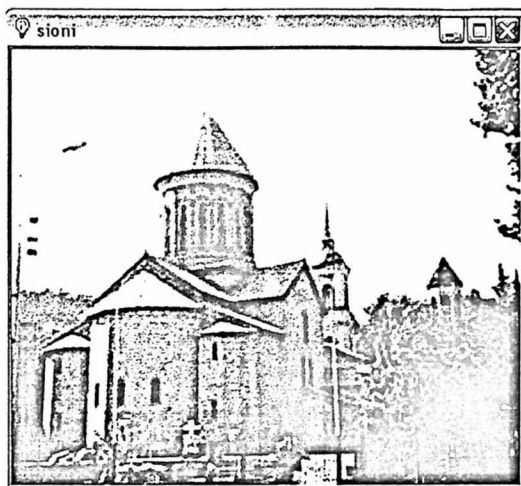
saaneba

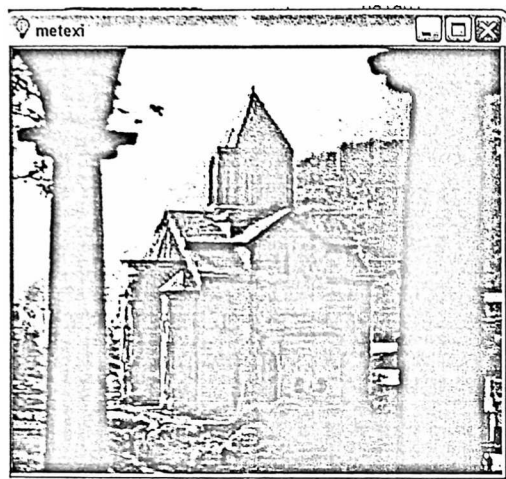
tipi	: tadzari
celi	: 1989-2005
saukune	: XX-XXI
qalaqi	: tbilisi
kutxe	: qartli
qvekana	: saqartvelo
daaaarsebeli:	

No.

ნახ. 4.45.

ეთქვათ პროგრამის მომხმარებლის შემდეგი მიზანია საქართველოს რომელიმე ქალაქში არსებული ისტორიული ძეგლების გაცნობა. ამისათვის მომხმარებელი ქვეფანჯარა „DATVALIEREBA“-ში მიმთითებლით დააწკაპუნებს ლიდაკზე „qalaqi“. ეკრანზე გამოსულ „Read“ ფანჯარაში მომხმარებელი ჩაწერს მისთვის სასურველი ქალაქის სახელს, ეთქვათ თბილისს. ეკრანზე გამოდის (კ/ბ-ში არსებული თბილისის ისტორიული ძეგლები. იხილეთ ნახ. 4.46. ამ ნახაზზე მოყვანილია მხოლოდ ნაწილი თბილისის ისტორიული ძეგლებისა.





ნახ. 4.46. თბილისის ისტორიული ძეგლები

გარდა გრაფიკული ფაილებისა (ფოტოსურათებისა) ინფორმაციის გამოტანის
გრაფიკული ქვეფანჯარაში გამოდის თბილისის ძეგლების სხეულები. მახასიათებლები. იხ.
 ნახ. 4.47.

manadaviti

tipi	: tadzari
celi	:
saukune	: VI
qalaqi	: tbilisi
kutxe	: qartli
qvekana	: saqartvelo
damaarsebeli	:

metexi

tipi	: tadzari
celi	: 1278-1289
saukune	: XIII
qalaqi	: tbilisi
kutxe	: qartli
qvekana	: saqartvelo
damaarsebeli	:

narikala

tipi	: cixesimagre
celi	:
saukune	:
qalaqi	: tbilisi
kutxe	: qartli
qvekana	: saqartvelo
damaarsebeli	:

ninocminda

tipi	: tadzari
celi	: 1550-1565
saukune	: XVI
qalaqi	: tbilisi
kutxe	: qartli

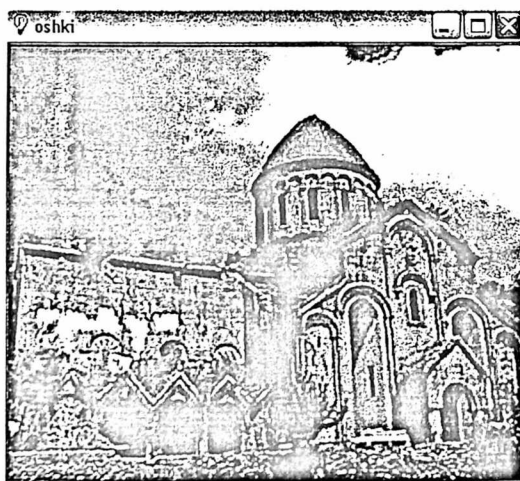
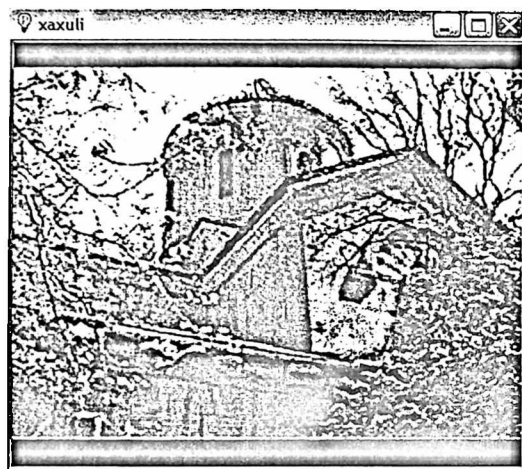
და ა.შ.

ნახ. 447.

„DATVALIEREBA“ ქვეფანჯარაში ღილაკზე „qvekana“ დაჭერისას შესაძლებელია მივიღოთ ვიზუალური და ტექსტური ინფორმაცია სხვა ქვეყნებში არსებული ქართული ისტორიული ძეგლების შესახებ. ასე მაგალითად, საინფორმაციო ფანჯრის „qvekana“ ეელში თურქეთის ჩაწერისას ეკრანზე გამოდის ინფორმაცია თურქეთის ტერიტორიაზე შემორჩენილი ქართული ხუროთმოძღვრების ძეგლების შესახებ. იხილეთ ნახ. 4.48. და ნახ. 4.49.

ამოცანის ძირითად ფანჯარაში „Istoriuli dzegelebi“ მოთავსებულია ღილაკი „DAMATEBA“, რომლის დანიშნულებაცაა ც/ბ-ის შევსება ახალი ინფორმაციით (ახალი ისტორიული ძეგლებით). ც/ბ-ში პროგრამის შემდგენელმა, *ჩვენი სპეციალისტის მაგისტრმა დავით ფარჯიკაძემ*, მოათავსა ინფორმაცია 50-ზე მეტი ისტორიული ძეგლის შესახებ. „DAMATEBA“ ღილაკზე მიმთითებლის დაწკაპუნებით გამოდის სპეციალურად შექმნილი ქვეფანჯარა, რომელიც ნახ. 4.50-ზეა გამოსახული. მომხმარებელი შეავსებს შესაბამის საინფორმაციო ბოქსებს საჭირო ჩასამატებელი ინფორმაციით და დააწკაპუნებს ღილაკზე

„SAVE“. ც/ბ შეიქმნა ფანჯარაში მოთავსებული ინფორმაციით. ც/ბ-ში ინფორმაციის ნაშთების შემდეგ 4.50. ქვეფანჯარა სპეციალური გრაფიკული პრედიკატების მეშვეობით სუფთავდება, ახალი ინფორმაციის მისაღებად.



ნახ. 4.48.

oshki

tipi	: tadzari
celi	: 958-961
saukune	: X
qalaqi	:
kutxe	: tao-klarjeti
qvekana	: turqeti
damaarsebeli	: adarnese kurapalati

tbeti

tipi	: tadzari
celi	:
saukune	: X
qalaqi	:
kutxe	: tao-klarjeti
qvekana	: turqeti
damaarsebeli	:

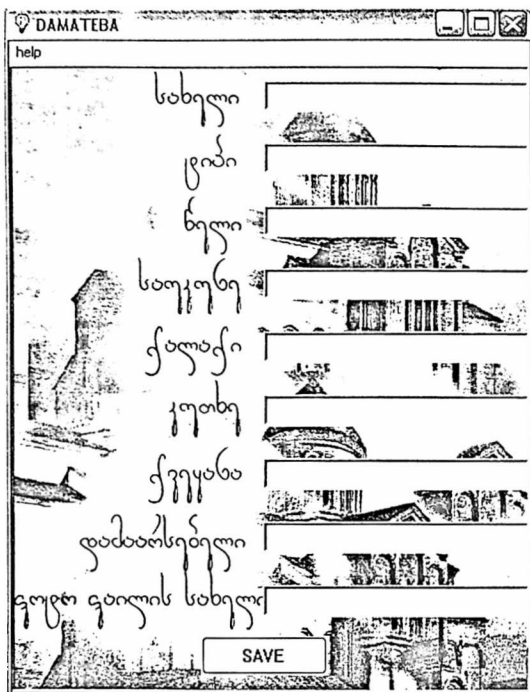
xaxuli

tipi	: monasteri
celi	:
saukune	: X
qalaqi	:
kutxe	: tao-klarjeti
qvekana	: turqeti
damaarsebeli	: davit kurapalati

ნახ. 4.49.

ძირითად ფანჯარაში „Istoriuli dzegelebi“, მენიუს კომპონენტ „Info“-ზე დაწკაპუნებით ეკრანზე გამოდის საინფორმაციო ფანჯარა საკურსო სამუშაოს შემსრულებლების შესახებ. იხ. ნახ.51.

დათვალიერების ქვეფანჯარაში ასევე მოთავსებულია სხვა დოკუმენტები: ნაშრომის ფორმატის შეზღუდულობის გამო ჩვენ ყველა მათგანს ვერ განვიხილავთ. დაინტერესებულ მკითხველს აქვს საშუალება ამოცანა გაუშვას სრულად და მიიღოს პასუხი ყველა შეკითხვაზე (ამოცანის სრული ტექსტი და მასთან მუშაობის ალწერილობა მოცემულია მესამე დანართში).



ნახ. 4.50

საქართველოს ტექნიკური უნივერსიტეტი

საერთო სამუშაო საგანში
ხელოვნური ინტელექტის საფუძვლები

საქართველოს ისტორიული ძეგლები

კურსი IV
ჯგუფი 1082312
სპეციალობა: კომპიუტერული სისტემები და ქსელები
სტუდენტები: ნატო სურგულაძე
დაცულ ფარულია
საქმისგანმცემი: ბეზინია

თბილისი, 2006წ.

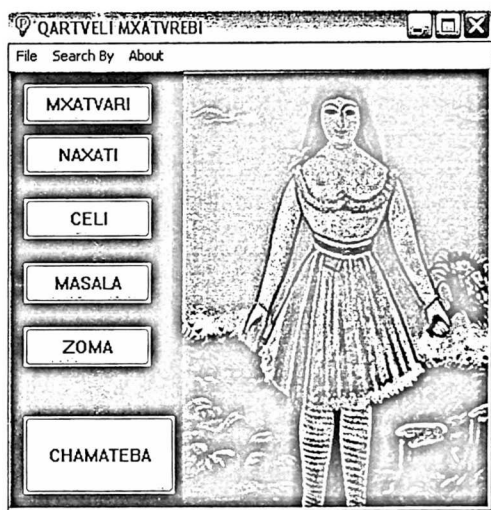
ნახ. 4.51

ამოცანა 4. ქართული მხატვრები

პროგრამა წარმოადგენს მონაცემთა ბაზას, რომელიც შეიცავს ინფორმაციას ქართული მხატვრების ნამუშევრების შესახებ. მომხმარებელს ეძლევა საშუალება მხატვრის სასურველი ნახატის ძიებისა ც/ზ-ში სხვადასხვა პარამეტრის მიხედვით: მხატვრის ეინობა, ნახატის შექმნის თარიღი, ნახატის ზომა, მასალა (ნახატის გრუნტი და გამოყენებული მასალა: აკვარელი, ზეთი და ა.შ.). ც/ზ-ში ძიების შედეგია არჩეული მხატვრის ნახატების მოკლე აღწერა, რომელსაც თან ახლავს გრაფიკული ფაილი (ნახატის ფოტოსურათი).

მოკლედ აღეწეროთ პროლოგ-პროგრამის მუშაობის პრინციპი.

პროგრამის შესრულებაზე გაშვებისას ეკრანზე გამოდის ამოცანის ძირითადი ფანჯარა, რომელიც ზემოთ განხილული გრაფიკული პრედიკატების ბაზაზე შექმნილი. იხ. ნახ. 4.52.

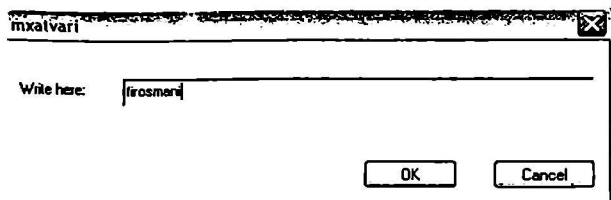


ნახ. 4.52.

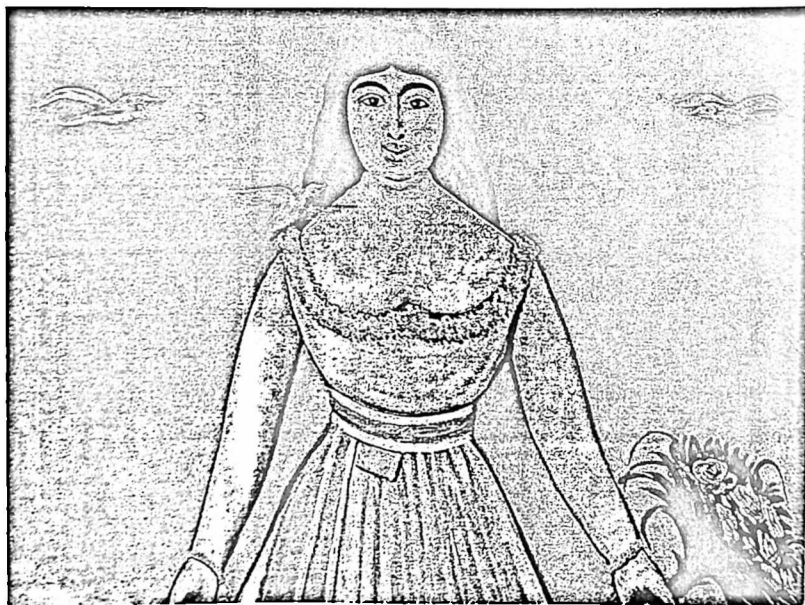
ძირითად ფანჯარაზე ფიროსმანის ნახატის ფონზე განლაგებულია ინფორმაციის საძიებელი ლილაკები: „MXATVARI“, „NAXATI“, „CELI“ და ა.შ. ასევე ფანჯარაში არის ლილაკი, „CHAMATEBA“, რომლის მეშვეობითაც წარმოებს ც/ზ-ის შევსება, ან მისი კორექტირება.

ლილაკზე „MXATVARI“ გიმთიიებლის დაწკაპუნებით ეკრანზე ჩნდება ინფორმაციის საძიებელი „Read“ ქვეფანჯარა (ნახ. 4.53). მასში მხატვრის ვინაობის (ეთქვათ ფიროსმანი) ჩაერისა და დადასტურების შემდეგ („OK“) ეკრანზე გამოდის ფიროსმანის ც/ბ-ში

მოთავსებული ნახატების ფოტოსურათები. იხ. ნახ.4.54. ინფორმაციის გამოტანის ქვეფანჯარაში გამოდის ტექსტური ინფორმაცია ამ ნახატების შესახებ. იხ. ნახ. 4.55.



ნახ. 4.53.





6sb. 454.

meezove

MXATVARI : firosmani

CELI : 1977

MASALA: <http://www.baiagallery.ge/art.html>

ZOMA: 100x100

aktrisa margarita

MXATVARI : firosmani

CELI : 1977

MASALA: <http://www.baiagallery.ge/art.html>

ZOMA: 90x100

6sb. 454.

ანალოგიური ტექნოლოგიით შესაძლებელია ეკრანზე გამოეიტანოს ც/ბ-ში მოთავსებული ნებისმიერი ქართული მხატვრის ნამუშევრები, მაგალითად ლადო გულიაშვილი. იხ. ნახ. 4.55.



ნახ. 4.55.

ასევე შესაძლებელია ც/ბ-ში მოთავსებული ქართული მხატვრების გალერეის დათვალიერება ნახატის შექმნის თარიღის, მასალის, ზომის და ა.შ. მიხედვით.

ლილაქზე „CHAMATEBA“ დაჭერით ეკრანზე გამოდის ჩამატების ქვეფანჯარა. იხ. ნახ. 4.56. ფანჯარაში ინფორმაციის შეტანისა და დადასტურების შემდეგ ც/ბ შეივსება ახალი ინფორმაციით.

მენიუს „About“ კომპონენტზე დაჭერით გამოდის საინფორმაციო ქვეფანჯარა პროგრამის შემქმნელების შესახებ. იხ. ნახ. 4.57.

CHAMATEBA

მხატვარი _____

ნახატი _____

წელი _____

მასალა _____

ზომა _____

ფაილი _____

SAVE

ნახ. 4.56.



ნახ. 4.57.

ამოცანით დაინტერესების შემთხვევაში შესაძლებელია მისი პროლოგ-პროგრამისა და აღწერის ნახვა დანართი 3-ში.

ამოცანა 5. მობილური ტელეფონების მოძიებები

პროდუქტ-პროგრამა რეაგირებას უკეთებს მობილური ტელეფონების (აპარატების) ცოდნის ბაზას. ც/ბ-ში შესაძლებელია მოიძებნოს მობილური ტელეფონების აპარატები ექსიდან ერთერთი პარამეტრის მიხედვით: **Name** (ფირმის სახელი), **Model** (მოდელი), აპარატის ფერი (**Color**), **Neto** (აპარატის წონა), აპარატის მეხსიერება (**Memory**), **Kamera** (კამერა). ც/ბ-ში ძებნა მთავრდება ეკრანზე გრაფიკული ფაილის (ფოტოსურათის) გამოტანით და **Output** ქვეფანჯარაში შედეგების მოთავსებით, რომელიც შეიცავს აპარატის პარამეტრებს.

ამოცანის შესრულებაზე გაშვების შემდეგ ეკრანზე გამოდის ძირითადი ფანჯარა, რომელიც მოცემულია ნახ. 4.58-ზე.

Brush Mobile

File Brush Mobile New window Help

Name

Model

Color

Neto

Memory

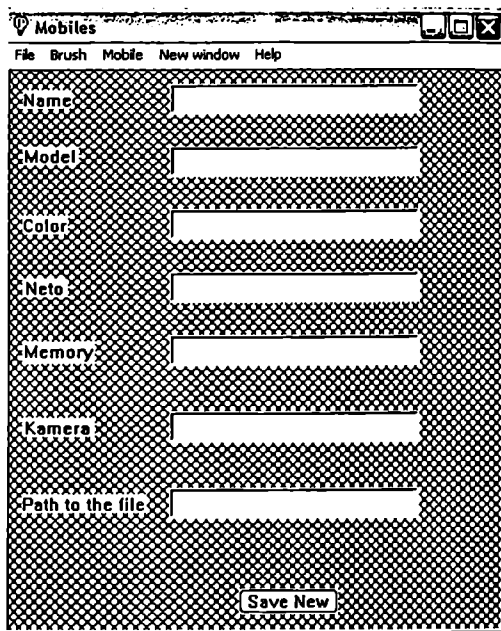
Kamera

Path to the file

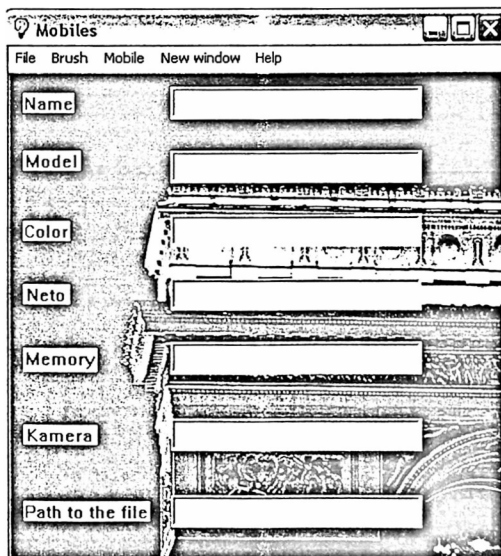
Save New

ნახ. 4.58.

მენიუს სტრიკონში „Brush“ (ფუნჯი) კომპონენტზე ყოველი დაწკაპუნებისას იცვლება ფანჯრის უკანა ფონი. იხ. ნახ. 4.59.

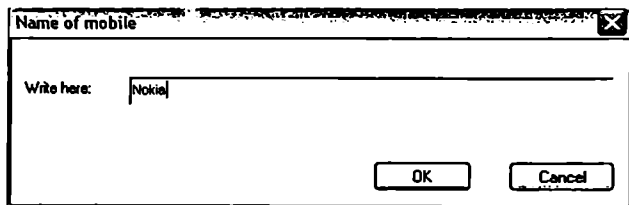


66b. 459.

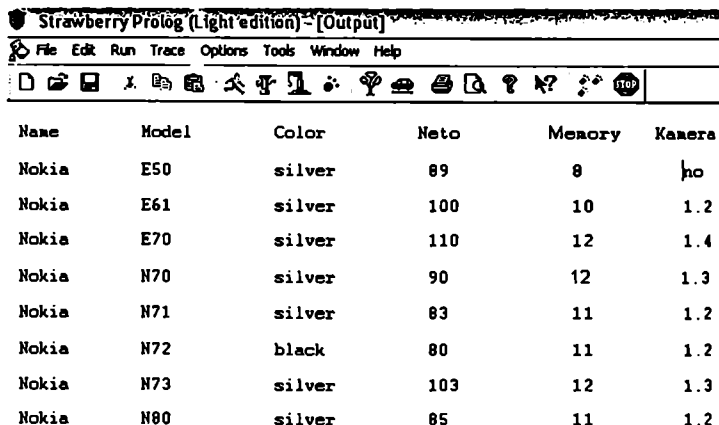


შენიშნული „Mobile“ პუნქტზე დატვირთი ჩამოიშლება მისი ქვემენიუ, რომელიც შეიცავს ც/ბ-ში მოთვსებული მობილური აპარატების პარამეტრებს. ქვემენიუ „Name“-ზე დაწკაპუნებით ეკრანზე გამოდის ინფორმაციის შეტანის ქვეფანჯარა. მასში (იხ. ნახ. 4.60.) ჩვენთვის საძიებელი მობილური აპარატის სახელის ჩაწერისა და დადასტურების შემთხვევაში ეკრანზე გამოდის შესაბამისი აპარატების გრაფიკული ფაილები (ფოტოსურათები). იხ. ნახ.

4.62. გარდა ამისა  Output ველში გამოიტანება ამ აპარატების პარამეტრები. იხ. ნახ. 4.61.



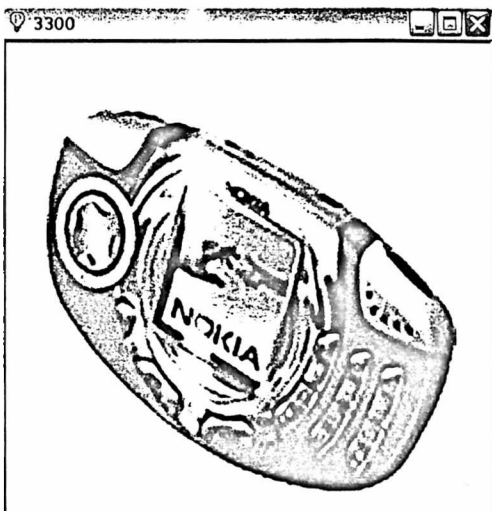
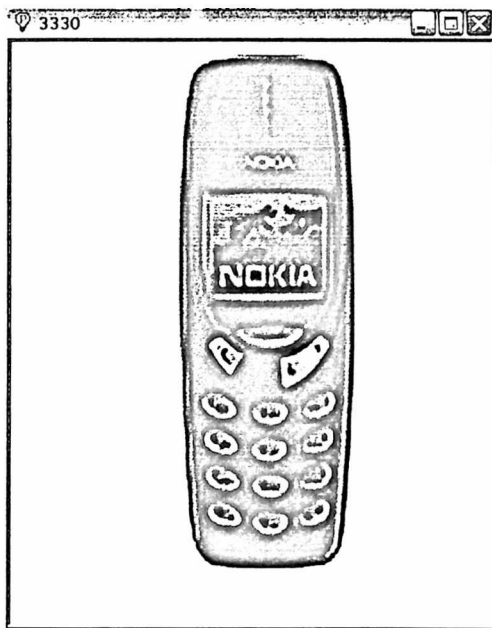
ნახ. 4.60.



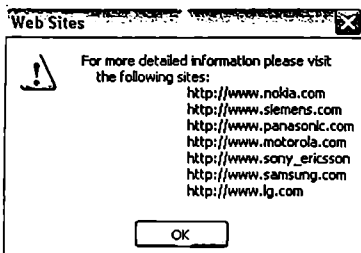
Name	Model	Color	Neto	Memory	Kamera
Nokia	E50	silver	89	8	no
Nokia	E61	silver	100	10	1.2
Nokia	E70	silver	110	12	1.4
Nokia	N70	silver	90	12	1.3
Nokia	N71	silver	83	11	1.2
Nokia	N72	black	80	11	1.2
Nokia	N73	silver	103	12	1.3
Nokia	N80	silver	85	11	1.2

ნახ. 4.61.

ძირითადი ფანჯრის საინფორმაციო ველებში (ნახ. 4.58.) სხვა დასახელების მობილური აპარატების შეტანისას ეკრანზე იმავე ტექნოლოგიით გამოდის მათი შესაბამისი ეიზუალური და ტექსტური ინფორმაცია. ეს ინფორმაცია თავსდება ამოცანის ც/ბ-ში.



მენიუს „Help“ პარამეტრი შეიცავს ორ ქვემენიუს, რომლებიც გვაწვდიან ინფორმაციას პროგრამის შემდგენლების შესახებ და ინტერნეტის იმ საიტების მისამართებს, სადაც დაინტერესებულ პირს შეუძლია მიიღოს დამატებითი ინფორმაცია მობილური აპარატების შესახებ. იხ. ნახ. 4.63.



ნახ. 4.63.

ამოცანის პროლოგ-პროგრამის სრული ტექსტი მოცემულია დანართი 3-ში.

ამოცანა 6 ელექტრო გიტარების ც/ბ-ის ფორმირება გრაფიკული პრედიკატების ბაზაზე

განვიხილოთ ამ ქვეთავის ბოლო ამოცანა, რომელიც ელექტრო გიტარების ც/ბ-თან მუშაობას შეეხება გრაფიკული პრედიკატების ბაზაზე.

ცოდნის ბაზაში მოთავსებულია ელექტრო გიტარების შემდეგი პარამეტრები:

Brand (დამამზადებელი ფირმა), **Model** (გიტარის მოდელი), **Color** (გიტარის ფერი), **Frets Number** (გიტარის ლადების რაოდენობა), **Wood** (ხის მასალა, რომლითაც დამზადებულია გიტარა), **Bridge** (სიმების დამჭერი).

ამოცანის პროლოგ-პროგრამის სრული ტექსტი მოყვანილია დანართი 3-ში.

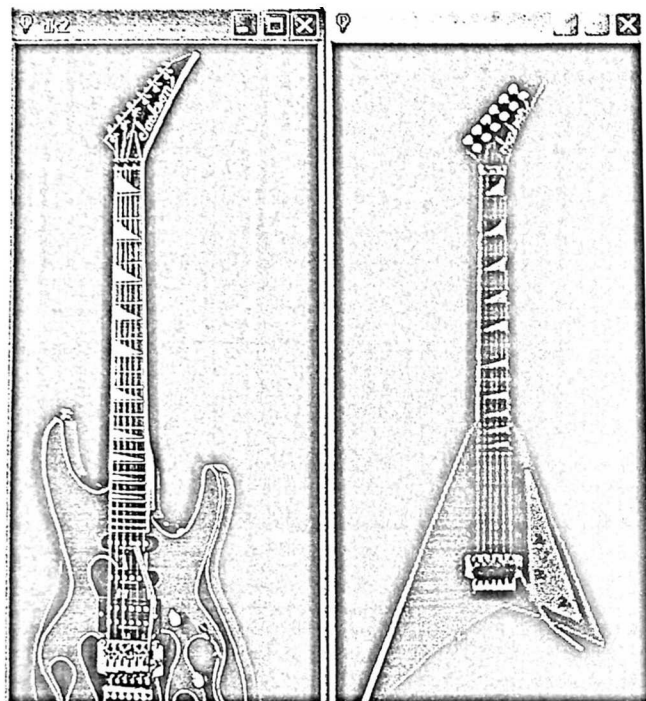
ამოცანის შესრულებაზე გაშვებისას ეკრანზე გამოდის ძირითადი ფანჯარა. იხ. ნახ. 4.64. მისი შექმნის ტექნოლოგია თითქმის ანალოგიურია ზემოთ განხილულ ამოცანებში ფანჯრების შექმნისა.

მომხმარებელი შედის მენიუს სტრიქონის „Guitars“ კომპონენტის, ეთქვით „Color“ ქვემენიუში. მასზე მიმთითებლის დაწკაპუნებით გამოდის ინფორმაციის შეტანის ქვეფანჯარა, რომელიც ნახ. 4.65-ზეა მოცემული. მასში ეთქვით red (წითელი) ჩაწერისას ც/ბ-დან ამოიკრეფება წითელი ფერის გიტარების გრაფიკული ფაილები (ფოტოსურათები).

ფოტოსურათები თავსდება ეკრანზე და პროგრამის მომხმარებელს ეძლევა საშუალება გადაათვალიეროს ისინი. ამასთან ეკრანის **Output** ქვეფანჯარაში გამოდის შესაბამისი გიტარების ზემოთ მოყვანილი პარამეტრების მნიშვნელობები. იხ. ნახ. 4.66. და ნახ. 4.67.

ნახ. 4.64.

ნახ. 4.65.



Tab. 4.66.

Brand	Model	Color	Frets	Wood	Bridge
bcrich	beast	red	24	alder	floyd rose
bcrich	bronze	red	24	beech	static bridge
fender	showmast	red	22	line	synchronized tremolo
gibson	blueshav	red	22	mahogany	static bridge
gibson	firebir	red	22	beech	tune-o-matic
gibson	sgs	red	22	mahogany	tune-o-matic
ibanez	rgal21	red	24	oha	edge pro
jackson	dk2	red	24	beech	double locking FR
jackson	firesto	red	24	mahogany	floyd rose
jackson	rr3	red	21	alder	synchronized tremolo

No

Tab. 4.67.

ცოდნის ბაზაში ინფორმაციის ჩასამატებლად გამოიყენება ღილაკი „Save“. მასზე მიმთითებლის დაწკაპუნებით, ძირითად ფანჯარაში ტექსტ-ბოქსებში ჩაწერილი ინფორმაცია ჩაიწერება ც/ბ-ის ბოლოში, ხოლო ტექსტ-ბოქსები სუფთადება ინფორმაციიდან ახალი ინფორმაციის შესატანად.

ამრიგად, განხილული ამოცანების საფუძველზე შესაძლებელია დავასკენათ შემდეგი. ხი-ის სისტემების დაპროექტებისას, პროგრამულ ინსტრუმენტად სახურველია გამოვიყენოთ დეკლარაციული ტიპის ალგორითმული ენა. კერძოდ, ჩვენ ვიყენებთ ამ კლასის ერთერთ საუკეთესო პროლოგ-იმიტატორს: „Strawberry Prolog“. განხილული ამოცანები ფართოდ იყენებენ რელაციურ ც/ბ-ს, გრაფიკულ ოპერატორებს.

შემდეგ თავში განვიხილავთ პროგრამა-ტესტების შედგენის მეთოდებს. მათი შედგენისას ასევე გამოყენებული იქნება ზემოთ აღწერილი ტექნოლოგიები.

თავი 5. ტესტ-პროგრამების შედგენა „Strawberry Prolog“ იმიტატორის ბაზაზე

თანამედროვე ეტაპზე ადამიანის მოღვაწეობის თითქმის ყველა სფეროში გამოიყენება ტესტირება. ეს სფეროებია: სამუშაოს მაძიებლის ტესტირება სამსახურში მიღებისას, უმაღლეს სასწავლებლებში მისაღები გამოცდების ჩაბარება, სხვადასხვა მოწყობილობების დიაგნოსტიკა-ტესტირება (აეტიომობილი, აუდიო ვიდეო აპარატურა, საფრენი აპარატები და ა.შ.), მედიცინა (კომპიუტერული ტომოგრაფები, ექოსკოპია) და ა.შ. ტესტ-პროგრამები შედგენილია სხვადასხვა პროგრამული საშუალებებით (გამოიყენება როგორც იმპერატიული ასევე დეკლარაციული ალგორითმული ენები). ამ თავში მოყვანილი ტესტ-პროგრამები შედგენილია ლოგიკური პროგრამირების ალგორითმული ენის „პროლოგის“ ბაზაზე. მოცემული სახელმძღვანელოს დანიშნულებიდან გამომდინარე, ჩვენს მიერ ტესტ-პროგრამები დაწერილია „Strawberry Prolog“-ზე.

საქართველოში დღეისათვის მიმდინარეობს განათლების სფეროს რეფორმა. საქართველოს ტექნიკურ უნივერსიტეტში, კერძოთ კი ჩვენს ფაკულტეტზე (ინფორმატიკისა და მართვის სისტემები) ჯერ კიდევ ხუთი წლის წინ, სტუდენტთა და მაგისტრთა სემესტრული შეფასებისათვის დინერგა ე.წ. „მანნესტერის სისტემა“. მისი არსი შემდეგში მდგომარეობს. შემოღებული იქნა შეფასების 100 ქულიანი სისტემა. შესასწავლი დისციპლინები დახარისხდა სხვადასხვა ნიშნულის ე.წ. „კრედიტების“ საშუალებით. მოცემული მომენტისათვის ფაკულტეტზე ისწავლება ხუთი და ათი კრედიტის მქონე საგნები. რასაკვირველია დისციპლინების ასეთი რანჟირება ეფუძნება საგნის მოცულობას, მის დანიშნულებასა და მნიშვნელობას მოცემული სპეციალობისათვის და სხვა ფაქტორებს. ჩვენი საგანი „ხელოვნური ინტელექტის საფუძვლები“ ათ კრედიტისა, ორ სემესტრისა და სამი სემესტრისაა. ორივე სემესტრში ტარდება 2 რეიტინგული შეფასება და საგამოცდო სესიისას გამოცდა. სემესტრული პირველი რეიტინგული შეფასება 30 ქულიანია, მეორე შეფასება 60 ქულიანი. დარჩენილი 10 ქულა საგნის მასწავლებლის „სარეზერვო ფონდია“. მასში გათვალისწინებულია სტუდენტის (მაგისტრის) ლექცია სემინარებზე, დასწრების პერიოდულობა, მისი ქცევა და ა.შ.

„კომპიუტერული სისტემებისა და ქსელების“ კათედრაზე (ისევე როგორც სხვა მაპროფილებელ კათედრებზე) პროფესორ-მასწავლებელთა მიერ შედგენილია თვითოეული დისციპლინის სასწავლო გეგმა (Curriculum) და მისი მოკლე შინაარსი (Syllabus). მოცემული სახელმძღვანელოს ერთერთი თანაავტორის ბ. კიზირიას მიერ შედგენილი ზემოთ მოყვანილი დოკუმენტი იწინახება კათედრის ელექტრონულ ბიბლიოთეკაში. ნებისმიერ დაინტერესებულ პირს აქვს შესაძლებლობა გაეცნოს მას.

სტუდენტთა და მაგისტრთა შეფასების ბოლო ხუთწლიანი გამოცდილება იძლევა შემდეგი დასკვნის საშუალებას. სტუდენტთა დიდი ნაკადის შემთხვევაში (2005-2006

სასწავლო წელს ჩვენი სპეციალობის IV კურსზე ამ საგანს სწავლობდა დაახლოებით 130 სტუდენტი. მათი რიცხვი მიმდინარე 2006-2007 სწ-ს თითქმის ორმაგდება) სემესტრული რეიტინგული შეფასების პროცესი გარკვეულად გართულებულია, როგორც სტუდენტისთვის ასევე პედაგოგისთვის. ეს გამოიხატება შემდეგში. რეიტინგული შეფასებები და გამოცდაც მიმდინარეობს წერითი ფორმით. ჩვენდა სამწუხაროდ, სტუდენტების ცოდნის მარაგისა და ძირითადად მათი მენტალიტეტიდან გამომდინარე, ადგილი აქვს ე.წ. „შპარგალკების“ გამოყენებას. ეს კი აძნელებს მათი ცოდნის ადეკვატურ შეფასებას. გარდა ამისა, ნაშრომების გასწორების პროცესი მეტად შრომატევადია და გამსწორებელს უკმნის გარკვეულ დისკომფორტს.

ყოველივე ზემოთ თქმულიდან გამომდინარე, მოცემული ნაშრომის ავტორთა ჯგუფმა გადაწყვიტა ამ დისციპლინის რეიტინგული შეფასებისათვის გამოყენებინა სტუდენტთა და მაგისტრთა ტესტირების მეთოდი. ამისათვის პროლოგზე დაიწერა ე.წ. პროგრამა-გარსი (მმართველი პროგრამა). ტესტირების მეთოდი აპრობირებული იქნა 2005-2006 სწ-ის მეორე სემესტრში. ტესტირების შედეგი დამაკმაყოფილებელია.

სტუდენტი ინდივიდუალური წესით, კომპიუტერზე სათანადო პროგრამის გაშვების შემდეგ, დასმულ შეკითხვაზე პასუხს იძლევა ალტერნატიული პასუხების სიმრავლიდან ერთერთის არჩევით. პროგრამა აფიქსირებს პასუხს და ახდენს მის შეფასებას სათანადო ქულით. ტესტირების დასრულებისას კომპიუტერის ეკრანზე გამოდის სტუდენტის ტესტირების შედეგი. პედაგოგის ჩარევა ამ პროცესში მინიმალურია (მაგალითად, იგი ერთგვარაა იმ შემთხვევაში, თუ შეკითხვა ან ალტერნატიული პასუხები სტუდენტისთვის გაუგებარია).

ტესტირების შედეგების ანალიზმა ცხადყო შემდეგი: თითქმის გამოირიცხა „შპარგალკები“-ს გამოყენება. ტესტირების პროცესი მიმდინარეობს დამშვიდებულ, არანეეროზულ ვითარებაში. პედაგოგის ჩარევა და მისი ურთიერთობა სტუდენტთან დაყვანილია მინიმუმამდე, აქედან გამომდინარე დადებითი შედეგით. სტუდენტი უშუალოთ ჩართულია ტესტირების პროცესში და ეს პროცესი გარკვეულად წააგავს თამაშს. შედეგი მყისიერადაა ცნობილი როგორც პედაგოგისთვის, ასევე სტუდენტისთვის, შესაბამისად პრეტენზიებიც მინიმუმამდეა დაყვანილი. ტესტირების პროცესის დრო მაქსიმალურად შემცირებულია. ყველაზე მნიშვნელოვანი კი შემდეგი ფაქტორია. შეფასების პროცესს აქვს ცივილური ფორმა, რაც სრულად აკმაყოფილებს განათლების სისტემისადმი წაყენებულ დღევანდელ მოთხოვნებს.

ამრიგად, სტუდენტთა და მაგისტრთა რეიტინგული შეფასების ტესტურმა ფორმამ გაამართლა ჩვენი მოლოდინი და მოგვცა დადებითი შედეგები. ჩვენს მიერ შედგენილია ტესტების რამოდენიმე ვარიანტი, რომელიც დაახლოებით შეიცავს 200-მდე შეკითხვას, შესაბამისი ალტერნატიული პასუხების სიმრავლით. მიმდინარეობს შეკითხვებისა და ალტერნატიული პასუხების დახვეწა.

ჩვენი ტესტ-პროგრამის (გარსის) დადებით მხარედ შესაძლებელია ჩათვალოს ის მომენტიც, რომ იგი დაწერილია იმავე ალგორითმულ ენაზე, რომლის იმიტაციის „Strawberry Prolog“-ის უახლეს (2005 წლის) ვერსიას ვიყენებთ პროგრამულ ინსტრუქციებზე, გარსის პროგრამის დაწერისას გამოყენებული პროლოგის ისეთი ძირითადი ოპერატორები და მექანიზმები, როგორებიცაა: მონაცემთა წარმოდგენის სისებრი სტრუქტურა, რეკურსის მექანიზმი, გლობალური ცვლადები, გრაფიკული ოპერატორები (პრედიკატები) და სხვა.

ტესტის შეკითხვები მოიცავს „ხელოვნური ინტელექტის საფუძვლების“ თეორიულ, ასევე პრაქტიკულ საკითხებს. ტესტის შეკითხვების შერჩევა ძირითადად დაეფუძნა მოცემული ნაშრომის ერთერთი თანაავტორის ბ.კიზირიას [7] ნაშრომს. შეკითხვებისა და პასუხების ცოდნის ბაზის დაპროექტებაში დიდი დახმარება გაგივიწიეს ჩვენი სპეციალისტების სტუდენტებმა და მაგისტრებმა.

დაპროექტებული გარსის გამოყენება შესაძლებელია „კომპიუტერული სისტემებისა და ქსელების“ კათედრის სხვა დისციპლინებშიც, რასაკვირველია თუ ამის სურვილი ექნებათ შესაბამისი საგნების პედაგოგებს. შესაძლებელია ტესტ-პროგრამის (გარსის) ტირაჟირება და მისი გამოყენება ჩვენი ფაკულტეტის სხვა სპეციალისტებზე, გარსის პროგრამა მოცულობით საკმაოდ კომპაქტურია. ქვემოთ მოვიყვანოთ გარსის დაწერილობითი აღწერა.

5.1. სტრუქტურა და მამბისტრატა რეპრეზენტაციული შეშასების ტესტ-პროგრამის (გარსის) აღწერა

მოვიყვანოთ ჩვენს მიერ დაპროექტებული ტესტ-პროგრამის (გარსის) მუშაობის პრინციპის აღწერა.

პროლოგზე დაწერილი გარსის პროგრამის ტექსტი მოყვანილია ქვემოთ:

%**** TEST-PROGRAMA (GARS) SHESAVALI.*****

%***** SAQARTVELOS TEQNIKURI UNIVERSITETI*****

%**KOMPIUTERULI SISTEMEBISA DA QSELEBIS KATEDRA**

%***** XELOVNURI INTELEQTIS SAFUDZVLEBI *****

%***** PROGRAMIS SHEMDGENELI: PROF. BESIKI KIZIRIA *****

%***** TBILISI. 2006 CELI *****

?-

G_Sum=0,

G_Num=1,

```

G_Variant:=V,
question(G_Num,Q|G_Variant),

G_Question:=Q,

window(_,_,win_func(),"Question"+print(G_Num),100,100,600,300).
%-----

win_func(init):-
static(_,_,fail(),G_Question,20,20,500,70),

menu(normal,_,_menu_about(),"About"),

make_but(G_Variant,70).
%-----

menu_about(press):-
message("About", "Test-programa: Xelovnuri intelegtis shesavali. ", i).
%-----

make_but([_|_|T],S):-
button(_,_,but_func(),A,80,S,400,25),

S1 is S+40,

make_but(T,S1).
%-----

but_func(press):-
N is get_text(),

add_sum(N,G_Variant),

close_window(parent()),

G_Num:=G_Num+1,

G_Variant:=V,

(question(G_Num,Q|G_Variant)->

G_Question:=Q,

window(_,_,win_func(),"Question"+print(G_Num),100,100,600,300)).
%-----

```

```

add_sum(N,[[N,P]]):-
G_Sum:=G_Sum+P,
write(G_Sum),nl,
G_Num=30->

message("Finish", "tqveni Sedegia: "+print(G_Sum),!).
%-----

add_sum(N,[_V]):-
add_sum(N,V).
%-----

% ***** CODNIS BAZA *****

%***SHEKITXVEBI DA MATI SAVARAUDO PASUXEBI***

question(1,"Ramdeni taobis kompiuteri arsebobs?",["Erti",0],["Sami",0],["Otkhi",1]).

question(2,"Rogor intelegts ucodeben adamianis intelegts?",["Khelovnur
intelegts",0],["Bunebriv intelegts",1],["Arc erts",0]).

question(3,"Ramden klasadaa pirobitad dakofili algoritmul
enebi?",["Orad",1],["Samad",0],["Khutad",0]).

question(4,"Romel algoritmul enebis klass miekutvneba
prologi?",["Imperatiuls",0],["Deklaraciuls",1],["Orives",0]).

question(5,"Sitkva kibernetes qartulad nishnavs:",["Shromis xelovnebas",0],["Martvis
xelovnebas",1],["Arc erts",0]).

question(6,"Vin itleba pirobitad pirvel programistad?",["Charlz bebiji",0],["Jon fon
neimani",0],["ada lavleisi",1]).

question(7,"Ra erteulit izomeba adamianis
intelegti?",["Bitebit",0],["Baitebit",0],["Lipsi",1]).

question(8,"ASCII kodebshi kovel simbolos sheesabameba:",["Ertbaitiani
veli",1],["Orbaitiani veli",0],["otkhbaitiani veli",0]).

question(9,"Chamotvilitagan romelia imperatiuli algoritmul
eaa?",["asembleri",1],["prologi",0],["Arc erti",0]).

question(10,"Rogor logikas efudzneba adamiani daskvnis migebisas?",["Aracxad
logikas",1],["Binarul logikas",0],["Orives",0]).

```

question(11,"Ra aris aritmetikul-logikuri mockobilobis mtavari kvandzi?",["Amjamavi",1],["Registri",0],["Procesori",0]).

question(12,"programebisa da monacemebis droebiti shenaxvisa da damushavebisatvis gamoikeneba:",["Operatiuli mexsiereba",1],["vinchesteri",0],["kheshe mekhsiereba",0]).

question(13,"Prologs gaachnia:",["Transliatori",0],["Imitatopri",1],["Orive",0]).

question(14,"Khelovnuri intelektis ganvitareba daakhloebit itvlis-",["50 cels",1],["10 cels",0],["30 cels",0]).

question(15,"Romel garemoshi mushaobs prologis Strawberry versia?",["MS DOS garemoshi",0],["Windows garemoshi",1],["Unix garemoshi",0]).

question(16,"Rogori operaciaa inversia?",["Logikuri shekreba",0],["Logikuri uarkofa",1],["Logikuri gamravleba",0]).

question(17,"Vin sheqmna pirveli gamomtvleli manqana?",["Karel chapekma",0],["Ada lavleisma",0],["Charlz bebidjma",1]).

question(18,"Romeli araa adamianis azrovnebis forma?",["Cnobieri",0],["Aracnobieri",0],["Binaruli logika",1]).

question(19,"Khelovnuri intelektis daproeqtibisas romel etapze kalibdeba codnis baza?",["Pirvel",1],["Meore",0],["Mesame",0]).

question(20,"Ramdeni nacilisagan shesdgeba Prolog programa?",["Sami",1],["Otkhi",0],["Khuti",0]).

question(21,"Ristvis gamoikeneba grafebis teoria?",["Monacemta bazis dasaproeqtblad",0],["Cesebis chamosakalibblad",0],["dzebnis metodebis shemushavebisatvis",1]).

question(22,"Rogor codnis bazებს ecodbat relაციური codnis ბაზები?",["Programis tanshi chashenebul codnis bazებს",1],["Damxmare codnis bazებს",0],["Aseti gansazgvra ar arseobs",0]).

question(23,"Mat vinc apoeqtoben intelektualur sistemებს ecodbat:",["Eqspertebi",0],["Codnis indjinrebi",1],["Arc erti",0]).

question(24,"Kompiuteri funqcionirebs:",["Aracxadi logikit",0],["Unaruli logikit",0],["Binaruli logikit",1]).

question(25,"Konkretuli saproblemo arisatvis sheqmnil sistemas ecodba:",["Eqspertuli sistema",1],["Codnis injinruli sistema",0],["Diletanturi sistema",0]).

question(26,"Rogori shecdomaa semantikuri?",["Shinaarsobrivi",1],["Gramatikuli",0],["Orive",0]).

question(27,"Rogori shecdomaa sintaqsuri?",["Shinaarsobrivi",0],["Gramatikuli",1],["Arc erti",0]).

question(28,"Vin anxorcielebs codnis bazebis ganaxlebas?",["Qselebis administratori",0],["Bazebis administratori",1],["Momkhmarebeli",0]).

question(29,"Romeli mecnieba ukavshirdeba jorj bulis saxels?",["Bulis aritmetika",0],["Bulis algebra",1],["Bulis geometria",0]).

question(30,"Ra ecodeba adamianis tavis tvinis ujredeb?",["Eleqtronebi",0],["Neironebi",1],["Neitronebi",0]).

.....

მოვიყვანოთ ტესტ-პროგრამის (გარსის) აღწერა.

როგორც თითქმის ყველა პროლოგ-პროგრამა, ტესტ-პროგრამაც შედგება სამი ნაწილისგან: ამოცანის ცოდნის ბაზისგან, რომელიც შეიცავს ტესტის შეკითხვებს და მათზე საგარაუდო პასუხების სიმრავლეს. ც/ბ-ის ეს კომპონენტები პროგრამაში აღწერილია პროლოგში გამოყენებული მონაცემთა წარმოდგენის სიისებრი სტრუქტურით (იხ. [7]-ის მეხუთე თავი). როგორც თეორიული კურსიდანაა ცნობილი, სია შედგება სიის თავისა და ტანისგან. ჩვენს შემთხვევაში ტესტის შეკითხვა და საგარაუდო პასუხები ამ ფორმითაა წარმოდგენილი. ასე მაგალითად, 23 შეკითხვა:

question(23,"Mat vinc aproeqteben intelegtualur sistemebs ecodebat:",["Eqspertebi",0],["Codnis indjinrebi",1],["Arc erti",0]).

შედგება პრელიკატისგან: „question“ (შეკითხვა), შეკითხვის რიგითი ნომრისგან 23, უშუალოდ შეკითხვის ტექსტისგან: "Mat vinc aproeqtebs int eleqtualur sistemebs ecodebat:" და საგარაუდო სამი პასუხისგან, რომლებიც წარმოდგენენ სიისებრი სტრუქტურას:

["Eqspertebi",0],["Codnis indjinrebi",1],["Arc erti",0]).

ამავე სიის კომპონენტია პასუხის შეფასების შესაბამისი ქულა. ჩვენს ტესტში სწორი პასუხისათვის იჭრება 1 ქულა, ხოლო არასწორი პასუხი ფასდება 0 ქულით.

გარსის პროლოგ-პროგრამაში შემოტანილია ოთხი გლობალური ცვლადი:

ცვლადი **G_Sum** აღნიშნავს სტუდენტის მიერ დაგროვილი ქულების რაოდენობას.

ცვლადი **G_Num** აღნიშნავს შეკითხვის რიგით ნომერს, რომელიც ამ ცვლადს მიენიჭება სათანადო მომენტში. ამოცანის საწყის ეტაპზე მას ენიჭება ერთი მნიშვნელობა:

G_Num = 1.

ცვლადს **G_Variant** ენიჭება სტუდენტის მიერ შერჩეული რომელიმე საგარაუდო პასუხი.

ცვლადს **G_Q**-ს ენიჭება უშუალოდ მიმდინარე შეკითხვის ტექსტი.

სოფადი ფორმით, ც/ბ-ში მოთავსებული ყველა შეკითხვა სიისებრი სტრუქტურით შესაძლებელია ჩაიწეროს შემდეგნაირად:

question(G_Num, Q | G_Variant).

სიისებრი სტრუქტურის არსიდან გამომდინარე, ამ კონკრეტულ შემთხვევაში სიის თავია: G_Num, Q, ანუ ტესტის შეკითხვის რიგითი ნომერი და თვით მიმდინარე შეკითხვა, ხოლო სიის ტანია პასუხის შესაძლო ვარიანტები (ჩვენს შემთხვევაში პასუხის სამი ვარიანტი).

ჩვენს მიერ ტესტირების პროცესის ვიზუალური მხარის უზრუნველსაყოფად გამოყენებული იქნა წინა თავებში აღწერილი პროლოგ-იმიტატორის გრაფიკული ოპერატორები (პრედიკატები). პროგრამულად შექმნილია ტესტირების ფანჯარა. შემდეგი მიზნის საშუალებით (იხილეთ ტესტ-პროგრამის ტექსტი):

?

...

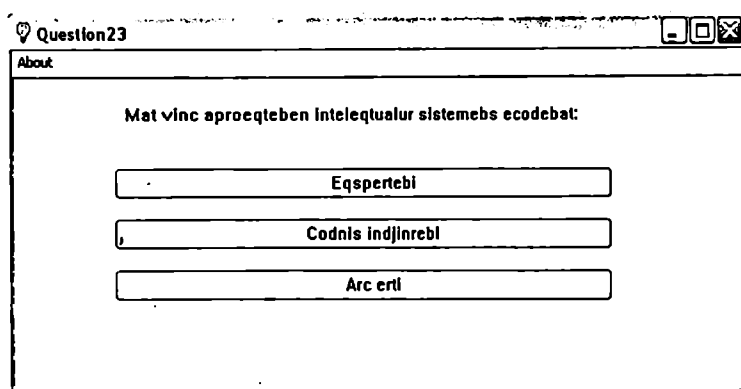
window(,win_func(),"Question"+print(G_Num),100,100,600,300).

დასახულია შემდეგი მიზანი: შეიქმნას ფანჯარა, რომლის სათაურის სტრიქონში მოთავსდება სათაური „Question“ და მიმდინარე შეკითხვის რიგითი ნომერი „G_Num“.

პროლოგზე ეს შემდეგნაირად ჩაიწერება:

“Question”+print(G_Num)”. აქ გრაფიკული პრედიკატი „print“ T ტერმს გარდაქმნის S სტრინგად ან პირიქით, ანუ ფანჯრის სათაურში სიტყვა „Question“-ს მიეწერება მიმდინარე შეკითხვის რიგითი ნომერი „G_Num“.

ეკრანზე გამოსატანი ფანჯრის მარცხენა ზედა წვეროს კოორდინატებია: 100, 100 (ფიქსელზე), ხოლო მისი ზომებია: 600 X 300-ზე. ფანჯარას აქვს ნახ. 5.1-ზე მოცემული სახე:



ნახ. 5.1. ტესტ-პროგრამის ძირითადი სამუშაო ფანჯარა

ფანჯრის ძირითად ფუნქციას „win_func()“-ს საჭიროებისამებრ ეგზავნება სხვადასხვა შეტყობინება (message).

გრაფიკული პრედიკატების მეშვეობით ფორმირდება ფანჯრის კომპონენტები:

სტატიკური ტექსტის ქვეფანჯარა, რომელშიც ტესტის შეკითხვის ტექსტი თავსდება.

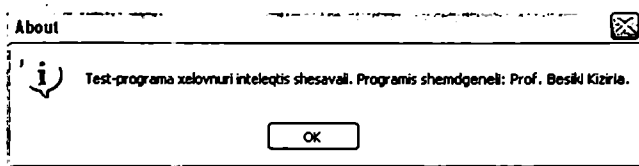
აუცილებელია აღინიშნოს ერთი მნიშვნელოვანი გარემოებაც, ძირითად ფანჯარაში პროგრამულად ფორმირებული ყველა კომპონენტი, მათ შორის სტატიკური ტექსტიც, თავსდება ასევე ქვეფანჯარაში. Window-სი ერთმანეთში ჩაშენებული ფანჯრების სისტემა სტატიკური ტექსტის ქვეფანჯრის უკანა ფონი თეთრი ფერისაა, მისი ფერი (თეთრი) ემთხვევა ძირითადი ფანჯრის ფონის ფერს (თეთრს). ამიტომ იგი უჩინარია. როგორც ცნობილია, ფანჯრის ფონის ფერი ფორმირდება გრაფიკული პრედიკატის „window brush“-ის მეშვეობით. ამოცანის მიზნიდან გამომდინარე, იგი ამ კონკრეტულ შემთხვევაში არ გამოიყენება.

პროლოგ-პროგრამის ეს ფრაგმენტი მოცემულია ნახ. 52-ზე:

```
static(,fail(),G_Question,20,20,500,70),  
menu(normal,_,menu_about(),"About"),  
make_but(G_Variant,70).
```

ნახ. 52.

პირველი გრაფიკული პრედიკატი: static(, _ fail(), G_Question, 20, 20, 500, 70), ფორმირდება უხილავი ქვეფანჯარა სომით: 500 X 70-ზე, რომელშიც თავსდება ტესტის მიმდინარე შეკითხვა G_Question="Mat vinc aproeqteben inteletualur sistemebs ecodebat:". მენიუს სტრიქონში გვაქვს ერთი კომპონენტი: „About“, რომელიც იძლევა ინფორმაციას პროგრამის შემდგენელის შესახებ (იგი ნახ. 52-ზე მოყვანილი პროგრამის ფრაგმენტის მეორე პრედიკატით „menu“ ფორმირდება). იხ. ნახ. 53.



ნახ. 53.

პროგრამის ფრაგმენტის მესამე კომპონენტი make_but ფანჯარაში აფორმირებს დიალ.კ, რომელზეც ტესტის მიმდინარე შეკითხვა თავსდება (იხ. ნახ. 51.).

ტესტ-პროგრამის შემდგომი ფრაგმენტის გრაფიკული პრედიკატები აფორმირებენ ლილაკებს, რომლებზეც თავსდება შეკითხვის საგარაუდო პასუხები (იხ. ნახ.54). ამ ფრაგმენტში გამოყენებულია პროლოგის ერთერთი ძირითადი კონსტრუქცია-წესი.

```
make_but([A, _]T, S):-
  button(_, but_func_, A, 80, S, 400, 25),
  S1 is S+40,
  make_but(T, S1).
```

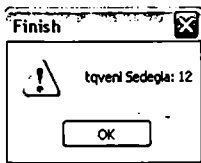
ნახ. 54.

წესის თავში მოთავსებული A ცვლადი საგარაუდო პასუხების სიის პირველი ვარიანტია. „დეფისის“ სიმბოლო, რითაც პროლოგში აღინიშნება ფიქტიური ცვლადი, ნიშნავს იმას, რომ საგარაუდო პასუხების ლილაკებზე პასუხის შემფასებელი ქულები არ გამოდის.

ქულები იწერება ინფორმაციის გამოტანის  ეკლში. T სიის ტანია, ე.ი. პასუხების დარჩენილი ვარიანტებია. S = 70 აღებულია make_but(G_Variant, 70)-დან (იხ. ნახ. 52). იგი გვიჩვენებს პასუხების პირველი ლილაკის დაცილებას შეკითხვის წარმოსახვითი ფანჯრიდან. S1 არის მანძილი შეკითხვის წარმოსახვითი ფანჯრიდან მეორე საგარაუდო პასუხის ლილაკამდე. 40 ეს არის დაცილება პასუხების ლილაკებს შორის: S1 is S + 40. პრედიკატით make_but(T, S1) ძირითად ფანჯარაში იქმნება შესაძლო პასუხების ლილაკები.

სტუდენტის მიერ საგარაუდო პასუხის მოფიქრების შემდეგ, იგი მიმოთიებულით აწყაუნებს სათანადო ლილაკზე. პროგრამის მიერ ფიქსირდება პასუხის სისწორე და

სათანადო ქულა, რომელიც გამოდის  ეკლში. ამის შემდეგ პროგრამა ავტომატურად გადადის ახალ შეკითხვაზე, იხურება „მშობელი“ ფანჯარა, ერთით იზრდება შეკითხვის რიგითი ნომერი. ახალ ძირითად ფანჯარაში ფორმირდება რიგით მომდევნო შეკითხვის შესაბამისი ყველა პარამეტრი. პროცესი გრძელდება ტესტის ყველა შეკითხვის ამოწურვამდე. პროგრამის ეს ნაწილი ეფუძნება პროლოგის ერთერთ ძირითად მქანოზმს-რეკურსიას. პროგრამულად ეს შემდეგნაირად ხდება. საგარაუდო პასუხების სიას ჩამოეჭრება პირველი საგარაუდო პასუხი, მომეწონ ვარიანტი გადაადგილდება სიის სათავეში და ა.შ. ტესტირების პროცესის დამთავრებისას ეკრანზე გამოდის ქვეფანჯარა შესაბამისი შეტყობინებით, სადაც მოცემულია ტესტირების შედეგი. იხ. ნახ. 55.



ნახ. 55.

ამოცანა 1. ტექსტ-პროგრამა საქართველოს ისტორიაში

ქვეთავში მოყვანილია ტექსტ-პროგრამა, რომელიც შედგენილია განსხვავებული ტექნოლოგიით. ტესტში არ ვიყენებთ გრაფიკულ პრედიკატებს. იგი ეფუძნება ტრადიციულ დიალოგურ რეჟიმს მომხმარებელსა და კომპიუტერს შორის. მასში გამოყენებულია „თუ-მაშინ, თუ არა და“ (IF-THEN-ELSE) კონსტრუქცია, ინფორმაციის შეტანა-გამოტანის ოპერატორები, პროგრამის შედგენის ბლოკური მეთოდი და პროლოგის სხვა მექანიზმები.

ტექსტ-პროგრამის „საქრებლემო არეა“ საქართველოს ისტორია. მისი დანიშნულებაა პიროვნების ტესტირება საქართველოს ისტორიაში. ტესტი შედგება 10 შეკითხვისგან. თვითოეულ შეკითხვაზე არსებობს 4 სავარაუდო პასუხი. სწორი პასუხის გაცემის შემთხვევაში გამოდის შეტყობინება „სწორია“. არასწორი პასუხის შემთხვევაში კი „მცდარია“. ეკრანზე ასევე ფიქსირდება მიღებული ქულების რაოდენობა. პროგრამაში ერთი შეკითხვიდან მეორეზე გადასვლა ხდება ავტომატურად. შემოტანილ გლობალურ ცვლადებში $G_A, G_B, \dots, G_J$ შეკითხვებზე სწორი პასუხის შემთხვევაში ფორმირდება „1“ (მაგ. $G_A=1$), მცდარი პასუხისას კი „0“ (მაგ. $G_C=0$).

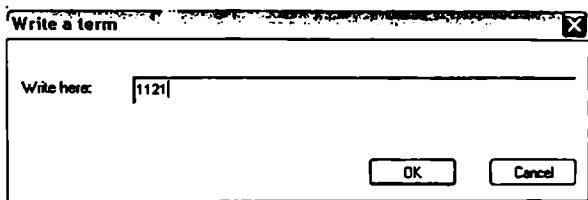
მოყვანათ ტესტირების პროცესის ფრაგმენტი. ტექსტ-პროგრამის სრული ტექსტი მოყვანილია დანართი 4-ში.

ტექსტ-პროგრამის გაშვების შემდეგ ეკრანზე, ინფორმაციის გამოტანის  Output ეულში, გამოდის პირველი შეკითხვა თავისი შესაბამისი 4 სავარაუდო პასუხით (იხ. ნახ. 5.6). აქვე გამოდის ინფორმაციის შეტანის „READ“ ფანჯარა.

```
*****KITXVA PIRVELI*****
ROMEL CELS MOXDA DIDGORIS BRDZOLA?
a) 1452  b)1121  c)1089  d)1745
```

ნახ. 5.6.

ოთხი სავარაუდო პასუხიდან ვირჩევთ ჩვენი აზრით სწორ პასუხს და შეგვაქვს ინფორმაციის შეტანის ქვეფანჯარაში (იხ. ნახ. 5.7). შემდეგ მიმთითებლით ვაწკაპუნებთ „OK“.  Output ეულში გამოდის შემდეგი შეტყობინება (იხ. ნახ. 5.8.):



ნახ. 5.7.

```

SCORIA!
*****
TQVEN MIIGET 1 QULA!

```

ნახ. 5.8.

ამის შემდეგ ტესტ-პროგრამა ავტომატურად გადადის მეორე კითხვაზე:

```

*****KITXVA MEORE*****
VIN IKO TANAR MEPIS MANA?
a) giorgi_3 b)davit2 c)erkle d)simoni

```

და ა. შ. ტესტირების პროცესის დამთავრებისას **Output** ველში გამოდის შემდეგი შეტყობინება:

```

*****
TQVEN MIIGET 1 QULA!
TQVENS MIER DAGROVILI QULATA JANIA: 8 QULA!Yes

```

ამით ტესტირების პროცესი დასრულებულია.

როგორც მეხუთე თავის დასაწყისში აღნიშნეთ, ტესტ-პროგრამები გამოიყენება ადამიანის მოღვაწეობის მრავალ სფეროში. შემდეგი ამოცანა შეეხება საყოფაცხოვრებო სფეროს, კერძოდ კი მოსკოვის ავტობაზრის ფუნქციონირებას.

ამოცანა 2. ქ. მოსკოვის ავტობაზარი „Mercedes“-ის ცენტრი.

პროლოგ-პროგრამა დაიწერა და ემსახურება მოსკოვის ავტოსახლს, კერძოდ კი „Mercedes“-ის ცენტრს. ამ პროგრამის, ისევე როგორც ზემოთ განხილული სხვა პროგრამების ც/ბ-ში, ჩაწერილია რეალური ინფორმაცია.

ავტომატურად მერსედესის პოტენციურ მყიდველს ეძლევა საშუალება მის ხელთ არსებული თანხისა და ავტომობილის პარამეტრების ანალიზის საფუძველზე შეიძინოს მისთვის შესაფერისი ავტომობილი.

პროლოგ-პროგრამაში გამოყენებულია მყიდველის ტესტირების მეთოდი. გრაფიკული ოპერატორების საშუალებით შექმნილია ძირითადი ფანჯარა, რომელსაც იყენებს მყიდველი გადაწყვეტილების მიღებისას. იხილეთ ნახ. 5.3.

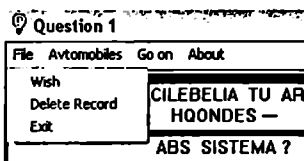
ძირითადი ფანჯრის სათაურის სტრიქონში თავსდება ავტომობილის პოტენციური მყიდველის ტესტ-კითხვის (Question) რიგითი ნომერი.

ძირითადი ფანჯრის მენიუების სტრიქონში მოთავსებულია მისი კომპონენტები, რომელთაც ჩვენ ტესტირებისას საჭიროებისამებრ გამოვიყენებთ.

ფანჯრის შავ ფონზე ჩასმულია მერსედესის ფოტოსურათები. ფანჯრის კომპონენტების შექმნის გრაფიკული პრედიკატები ჩვენს მიერ დაწერილებითაა განხილული წინა მაგალითებში. ამ საკითხებს ჩვენ აღარ შეეხებით.

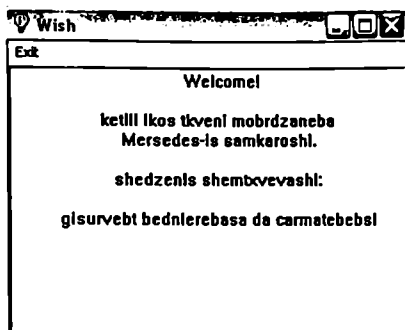
მენიუს სტრიქონი შედგება ოთხი კომპონენტისგან: „File“, „Avtomobiles“, „Go on“, „About“.

„File“ კომპონენტი შედგება ქვემენიუსგან. იხ. ნახ. 5.1.

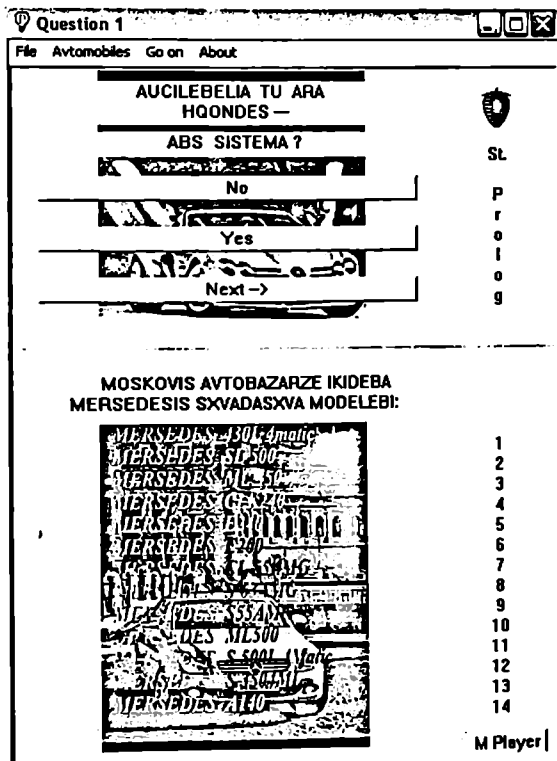


ნახ. 5.1.

„Wish“ (სურვილი) ქვემენიუს ეკრანზე გამოიყენებს პოტენციური მყიდველისადმი ავტობაზრის მიწვევა (სურვილი). იხ. ნახ. 5.2.



ნახ. 52

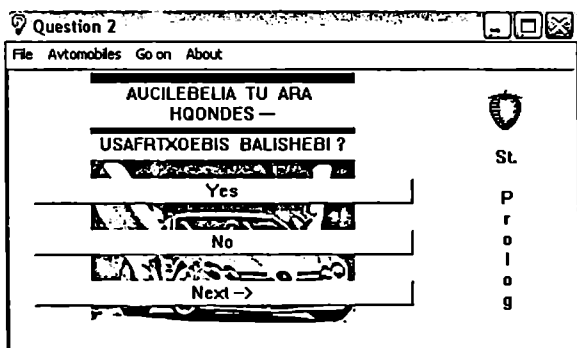


ნახ. 53. ამოცანა „Mercedes“-ის ძირითადი ტიპები

ფანჯარაში მოთავსებულია სტატიკური (უცვლელი) ტექსტი: „AUCILEBELIA TU ARA HQONDES” და პროლოგ-იმიტატორის ლოგო მარწყეი. სტატიკური ტექსტის ქვეშ ჩასმულია ტექსტი, რომელიც ავტომობილის კომპონენტებს ასახავს. პირველი შეკითხვა ასე ფორმირდება: „AUCILEBELIA TU ARA HQONDES ABS SISTEMA?”

ფანჯარაში ფორმირებულია პასუხის სამი ღილაკი: „No”, „Yes”, „Next”. მყიდველი თავისი სურვილის შესაბამისად მიმთითებლით დააწკაპუნებს რომელიმე მათგანზე. პროგრამა აფიქსირებს პასუხს და მთელელში აჯამებს შესაბამის კულებს. „No” და „Next” ღილაკზე დაჭერისას იწერება „ 0 ” კულა, ხოლო „Yes” ღილაკზე დაჭერისას „ 5 ” კულა. მთელელის ფუნქციას პროგრამაში გლობალური ცვლადი (G_Sum) ასრულებს.

რომელიმე პასუხის ღილაკზე მიმთითებლის დაწკაპუნებით გამოდის ძირითადი ფანჯარა ახალი შეკითხვით. იხ. ნახ. 5.4.



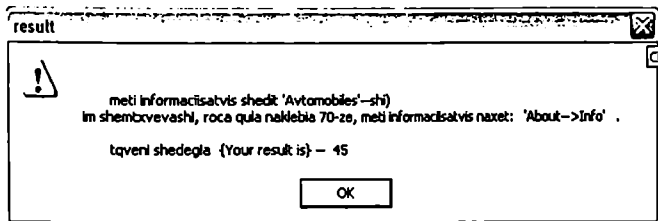
ნახ. 5.4.

შეკითხვა ასე დაფორმირდა: „AUCILEBELIA TU ARA HQONDES USAFRTXOEBIS BALISHEBI?” და ა.შ. სულ ტესტში 18 შეკითხვაა. ცოდნის ბაზა პირობითად დაყოფილია ორ ნაწილად. პირველი ნაწილი შეიცავს ტესტის შეკითხვებს. იხ. ნახ. 5.5.

ცოდნის ბაზის მეორე ნაწილში ჩაწერილია მოსკოვის ავტობაზარში გასაყიდი მერსედესის ძირითადი პარამეტრები, ავტომობილის ღირებულების ჩათვლით. პოტენციური მყიდველი პასუხს გასცემს რა ტესტის ყველა შეკითხვას პროლოგ-პროგრამას გამოაქვს ტესტირების ამსახველი ქვეფანჯარა. იხ. ნახ. 5.6.

question(1,"ABS SISTEMA ?", ["No", 0], ["Yes", 5], ["Next ->", 0]).
 question(2,"USAFRTXOEBIS BALISHEBI ?", ["Yes", 5], ["No", 0], ["Next ->", 0]).
 question(3,"SACHIS HIDROMADZLIEREBELI ?", ["No", 0], ["Yes", 5], ["Next ->", 0]).
 question(4,"SACHIS KALONKIS MAREGULIREBELI ?", ["Yes", 5], ["No", 0], ["Next ->", 0]).
 question(5,"DAMCAVI SISTEMA ?", ["No", 0], ["Yes", 5], ["Next ->", 0]).
 question(6,"IMOBILLAIZERI ?", ["Yes", 5], ["No", 0], ["Next ->", 0]).
 question(7,"CENTRALURI SAKETI ?", ["No", 0], ["Yes", 5], ["Next ->", 0]).
 question(8,"AMIND--MAREGULIREBELI ?", ["Yes", 5], ["No", 0], ["Next ->", 0]).
 question(9,"KONDINCIONERI ?", ["No", 0], ["Yes", 5], ["Next ->", 0]).
 question(10,"AUDIOSISTEMA ?", ["Yes", 5], ["No", 0], ["Next ->", 0]).
 question(11,"KRUIZ--MAREGULIREBELI ?", ["No", 0], ["Yes", 5], ["Next ->", 0]).
 question(12,"TKAVIS SALONI ?", ["Yes", 5], ["No", 0], ["Next ->", 0]).
 question(13,"SARKEEBIS ELEKTRO AMDZRAVI ?", ["No", 0], ["Yes", 5], ["Next ->", 0]).
 question(14,"SHUSHEBIS ELEKTRO AMDZRAVI ?", ["Yes", 5], ["No", 0], ["Next ->", 0]).
 question(15,"SKAMEBIS ELEKTRO AMDZRAVI ?", ["No", 0], ["Yes", 5], ["Next ->", 0]).
 question(16,"SKAMEBIS SHEMTBOBI ?", ["Yes", 5], ["No", 0], ["Next ->", 0]).
 question(17,"LUKI ?", ["No", 0], ["Yes", 5], ["Next ->", 0]).
 question(18,"MSUBUQSHENADNOBI DISKEBI ?", ["Yes", 5], ["No", 0], ["Next ->", 0]).

ნახ. 5.5.



ნახ. 5.6.

ქვეფანჯარაში აღნიშნულია, რომ ჩვენი შედეგია { Your result is } —> 45 ქულა. აქვე მოთავსებულია რჩევა იმის შესახებ, რომ მეტი ინფორმაციისათვის მიემართოთ მენიუს კომპონენტს „Avtomobiles“. მის ქვემენიუში ჩამოწერილია ქულათა რაოდენობა: 90, 85, 80, 75, 70. მეიდელის მიერ დაგროვებული ქულის შესაბამის რიცხვზე დაჭერით გამოდის „Read“ ქვეფანჯარა. მასში ქულათა რაოდენობის ჩაწერისა და დადასტურების შემდეგ კერძო გამოდის ინფორმაციის დამაზუსტებელი ქვეფანჯარა. ეოქათ, შეიდეგლმა

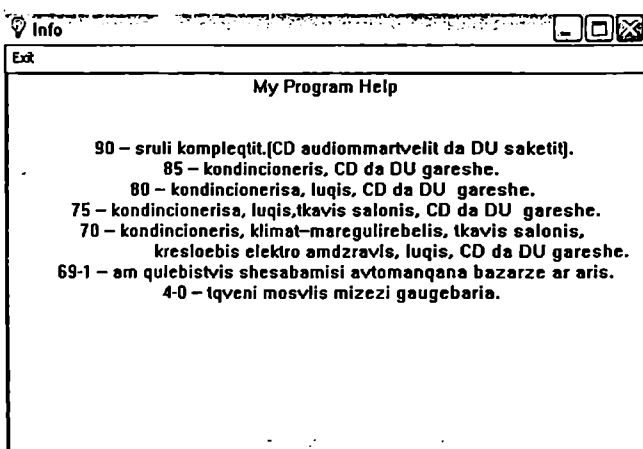
ტესტირების შედეგად დააგროვა 80 ქულა. ამ რიცხვის „Read“ ქვეფანჯარაში შეტანისა და დადასტურების შემდეგ ეკრანის **Output** ელში გამოდის იმ მერსედესების პარამეტრების ჩამონათვალი (ღირებულებების ჩათვლით), რომლებიც ჯდება 80 ქულიან ლიაპაზონში. პოტენციურ მყიდველს ეძლევა საშუალება აირჩიოს მისთვის მისაწვდომი და სასურველი ავტომობილი. 80 ქულის შესაბამისი **Output** ელში მოცემულია ნახ. 5.7.

Qula	Model	Dzaris Tipi	Karis da Adgilis Raod.	Dzravis Tipi
80	SL 500	ROADSTER	2/2	B FRISKIT
80	CLK 240 Elegance	KUPE	2/4	B FRISKIT

Cilindris Raod.	Samushao Moculoba	Sindzlavre (c.dz.)	Amdzravis	Gadacenis Kolophi
8	4966 sm/kubi	306(c.dz.)	UKANA	A
6	2597 sm/kubi	177(c.dz.)	UKANA	A

ნახ. 5.7.

ქულათა რაოდენობის მიხედვით, ავტომობილის კომპონენტების არსებობა-არარსებობის შესახებ, მყიდველს ინფორმაცია მიეწოდება „About“ მენიუს „Info“ ქვემენიუს შესაბამისი სპეციალური ქვეფანჯრით. იხ. ნახ. 5.8.



ნახ. 5.8.

ამრიგად, რელაციური ცოდნის ბაზებისა და გრაფიკული ოპერატორების დახმარებით, ჩვენ წარმოვადგინეთ რეალურ ინფორმაციაზე დაყრდნობილი მუშა ტესტ-პროგრამა. იგი ეხმარება პოტენციურ მყიდველს მიიღოს მისთვის სასურველი სწორი გადაწყვეტილება.

პროლოგ-პროგრამის ტექსტის სრული ვარიანტი მოყვანილია დანართი 4-ში.

ამრიგად, წარმოდგენილი ნაშრომის I ნაწილი შეიცავს აღნიშნული კურსის პირველ სემესტრში პრაქტიკულ და ლაბორატორიულ მეცადინეობებზე ამოსახსნელ ამოცანებს და შესაბამის პროლოგ-პროგრამებს. ლაბორატორიულ მეცადინეობებზე პროლოგ-პროგრამების ამოხსნის თანმიმდევრობას ირჩევს პედაგოგი. ამასთან, მან უნდა გაითვალისწინოს ლექციებზე წაკითხული თეორიული მასალისა და ლაბორატორიულ მეცადინეობებზე ამოსახსნელი ამოცანების შესატყვისობა.

ნაშრომის II ნაწილი, აუტორთა აზრით, სავარაუდოდ მომავალი 2007 წლისათვის გამოვა. მასში შევა კურსის „ხელოვნური ინტელექტის საფუძვლების“ II სემესტრის შესაბამისი ამოცანები და პროლოგ-პროგრამები. ამოცანების თემატიკა მრავალფეროვანი იქნება. განიხილება ზი-ის ისეთი საკითხები, როგორიცაა: ე.წ. პროდუქციული სისტემების პროექტირება, ექსპერტული სისტემების ცალკეული კომპონენტების ფორმირება, ამოცანის ამონახსნის გრაფში სხვადასხვა მეთოდებით ძებნა, გადაწყვეტილების მიღება არაცხად გარემოში, პროლოგის ე.წ. „აღრთმეტიკულ ნაწილი და სხვა.“

ავტორთა აზრით, სახელმძღვანელოს ორივე ტომი, [7] ნაშრომთან ერთად, სრულად მოიცავს აღნიშნული კურსის თეორიულ და პრაქტიკულ ასპექტებს,

ამოცანა 1. ქართული მარანი

ამოცანის პროლოგ-პროგრამის სრული ტექსტი:

%@@@@@ Q A R T U L I M A R A N I @@@@@@

%SSSSSSSSSS G E O R G I A N W I N E S SSSSSSS

%***** B E S I K I K I Z I R I A *****

%-----

%&&&&&&& C O M M E N T A R Y :&&&&&

% W I N E (c o m p o n e n t s):

- % 1. name-saxeli;
- % 2. manufactured year (pirveladi gamoshvebis celi);
- % 3. made from (damzadebulia);
- % 4. cultivate (damushavebulia-regioni);
- % 5. alcohol content (alcoholi);
- % 6. gold medals (okros medali);
- % 7. silver medals (verexlis medali);
- % 8. bronze medals (brinjaos medali);
- % 9. sugar (shakrianoba);
- % 10. difference (gansxvaveba).

%*****

% 1. difference (gvini saxeobebi):

% 2. table dry vintage wines-samarko gvino;

% 3. table dry ordinary wines-chveulebrivi(ordinaruli) gvino;

% 4. semi dry wines-naxebradmsrali gvino;

% 5. natural semi sweet wines-naturaluri naxebradtkbili gvino;

% 6. fortified wines-daspirtuli gvino;

% 7. dessert sweet wines-sadeserto gvino;

% 8. sparkling wines-shushxuna gvino;

% 9. old wines-dzveli gvino;

% 10. more new wines-shedarebit axali gvino;

% 11. new wines-axali gvino.

%*****

% @@@@ G O A L !!! @@@@

```
/*?-write(" MANUFACTURED YEAR: "),
nl,nl,write(" (pirveladi gamoshvebis celi) "),
nl,nl,wine(Name,Year,_,_,_,_),write(Name),write(" "),
write(Year),nl,nl,fail.*/
```

```
/*?-write(" M A D E F R O M: "),
nl,nl, write(" ( D A M Z A D E B U L I A) "),
nl,nl,wine(Name,_,Made_from,
_,_,_),
write(Name),write(" "),
write(Made_from),nl,nl,fail.*/
```

```
/*?-write(" C U L T I V A T I O N A R E A. "),
nl,nl,write(" (damzadebis adgili) "),
nl,nl,wine(Name,_,Cultivate,
_,_,_),write(Name),
write(" "),
write(Cultivate),nl,nl,fail.*/
```

```
/*?-write(" DAMZADEBULIA KAXETSHI. "),
nl,write("*****"),
nl,nl,wine(Name,_,kaxeti,_,
_,_),write(Name),nl,fail.*/
```

```
/*?-write("DAMZADEBULIA DASAVLET SAKARTVELOSHI. "),
nl,write("*****"),
nl,nl,wine(Name,_,west_georgia,
_,_,_),write(Name),nl,fail.*/
```

```
/*?-write(" DAMZADEBULIA IMERETSHI. "),
nl,nl,write("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"),
nl,nl,wine(Name,_,imereti,_,
_,_,_),write(Name),nl,nl,
fail.*/
```

```
/*?-write(" DAMZADEBULIA ABXAZETSHI. "),
nl,write("*****"),
nl,nl,wine(Name,_,abxazeti,_,
_,_,_),write(Name),nl,nl,fail.*/
```

```
/*?-write(" OCROS MEDALOSANI GVINOEBIA. "),
```

```

nl,write("*****"),
nl,nl,wine(Name,_,_,_,Gold,_,
_,_),Gold>=1,write(Name),write(" dajildovebulia "),
write(Gold),write(" medlit."),nl,nl,fail.*/

/*?-write("  VERCXLIS MEDALOSANI GVINOEBIA.  "),nl,
nl,write("*****"),
nl,nl,wine(Name,_,_,_,Silver,
_,_),Silver>=1,write(Name),
write(" dajildovebulia "),
write(Silver),write(" medlit."),nl,nl,fail.*/

/*?-write("  BRINJAOS MEDALOSANI GVINOEBIA.  "),
nl,nl,write("*****"),
nl,nl,wine(Name,_,_,_,Bronze,
_,_),Bronze>=1,write(Name),
write(" dajildovebulia "),
write(Bronze),write(" medlit."),nl,nl,fail.*/

/*?-write("  GVINOSHI ALCOHOLIS SHEMCVELOBA.  "),
nl,nl,write("*****"),
nl,nl,wine(Name,_,_,Alcohol,_,_,
_,_),write(Name),write("  "),
write(Alcohol),nl,nl,fail.*/

/*?-write("  GVINOSHI SHACRIS SHEMCVELOBA.  "),
nl,nl,write("*****"),
nl,nl,wine(Name,_,_,_,Sugar,_,
_,_),write(Name),write("  "),
write(Sugar),nl,nl,fail.*/

/*?-write("  SAMARKO GVINOEBIA.  "),
nl,nl,write("table dry wintage wines"),
nl,write("*****"),
nl,nl,wine(Name,_,_,_,_,_,1),
write(Name),nl,nl,fail.*/

/*?-write("  CHVEULEBRIVI(ORDINARULI) GVINOEBIA."),
nl,nl,write('    table dry ordinari wines'),
nl,write("*****"),
nl,nl,wine(Name,_,_,_,_,_,2),
write(Name),nl,nl,fail.*/

/*?-write("  NAXEVRADMSHRALI GVINOEBIA."),
nl,nl,write("    semi dry wines:"),
nl,write("*****"),
nl,nl,wine(Name,_,_,_,_,_,3),
write(Name),nl,nl,fail.*/

/*?-write("  NATURALURI NAXEVRAD-TKBILI GVINOEBIA."),
nl,nl,write("    natural semi sweet wines:"),

```

```

nl,write("*****"),
nl,nl,wine(Name,_,_,_,_,4),
write(Name),nl,nl,fail.*/

/*?-write("  DASPIRTULI GVINOEBIA: "),
nl,nl,write("  fortified wines. "),
nl,write("*****"),
nl,nl,wine(Name,_,_,_,_,5),
write(Name),nl,nl,fail.*/

/*?-write("  SADESERTO GVINOEBIA: "),
nl,nl,write("  dessert sweet wines. "),
nl,write("*****"),
nl,nl,wine(Name,_,_,_,_,6),
write(Name),nl,nl,fail.*/

/*?-write("  SHUSHXUNA GVINOEBIA: "),
nl,nl,write("  spearking wines "),
nl,write("*****"),
nl,nl,wine(Name,_,_,_,_,7),
write(Name),nl,nl,fail.*/

/*?-write("*****DZVELI GVINOEBIA*****"),
nl,nl,write("  old wines "),

nl,write("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@"),
nl,nl,old_wines(Name),write(Name),
nl,nl,fail.*/

/*?-write("*****AXALI GVINOEBIA*****"),
nl,nl,write("  new wines "),

nl,write("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@"),
nl,nl,new_wines(Name),write(Name),
nl,nl,fail.*/

/*?-write("    W I N E    "),
nl,nl,write("  KARTULI GVINOEBIA:
"),nl,write("*****"),nl,nl,wine(Wine,_,_,_,_,_),
write(Wine),nl,nl,fail.*/

/*?-write("*****SHEDAREBIT AXALI GVINOEBI*****"),

nl,nl,write("  more new wines: "),
nl,write("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@"),
nl,nl,more_new_wines(Name),write(Name),
nl,nl,fail.*/

```

%*****

%*****CODNIS BAZA*****

%*****FACTS*****

wine(tsinandali,1886,rkatsiteli_and_mcvane,
kaxeti,"10.5-12%",10,9,0,"0%",1).
wine(gurdjaani,1887,rkatsiteli_and_mcvane,
kaxeti,"10.5-12.5%",1,9,1,"0%",1).
wine(napareuli,1882,rkatsiteli,kaxeti,
"10-12%",1,6,1,"0%",1).
wine(manavi,1938,mcvane,kaxeti,
"10.5-11.5%",0,3,0,"0%",1).
wine(cicka,1890,cicka,west_georgia,
"10.5-12.5%",0,0,0,"0%",1).
wine(colikauri,1890,grape_variety,
imereti,"10.5-12.5%",0,1,0,
"0%",1).
wine(rkaciteli,1892,rkaciteli,
kaxeti,"11.5-13%",1,1,2,"0%",
1).
wine(tibaani,1948,rkaciteli,kaxeti,
"11.5-13%",2,5,0,"0%",1).
wine(sviri,1890,colikauri_and_cicka_and_kraxuna,
west_georgia,"11-12.5%",
1,0,0,"0%",1).
wine(teliani,1897,kaberne,kaxeti,
"10.5-12%",4,6,0,"0%",1).
wine(napareuli,1890,saperavi,kaxeti,
"11-12.5%",6,2,0,"0%",1).
wine(mukuzani,1888,saperavi,kaxeti,
"10.5-12.5%",8,4,1,"0%",1).
wine(ereti,1977,rkaciteli_mcvane,
unknowns,"10-12.5%",0,0,0,"0%",
2).
wine(garedji,1977,rkaciteli_mcvane,
kaxeti,"10-12.5%",0,0,0,"0%",
2).
wine(gelati,1977,colikauri_kraxuna,
west_georgia,"10-12.5%",
0,1,1,"0%",2).
wine(kaxeti,1948,rkaciteli_and_mcvane,
kaxeti,"10.5-13%",0,1,1,"0%",
2).
wine(dimi,1977,colikauri_and_kraxuna,
imereti,"10.5-13%",0,0,0,"0%",
2).
wine(shuamta,1984,rkaciteli_and_mcvane,
kaxeti,"10-12.5%",0,0,0,"0%",
2).

wine(saperavi,1886,saperavi,kaxeti,
"10.5-12.5%",1,1,0,"0%",2).

wine(anakopia,1978,colikauri,abxazeti,
"9-11%",0,0,0,"1-2%",3).

wine(aguna,1948,saperavi_kaberne_rkaciteli,
east_georgia,"9-11.5%",0,0,0,
"1-2%",3).

wine(sachino,1984,aleksandrouli_aladasturi_ojaleshi_cicka_colikauri,
west_georgia,"9-11.5%",0,0,0,
"1-2%",3).

wine(pirosmani,1981,saperavi,kaxeti,
"10-12.5%",0,0,0,"1.5-2.5%",
3).

wine(axmeta,1958,mcvane,kaxeti,
"9.5-11.5%",1,2,0,"3-5%",4).

wine(tvishi,1952,colikauri,west_georgia,
"10-11.5%",1,2,0,"3-5%",4).

wine(tetra,1945,rachuli_tetra,west_georgia,
"9-10%",0,2,0,"3-5%",4).

wine(chxaveri,1934,unknown,west_georgia,
"9.5-11%",1,4,1,"3-5%",4).

wine(psou,2000,colikauri,abxazeti,
"9-10.5%",0,2,0,"3-5%",4).

wine(alaznis_veli,1977,
rkaciteli_tetra_colikauri,
kaxeti,"9-11%",0,0,0,"0%",
4).

wine(xvanchkara,1907,aleksandrouli_mudjuretuli,
west_georgia,"10.5-12%",
2,4,0,"3-5%",4).

wine(kindzmarauli,1942,saperavi,
kaxeti,"10.5-12%",3,4,1,"3-5%",4).

wine(axasheni,1958,saperavi,kaxeti,
"10.5-12%",6,5,0,"3-5%",4).

wine(ojaleshi,1933,grape_variety,
west_georgia,"10-12%",2,1,0,
"3-5%",4).

wine(usaxelaury,1943,usaxelaury,
west_georgia,"10.5-12%",
2,3,0,"3-5%",4).

wine(kardanaxi,1926,rkaciteli,
kaxeti,"18%",8,1,0,"10%",5).

wine(anaga,1925,rkaciteli_xixvi_mcvane,
kaxeti,"19%",0,1,0,"4%",5).

wine(signagi,2000,rkaciteli,kaxeti,
"19%",0,0,0,"3%",5).

wine(iveria,1977,rkaciteli_mcvane_chinuri,
kaxeti,"18%",1,0,0,"7%",5).

wine(taribana,2000,rkaciteli,kaxeti,
"19%",0,0,0,"4%",5).

wine(marabda,1985,rkaciteli,kaxeti,
"19%",0,0,0,"5%",5).

wine(kolxeti,1977,colikauri_cicka,
west_georgia,"18%",0,1,0,"7%",
5).
wine(xixvi,1924,xixvi,kaxeti,"19%",
4,0,0,"13%",6).
wine(saamo,1923,rkaciteli,kaxeti,"17%",
4,1,0,"13%",6).
wine(salxino,1928,izabella_dzvelshava_colikauri,
west_georgia,"15%",6,0,0,"30%",6).
wine(atenuri,2000,guruli_mcvane_chinuri,
kartli,"10%",0,0,0,"0%",7).
wine(aisi,1977,saperavi_alexandrouli_odjaleshi_muguretuli,
eastern_georgia,"10.5-12.5%",
0,0,0,"0%",7).
wine(sadarbazo,1977,saperavi_alexandrouli_odjaleshi,
eastern_georgia,"10.5-12.5%",
0,0,0,"0%",7).
wine(terdzola,2000,ochxanuri_alexandrouli_mudzuretuli_cicka,
west_georgia,"10.5-12.5%",0,0,0,"3-5%",
7).
wine(achandara,2000,izabela_colikauri,west_georgia,
"11-13%",0,0,0,"0%",7).
wine(guruli_shushxuna,2000,izabela,
west_georgia,"9-11%",0,0,0,
"0%",7).

%***** R U L E S: *****

old_wines(Name):-
wine(Name,Year,_,_,_,_,_),
Year<1900.

new_wines(Name):-
wine(Name,Year,_,_,_,_,_),
Year>=1980.

more_new_wines(Name):-
wine(Name,Year,_,_,_,_,_),
Year<1980,Year>1900.

%*****

ამოცანა 2. ფეხბურთში ჩემპიონთა ლიგა: 1999-2000 წ

მოიყენეთ პროლოგ-პროგრამის სრული ტექსტი:

%*****
%***** UEFA *****
%*****


```

write(Tasi),write(" tasi. "),write(Year),write(" cels."),nl,nl,fail.*/

%?-club(dynamo,_,_,_,_,5,_,_,_).

%?-club(real,_,_,_,_,_,_,_,_,one_group).

%?-club(A,_,_,_,_,_,_,_,_,B),write(A),write(" OFF FROM "),write(B),nl,nl,fail.

/*?-write("      OLD CLUB
"),nl,nl,write("*****"),nl,nl,
club(Name,_,Year,_,_,_,_,_,Year<=1900,write(Name),write(" "),write(Year),
nl,nl,fail.*/

/*?-write("      YOUNG CLUB
"),nl,nl,write("*****"),nl,nl,
club(Name,_,Year,_,_,_,_,_,Year>=1911,
write(Name),write(" "),write(Year),nl,nl,fail.*/

/*?-write("      MORE OLD CLUB  "),nl,
write("*****"),
nl,nl,club(Name,_,Year,_,_,_,_,_,Year>1900,Year<1911,write(Name),write(" "),
write(Year),nl,nl,fail.*/

/*?-write("      MORE GET MONEY CLUB
"),nl,nl,write("*****"),nl,nl,
club(Name,_,_,_,_,_,Money,_,Money>=20,
write(Name),write(" "),write(Money),write(" mln."),nl,nl,fail.*/

/*?-write("      LEAST GET MONEY CLUB
"),nl,nl,write("*****"),nl,nl,
club(Name,_,_,_,_,_,Money,_,Money>=3.5,
write(Name),write(" "),write(Money),write(" mln."),nl,nl,fail.*/

/*?-write("      LEGENDARY CLUB
"),nl,nl,write("*****"),nl,nl,
club(Name,_,Win,_,_,_,_,_,Win>=4,
write(Name),write(" "),write(Win),write(" CUP WINNER "),nl,nl,fail.*/

/*?-write("      KICK CLUB
"),nl,nl,write("*****"),nl,nl,
club(Name,_,_,_,_,Kick,_,_,_,Kick>=31,
write(Name),write(" kick "),write(Kick),write(" goal"),nl,nl,fail.*/

```

```

/*?-write("      LET CLUB
"),nl,nl,write("*****"),nl,nl,
club(Name,_,_,_,_,Let,_,_,_,_),Let>=21,
write(Name),write(" let "),write(Let),write(" goal "),nl,nl,fail.*

%#####

%*****CODNIS BAZA*****

%*****F A C T S*****

club(arsenal,england,1901,0,"0",100,9,9,2,2,4.66,one_group).

club(aik,sweden,1914,0,"0",25,4,16,0,1,5,2.33,one_group).

club(barcelona,spain,1899,1,"1991-92",120,41,21,11,3,2,17,semy_final).

club(bayern,germany,1900,3,"1973-74@74-75@75-76",120,23,18,8,5,3,24.65,seniy_final).

club(bordeaux,france,1911,0,"0",45,12,18,3,5,4,8.65,two_group).

club(boavista,portugal,1910,0,"0",35,4,10,1,2,3,3.66,one_group).

club(valencia,spain,1903,0,"0",85,27,18,8,4,5,29.32,final).

club(villem2,holland,1906,0,"0",10,7,15,0,2,4,2.66,one_group).

club(galatasaray,turkey,1905,0,"0",60,10,13,2,1,3,3.33,one_group).

club(herta,germany,1901,0,"0",35,10,18,2,4,6,7.32,two_group).

club(dortmund,germany,1901,1,"1996-97",70,7,9,1,3,2,4,one_group).

club(dynamo,ukraine,1905,0,"0",60,18,16,5,2,5,9.66,two_group).

club(croatia,croatia,1909,0,"0",20,7,7,1,2,3,3.66,one_group).

club(chelsea,england,1900,0,"0",95,22,14,7,3,4,16,quarter_final).

club(lazio,italy,1900,0,"0",120,26,12,8,4,2,17.32,quarter_final).

club(leverkusen,germany,1911,0,"0",70,7,7,1,4,1,4.32,one_group).

club(manchester,england,1899,2,"1967-68@98-99",450,22,12,8,3,3,17,quarter_final).

club(milan,italy,1899,5,"1962-63@68-69@88-89@89-90@93-94",250,6,7,1,3,2,4,one_group).

club(molde,norway,1999,0,"0",10,5,13,1,0,5,3,one_group).

club(maribor,slovenia,1912,0,"0",10,2,12,1,1,4,3.33,one_group).

```

club(olimpiakos,greece,1889,0,"0",55,9,11,2,1,3,4.33,one_group).

club(olympik,france,1902,1,"1992-93",60,13,19,4,2,6,8.66,two_group).

club(porto,portugal,1903,1,"1986-87",50,19,18,7,2,5,15.66,quarter_final).

club(psv,holland,1903,1,"1987-88",60,5,10,1,1,4,3.33,one_group).

club(rangers,scotland,1873,0,"0",85,7,7,2,1,3,4.33,one_group).

club(real,spain,1890,8,"1955-56@56-57@57-58@58-59@59-60@65-66@97-98@99-2000",320,33,24,10,3,4,36,winner).

club(rosenborg,norway,1901,0,"0",50,17,16,3,3,6,8,two_group).

club(spartak,russia,1903,0,"0",35,9,12,1,2,3,3.66,one_group).

club(sparta,czech,1912,0,"0",20,19,18,4,5,3,9.65,two_group).

club(sturm,austria,1918,0,"0",20,5,12,2,0,4,4,one_group).

club(fiorentina,italy,1901,0,"0",70,15,15,4,5,3,9.65,two_group).

club(feenoord,holland,1904,1,"1969-70",65,14,13,3,7,2,9.31,two_group).

%*****END*****

ამოცანა 3. ავტომანქანების კატალოგი

მოვიყენოთ ამოცანის პროლოგ-პროგრამის სრული ტექსტი:

```
%****Programma Katalog Mashin****
%***** G O A L *****
%*****KOMENTARII*****

% 1. Marka mashini;
% 2. Model mashini;
% 3. Tip kuzava(s-sedan,c-cupe,ca-cupecabroileti,u-universali,x-xechbeki,m-miniveni,ma-
mikroavtobusi);
% s-sedan:4 kariani(umino sabargulit);
% c-cupe:2
% 4. Tip dvigatelya(b-benzin,d-disel);
% 5. Cillindri(4,5,6,8,10,12);
% 6. L/S(Loshadinnie silli);
% 7. Cena.
```

%***** G O A L !!! *****

```
/*?-write("Marka Mashin:"),nl,  
    mashina(Marka,_,_,_,_),write(Marka),nl,fail.*/  
  
/*?-write("Model Mashin:"),nl,write("*****"),nl,nl,  
    mashina(Marka,Model,_,_,_),write(Marka),  
    write("    "),write(Model),nl,fail.*/  
  
/*?-write("Tip Dvigatelya-benzin:"),nl,nl,  
    mashina(Marka,Model,_,Z,_,_),write(Marka),write("    "),  
    write(Model),write("    "),write(Z),nl,fail.*/  
  
/*?-write("Tip Dvigatelya-dizel:"),nl,nl,  
    mashina(Marka,Model,_,d,_,_),write(Marka),write("    "),  
    write(Model),write("    "),nl,fail.*/  
  
/*?-write("Tip Kuzova-cupe:"),nl,nl,  
    mashina(Marka,Model,c,_,_,_),write(Marka),write("    "),  
    write(Model),write("    "),nl,nl,fail.*/  
  
/*?-write("Tip Kuzova-sedan:"),nl,nl,  
    mashina(Marka,Model,s,_,_,_),write(Marka),write("    "),  
    write(Model),write("    "),nl,nl,fail.*/  
  
/*?-write("Tip Kuzova-universal:"),nl,nl,  
    mashina(Marka,Model,u,_,_,_),write(Marka),write("    "),  
    write(Model),write("    "),nl,nl,fail.*/  
  
/*?-write("Tip Kuzova-cupe_cabrioleti:"),nl,nl,  
    mashina(Marka,Model,ca,_,_,_),write(Marka),write("    "),  
    write(Model),write("    "),nl,nl,fail.*/  
  
/*?-write("Cillindri-4:"),nl,nl,  
    mashina(Marka,Model,_,_,4,_,_),write(Marka),write("    "),  
    write(Model),write("    "),nl,nl,fail.*/  
  
/*?-write("Cillindri-5:"),nl,nl,  
    mashina(Marka,Model,_,_,5,_,_),write(Marka),write("    "),  
    write(Model),write("    "),nl,nl,fail.*/  
    ,  
  
/*?-write("Cillindri-6:"),nl,nl,  
    mashina(Marka,Model,_,_,6,_,_),write(Marka),write("    "),  
    write(Model),write("    "),nl,nl,fail.*/  
  
/*?-write("Cillindri-8:"),nl,nl,  
    mashina(Marka,Model,_,_,8,_,_),write(Marka),write("    "),  
    write(Model),write("    "),nl,nl,fail.*/  
  
/*?-write("Cillindri-10:"),nl,nl,  
    mashina(Z,C,_,_,10,_,_),write(Z),write("    "),  
    write(C),write("    "),nl,nl,fail.*/>
```

```
/*?-write("Cillindri-12:"),nl,nl,
    mashina(Marka,Model,_,_,12,_,_),write(Marka),write(" "),
    write(Model),write(" "),nl,nl,fail.*/
```

```
/*?-write("Mashini_1_classa:"),nl,nl,
    mashini_1_classa(Marka,Model),write(Marka),write(" "),
    write(Model),write(" "),point,nl,nl,fail.*/
```

```
/*?-write("Mashini_2_classa:"),nl,nl,
    mashini_2_classa(Marka,Model),write(Marka),write(" "),
    write(Model),write(" "),point,nl,nl,fail.*/
```

```
/*?-write("Mashini_3_classa:"),nl,nl,
    mashini_3_classa(Marka,Model),write(Marka),write(" "),
    write(Model),write(" "),point,nl,nl,fail.*/
```

```
/*?-write("Moshnie mashini:"),nl,nl,
    mashini_moshnie(Marka,Model),write(Marka),write(" "),
    write(Model),write(" "),point,nl,nl,fail.*/
```

```
/*?-write("Slabie mashini:"),nl,nl,
    mashini_slabie(Marka,Model),write(Marka),write(" "),
    write(Model),write(" "),point,nl,nl,fail.*/
```

%***** B/ D *****

%***** FAKTI *****

```
mashina(alfa_romeo,"156",s,b,4,120,24900).
mashina(alfa_romeo,"gtv 2.0 ts",c,b,4,150,39900).
mashina(audi,"a3 1.9 tdi",x,d,4,110,29431).
mashina(audi,"a4 avant 2.4",u,b,6,165,42393).
mashina(audi,"a6 2.5 tdi quattro",s,d,6,150,56240).
mashina(audi,"a8 4.2 quattro",s,b,8,300,115411).
mashina(audi,"s8 4.2 quattro",s,b,8,340,111965).
mashina(audi,"cabrio 2.8",ca,b,6,174,53360).
mashina(audi,"tt 1.8 turbo",c,b,4,225,57500).
mashina(bmw,"e36-318 tds compact",x,d,4,90,30260).
mashina(bmw,"e39-540i",s,d,6,143,46130).
mashina(bmw,"m3",c,b,6,321,68540).
mashina(bmw,"328i cabrio",ca,b,6,193,52540).
mashina(bmw,"z3 m s50",ca,b,6,321,70350).
mashina(bmw,"540 a touring",u,b,8,286,80480).
mashina(bmw,"e38-750 il",s,b,12,326,139410).
mashina(bmw,"e31-840 ci",c,b,8,286,102270).
mashina(volvo,"v40",u,b,4,140,31000).
mashina(volvo,"940",s,b,4,155,36000).
mashina(volvo,"s90",s,b,4,204,48000).
mashina(volvo,"s70",s,b,5,240,53000).
mashina(volvo,"v70 awd",u,b,5,193,53500).
mashina(volvo,"c70",c,b,5,240,64800).
```

mashina(jeep,"cherokee 2.5 td",u,d,4,116,37900).
 mashina(jeep,"cherokee 4.0 a",u,b,6,177,47000).
 mashina(jeep,"grand cherokee-limited 5.9",u,b,8,250,73500).
 mashina(jeep,"grand cherokee-laredo 2.5 td",u,d,4,116,51200).
 mashina(jeep,"wrangler se 4.0",u,b,6,184,38500).
 mashina(dodge,"durango 5.9",u,b,8,250,58000).
 mashina(dodge,"grand caravan 3.3",m,b,6,158,46490).
 mashina(dodge,"interpid 3.2",s,b,6,223,49990).
 mashina(cadillac,"seville sts",s,b,8,305,8846).
 mashina(chrysler,"300m",s,b,6,253,48000).
 mashina(chrysler,"grand voyager le 3.8 a",u,b,6,166,56000).
 mashina(chrysler,"grand goyager se 2.5 td",u,d,4,115,46700).
 mashina(chrysler,"viper",c,b,10,455,120000).
 mashina(lexus,"gs 300",s,b,6,212,65000).
 mashina(land_rover,"discovery 2.5tdi",u,d,4,113,45500).
 mashina(land_rover,"range rover 4.6",u,b,8,225,91000).
 mashina(lincoln,"town car",s,b,8,210,64700).
 mashina(lincoln,"navigator",u,b,8,232,68500).
 mashina(mersedes_benz,"a160",x,b,4,102,35300).
 mashina(mersedes_benz,"e430",s,b,8,279,96000).
 mashina(mersedes_benz,"s600(w-140)",s,b,12,394,169000).
 mashina(mersedes_benz,"g320",u,b,6,210,100800).
 mashina(mersedes_benz,"sprinter",ma,d,5,122,52500).
 mashina(mersedes_benz,"v230",m,b,4,143,58000).
 mashina(mersedes_benz,"slk 230 compressor",c_ca,b,4,193,39100).
 mashina(mitsubishi,"eclipse gs",c,b,4,145,25000).
 mashina(mitsubishi,"l 400 glx",ma,b,4,128,40000).
 mashina(mitsubishi,"galant gls",s,b,4,164,32150).
 mashina(mitsubishi,"pajero-wagon gli 2.8td",u,d,4,125,42220).
 mashina(mitsubishi,"pajero-wagon gls 3.5",u,b,6,208,61540).
 mashina(nissan,"maxima 3.0 se a",s,b,6,193,38900).
 mashina(nissan,"primera gx",x,b,4,90,20400).
 mashina(nissan,"patrol-tdse",u,d,6,130,46500).
 mashina(opel,"vectra-cd",s,b,4,136,28200).
 mashina(opel,"omega mv6",s,b,6,210,49800).
 mashina(opel,"frontera 2.5td",u,d,4,115,37300).
 mashina(peugeot,"605 stdt",s,d,4,109,32170).
 mashina(peugeot,"406 coupe",c,b,4,194,50610).
 mashina(peugeot,"306 xnd 1.9",x,d,4,68,17290).
 mashina(peugeot,"605 sv 3.0",s,b,6,167,43650).
 mashina(porsche,"boxter",ca,b,6,204,70290).
 mashina(porsche,"911 carrera",c,b,6,285,119304).
 mashina(porsche,"911 carrera cabrio",ca,b,6,285,133360).
 mashina(saab,"9-3 se 2.0t cabrio",ca,b,4,185,49990).
 mashina(saab,"9-5 2.3 lpt",s,b,4,170,46585).
 mashina(saab,"9-3 SE 2.0t",c,b,4,185,40050).
 mashina(chevrolet,"tahoe 5.7",u,b,8,255,60000).
 mashina(chevrolet,"blazer 4.3",u,b,8,200,43000).
 mashina(chevrolet,"cavalier(1j69)",s,b,4,120,28000).
 mashina(chevrolet,"suburban 7.4",u,b,8,290,65000).
 mashina(jaguar,"sovereign 4.0 lwb",s,b,8,295,117500).
 mashina(jaguar,"executive 3.2",s,b,8,245,91500).

```
mashina(jaguar,"xk-8 coupe",c,b,8,295,120000).
mashina(jaguar,"xkr",c,b,8,375,130000).
```

```
%***** P R A V I L A *****
```

```
mashini_1_classa(Marka,Model):-
mashina(Marka,Model,_,_,_,Cena), 80000<Cena,Cena=<170000.
```

```
mashini_2_classa(Marka,Model):-
mashina(Marka,Model,_,_,_,Cena), 30000<Cena,Cena=<80000.
```

```
mashini_3_classa(Marka,Model):-
mashina(Marka,Model,_,_,_,Cena), 10000<Cena,Cena=<30000.
```

```
mashini_moshnie(Marka,Model):-
mashina(Marka,Model,_,_,X,_), 180<X,X=<455.
```

```
mashini_slabie(Marka,Model):-
mashina(Marka,Model,_,_,X,_), 50<X,X=<180.
```

```
.....
```

ამოცანა 4. მსოფლიო კურორტები.

ამოცანის სრული ტექსტი:

```
%***** OAZISI *****
```

```
%*****KOMENTARI:*****
```

```
%1.kurortis adgilmdebareoba;
%2.sastumro(kotegebi);
%3.sastumros saxelcodeba;
%4.varskvlavebis raodenoba;
%5.momsaxureba;
%6.erti dgis girebuleba(lari);
%7.qveknebi,
```

```
%***** G O A L !!!*****
```

```
/*?-write("KURORTIS ADGILMDEBAREOBA"),nl,
write("*****"),nl,nl,
kurorti(Adgili,_,_,_,_),
write(Adgili),nl,fail.*/*
```

```
/*?-write( " SASTUMROEBI " ),nl,
write("*****"),nl,nl,
```

```

kurorti(Adgili,_,Dasaxeleva,_,_,_),
write(Adgili),write("  "),
write(Dasaxeleva),nl,fail.*/

/*?-write( " MOMSAXUREBA " ),nl,
write("*****"),nl,nl,
kurorti(,_,Name,_,Momsaxureba,Pasi,_,),
write(Name),write("  "),
write(Momsaxureba),write("  "),
write(Pasi),write(" lari."),nl,fail.*/

/*?-write( " QVEKANA " ),nl,
write("*****"),nl,nl,
kurorti(Adgili,_,_,_,_,Qvekana),
write(Adgili),write("  "),
write(Qvekana),nl,fail.*/

/*?-write("SASTUMROS PASI"),nl,
write("*****"),nl,nl,
kurorti(,_,,"marko_polo",_,_,Pasi,_,),
write(marko_polo),
write("  "),write(Pasi),
write(" lari."),nl,nl,*/

/*?-write("SAQARTVELOS KURORTEBI"),nl,
write("*****"),nl,nl,
kurorti(Adgili,_,_,_,_,saqartvelo),
write("saqartvelo"),write("  "),
write(Adgili),nl,nl,fail.*/

/*?-write("BATUMIS SASTUMROEBI"),nl,
write("*****"),nl,nl,
kurorti(batumi,_,Dasaxeleva,_,_,_),
write("  "),
write(Dasaxeleva),nl,nl,fail.*/

/*?-write("SASTUMROS PASI"),nl,
write("*****"),nl,nl,
kurorti(,_,_,Dasaxeleva,_,_,Pasi,_,),
write(Dasaxeleva),
write("  "),write(Pasi),
write(" lari."),nl,fail.*/

/*?-write("SASTUMROS DONE"), nl,
write("*****"), nl,nl,
kurorti(,_,_,Dasaxeleva,Star,
_,_,_),write(Dasaxeleva),
write("  "),write(Star),write(" ***"),
nl,fail.*/

```

```
/*?-write("MAGALI MOMSAXUREBIS SASTUMRO"),nl,
write("*****"),
nl,nl,
magali_momsaxurebis_sastumro(S),fail.*/
```

```
/*?-write("DABALI MOMSAXUREBIS SASTUMRO"),
nl,write("*****"),
nl,nl,
dabali_momsaxurebis_sastumro(D),fail.*/
```

%*****CODNIS BAZA*****

%*****F A C T E B I *****

```
kurorti(gudauro,sastumro,"marko_polo",
4,sam_dgeradi_kveba,
80,saqartvelo).
kurorti(bakuriani,kotegi,"mshenebeli",
0,or_dgeradi_kveba,
30,saqartvelo).
kurorti(bakuriani,kotegi,"geologebi",
0,or_dgeradi_kveba,
20,saqartvelo).
kurorti(bakuriani,kotegi,"xrustali",
0,or_dgeradi_kveba,
18,saqartvelo).
kurorti(bordgomi,kotegi,"piruzi",
0,vor_dgeradi_kveba,
25,saqartvelo).
kurorti(batumi,sastumro,"L.bakuri",
5,sam_dgeradi_kveba,
70,saqartvelo).
kurorti(batumi,sastumro,"tanamgzavri",
4,sam_dgeradi_kveba,
60,saqartvelo).
kurorti(batumi,sastumro,"inturisti",
0,vert_dgeradi_kveba,
15,saqartvelo).
kurorti(shovi,sastumro,"samotxe",
0,sam_dgeradi_kveba,
20,saqartvelo).
kurorti(shovi,kotegi,"edemi",
0,or_dgeradi_kveba,
15,saqartvelo).
kurorti(maltakva,kotegi,"maltakva",
0,or_dgeradi_kveba,
20,saqartvelo).
kurorti(kvareli,kotegi,"saundge",
0,sam_dgeradi_kveba,
```

```

10.saqartvelo).
kurorti(isxvarithamia,kotegi,"kosmosi",
0.or_dgeradi_kveba,
10.saqartvelo).
kurorti(trapisoni,sastumro,"besti",
3.sam_dgeradi_kveba,
40.turketi).
kurorti(opatia,sastumro,"vila de almasadori",
4.otx_dgeradi_kveba,70.xorvatia).
kurorti(opatia,sastumro,"hotel and vila",
3.sam_dgeradi_kveba,
60.xorvatia).

```

%*****CESEBI*****

```

magali_momsaxurebis_sastumro(A):-
kurorti(Qalaci,_Name,Star,_Pasi,_),
Star>=3,Pasi>=40,write(Qalaci),
write(" "),write(Name),write(Star),
write(" ").write(" "),write(Pasi),
write(" lari."),nl,nl.

```

```

dabali_momsaxurebis_sastumro(D):-
kurorti(Qalaci,_Name,Star,_Pasi,_),
Star<3,Pasi<60,write(Qalaci),
write(" "),write(Name),write(Star),
write(" ").write(" "),write(Pasi),
write(" lari."),nl,nl.

```

%*****

ამოცანა 5. მსოფლიოს ქვეყნები

ამოცანის პროლოგ-პროგრამის სრული ტექსტი:

%***** MSOFLIOS QVEKNEBI: *****

%*****KOMENTARI:*****

% 1. gveknis saxeli;

% 2. kontinenti;

% 3. saxelmcpis mockobis porma;

% 4. dedacalaci;

% 5. gveknis mosaxleobis raodenoba(ml.);


```

%?-write(" SAMXRET AMERICIS CVEKNEBIS MOSAXLEOBIS
RAODENOBAA:").nl,write("*****"),nl,nl,country(Name,samxret_america_._,Pop_._),write(Name),write("  "),write(Pop),write(" ml.
"),nl,nl,fail.

%?-write(" CRDILOET AMERICIS CVEKNEBIS MOSAXLEOBIS
RAODENOBAA:").nl,write("*****"),nl,nl,country(Name,crdiloet_america_._,Pop_._),write(Name),write("  "),write(Pop),write(" ml.
"),nl,nl,fail.

%?-country(france,._,Cap,._,._),write("FRANCE"),write("  "),write(Cap),nl,nl.

%?-country(france,europa,._,._,._).

%?-country(france,W,._,._,._),write(W),nl,nl.

%?-country(italy,._,._,._,Ter,._),write("Ter="),write(Ter),write(" 1000 kv.km."),nl,nl.

%?-write(" EUROPA -TERRITORY:
").nl,write("*****"),nl,nl,country(Name,europa,._,._,Ter,._),write(Name),write("  "),write(Ter),write(" 1000 kv.km."),nl,nl,fail.

%?-write(" ASIA-TERRITORY:
").nl,write("*****"),nl,nl,country(Name,asia,._,._,Ter,._),write(Name),write("  "),write(Ter),write(" 1000 kv.km."),nl,nl,fail.

%?-write(" SAMXRET AMERICA-TERRITORY:
").nl,write("*****"),nl,nl,country(Name,samxret_america_._,._,Ter,._),write(Name),write("  "),write(Ter),write(" 1000 kv.km."),nl,nl,fail.

%?-write(" CRDILOET AMERICA-TERRITORY:
").nl,write("*****"),nl,nl,country(Name,crdiloet_america_._,._,Ter,._),write(Name),write("  "),write(Ter),write(" 1000 kv.km."),nl,nl,fail.

%?-country(italy,._,._,._,U),write(U),write(" cels."),nl,nl.

%?-write("EVROPIS GVEKNEBSHI MOSAXLEOBIS AKCERIS
CELI.").nl,write("*****"),nl,nl,country(Name,europa,._,._,._,Celi),write(Name),write("  "),write(Celi),nl,nl,fail.

%?-write("SAMXRET AMERICIS KVEKNEBIS MOSAXLEOBIS AKCERIS
CELI.").nl,write("*****"),nl,nl,country(Name,samxret_america_._,._,._,Celi),write(Name),write("  "),write(Celi),nl,nl,fail.

%?-write("CRDILOET AMERICIS KVEKNEBIS MOSAXLEOBIS AKCERIS
CELI.").nl,write("*****"),nl,nl,country(Name,crdiloet_america_._,._,._,Celi),write(Name),write("  "),write(Celi),nl,nl,fail.

%?-country(dania,._,._,Cap,._,._),write(Cap),nl,nl.

%?-country(china,europa,._,._,._,._).

```


country(malta.europa,"republika",valleta,0.363,0.316,1993).
country(monaco.europa,"satavado",monaco,0.0305,0.00195,1993).
country(norvegia.europa,"samepo",oslo,4.3,387.0,1993).
country(shveicaria.europa,"confederacia",berni,6.99,41.3,1993).
country(shvecia.europa,"samepo",stogholmi,8.7,450.0,1993).
country(sakartvelo.europa,"republika",tbilisi,5.493,69.7,1993).

%***** A S I A *****

country(azerbajjani.asia,"republika",baco,7.4,86.6,1993).
country(somxeti.asia,"republika",erevani,3.412,29.8,1993).
country(vietnami.asia,"soc.republika",xanoi,70.9,332.0,1993).
country(izraeli.asia,"republika",ierusalimi,5.5,20.8,1993).
country(indoeiti.asia,"republika",deli,897.0,3300.0,1993).
country(irani.asia,"islamuri republika",teirani,61.0,1650.0,1993).
country(chineti.asia,"saxalxo republika",pekini,1179.0,9600.0,1993).
country(kuveiti.asia,"republika",el_kuveiti,1.4,17.8,1993).
country(nepali.asia,"samepo",katmandu,19.3,147.2,1993).
country(singapuri.asia,"republika",singapuri,2.88,0.639,1993).
country(siria.asia,"republika",damasko,13.4,185.2,1993).
country(tailandi.asia,"samepo",bankoki,57.8,514.0,1993).
country(turketi.asia,"republika",ankara,59.9,780.6,1993).
country(iaponia.asia,"*****",tokio,127.0,372.0,1993).

%***** S A M X R E T A M E R I C A *****

country(kolumbia.samxret_america,"republika",bogota,41.9,1139.0,1995).
country(venesuela.samxret_america,"republika",karakasi,24.1,912.0,1995).
country(ekvadori.samxret_america,"republika",kito,12.6,284.0,1995).
country(peru.samxret_america,"republika",lima,25.6,1285.0,1995).
country(brazilia.samxret_america,"republika",brazilia,165.9,8512.0,1995).
country(bolivia.samxret_america,"republika",lapasi,8.3,1099.0,1995).
country(paragvai.samxret_america,"republika",asensioni,5.4,407.0,1995).
country(cile.samxret_america,"republika",santiago,15.2,757.0,1995).
country(argentina.samxret_america,"republika",buenos_airesi,36.1,2777.0,1995).
country(urugvai.samxret_america,"republika",montevideo,3.0,178.0,1995).

%***** C R D I L O E T A M E R I C A *****

country(usa.crdiloet_america,"republika",vashingtoni,273.7,9363.0,1996).
country(kanada.crdiloet_america,"republika",otava,30.5,9976.0,1996).
country(meksika.crdiloet_america,"republika",meksiko,98.7,1973.0,1996).
country(gvatemala.crdiloet_america,"republika",gvatemala,9.8,109.0,1996).
country(salvadori.crdiloet_america,"republika",san_salvadori,6.2,21.0,1996).
country(hondurasi.crdiloet_america,"republika",tegusigalpa,6.4,112.0,1996).
country(nikaragua.crdiloet_america,"republika",managua,4.9,130.0,1996).
country(kostarika.crdiloet_america,"republika",san_xose,3.6,51.0,1996).

country(panama,crdiloet_america,"republika",panama,2.8,77.0,1996).

%*****

%@@@@@ RULES @@@

big_country(A) :-
country(A,_,_,M,D,_), M >= 30.0, D >= 200.0.

small_country(C) :-
country(C,_,_,S,F,_),S <= 10.0, F <= 100.0.

%*****

პროგრამა 1. ფირმაში ვაკანტური თანამდებობების შერჩევა

ა) ამოცანა: სამეცნიერო ფირმა, ამოცანის სრული ტექსტი:

%*****V A K A N S I A*****

%***MMARTVELI GADACKVETILEBIS MIGEBIS AMOCANA****

%*****A M O C A N I S A G C E R A*****

% VTQVAT ROMELIGAC SAMECNIERO-TEQNIKUR PIRMA-

% SHI ARIS TANAMDEBOBRIVI VAKANSIEBI.

% ES VAKANSIEBIA:

% 1.MECNIERI-MUSHAKI;

% 2.INGINER-KONSTRUQTORI;

% 3.EQSPLUOTACHI-INGINERI;

% SAMUSHAOS MADZIEBLEBI MOCEMUL KONKRETUL SHEM-

% TXVEVASHI (AM AMOCANASHI) FASDEBIAN SHEMDEGI

% PARAMETREBIT:

% 1. SAMECNIERO XARISXI (DEGREE);

% 2. SAMECNIERO AGMOCHENEBI (DISCOVERY);

% 3. DIPLOMIS SASHUALO NISHANI (GRADE);

% 4. ERTI SPECIALOBIT MUSHAOBIS UCKVETI STAGI

% (EXPERIENCE-gamocdileba).

%-----

% AMOCANIS AMOXSNIS I ETAPI.

%-----

% SAMUSHAQS MADZIEBLIS MONACEMEBIS SHETANA.

%-----

?-monacemi.

monacemi:-

write(" MOGVACODET TKVENI MONACEMEBI:"),

nl,nl,

read (Monacemi),write(Monacemi),nl,nl,

monacemi.

%%-----

%% AMOCANIS AMOXSNIS II ETAPI.

%%-----

%% SAMUSHAOS MADZIEBLIS PARAMETREBIS FORMIREBA

%% DA MAT BAZAZE GADACKVETILEBIS MIGEBA.

%%-----

%%?-qualify.

qualify:-

write(" GAQVT SAMECNIERO XARISXI?"),nl,nl,

read(Degree),write("DEGREE="),write(Degree),
nl,nl,

%%-----

Degree=no->Qualify=no,

%%-----

write("PIRMASHI MIGEBAZE UARI ETQVAS!"),nl,

write("*****"),nl,nl,

write("Qualify="),write(Qualify),point,nl,nl

else Qualify=yes,

%%-----

write("SHESADZLEBELIA MIVIGOT PIRMASHI."),nl,

write("*****"),
nl,nl,

write("Qualify="),write(Qualify),point,nl,nl.

%%@@

%%---TANAMDEBOBEBIS(VAKANSIEBIS) SHERCHEVA.-----

--position1.

```
%-----  
% Position-Tanamdeboba.  
% Posicion1:mecnieri-mushaki.  
  
%-----  
position1:-  
  
write(" GAQVT SAMECNIERO XARISXI?"),nl,nl,  
read(Degree),write("DEGREE="),write(Degree),  
nl,nl,write(" GAQVT MECNIERULI AGMOCHENEBI?"),  
nl,nl,read(Discovery),write("DISCOVERY="),  
write(Discovery),nl,nl,  
  
%-----  
Degree=yes,Discovery=yes->  
Position=mecnieri_mushaki,  
  
%-----  
write(" MIVIGOT PIRMASHI MECNIER-MUSHAKAD."),nl,  
write("*****"),  
nl,nl,write(" POSITION="),write(Position),nl,  
write("*****"),nl,nl  
else Position=no,  
  
%-----  
,  
write(" PIRMASHI MIGEBAZE UARI ETQVAS!"),nl,  
write("*****"),  
nl,nl,write(" Position="),write(Position),point,  
nl,write("*****"),nl,nl.  
  
% a a a a a a a u a a a a u a a a a a a a a a a a a a a a a a a a a a a
```

%%?-position2.

%%-----

% Position2:inginer-konstruqtori.

Position2:-

write("GAQVT SHESABAMISI KVALIPIKACIA?"),

nl,nl,read(Qualify),write("QUALIFY="),

write(Qualify),point,nl,nl,

write("DIPLOMIS SASHUALO NISHANI?"),nl,nl,

read(Grade),write("GRADE="),write(Grade),

point,nl,nl,

%%-----

Qualify=yes,Grade>=3.5->

Position=inginer_konstruqtori,

%%-----

write("MIVIGOT PIRMASHI INGINER-KONSTRUQTORAD."),

nl,write("*****"),

nl,nl,write("Position="),write(Position),nl,

write("*****"),nl,nl

else Position=no,

%%-----

write("PIRMASHI MIGEBAZE UARI ETQVAS!"),nl,

write("*****"),

nl,nl,write(" Position="),write(Position),nl,

write("*****"),nl,nl.

%%@@

%-position3.

%-----

% Position3:ekspluotaciis ingineri.

%-----

position3:-

write("GAQVT SHESABAMISI KVALIPIKACIA?"),

nl,nl,read(Qualify),write("Qualify="),

write(Qualify),point,nl,nl,

write("DIPLOMIS SASHUALO NISHANI?"),nl,nl,

read(Grade),write("Grade="),write(Grade),

point,nl,nl,

write("MOCEMULI SPECIALOBIT MUSHAOBIS STAGI?"),

nl,nl,read(Experience),write("Experience="),

write(Experience),point,nl,nl,

%-----

Qualify=yes,Grade<3.5,Experience>=2->

Position=ekspluotaciis_ingineri,

%-----

write("MIVIGOT PIRMASHI EQSPLUOTACIIS INGINRAD."),

nl,write("*****"),

nl,nl,write("Position="),write(Position),point,

nl,write("*****"),

nl,nl else Position=no,

%-----

write("PIRMASHI MIGEBAZE UARI ETQVAS!"),nl,

write("*****"),nl,

nl,write("Position="),write(Position),nl,

write("*****"),nl,nl.

%*****PROGRAMIS DASASRULI.*****

ბ) ამოცანა: სარეკლამო სააგენტო. ამოცანის სრული ტექსტი:

%***** VAKANSIA *****

%***** AMOCANIS AGCERA *****

% VTQVAT ROMELIGAC FIRMASHI (SAREKLAMO SAAGENTOSHI)

% ARIS TANAMDEBOBRIVI VAKANSIEBI.

% ES VAKANSIEBIA :

% 1. GRAFICIS DIZAINERI;

% 2. DAMKABADONEBELI;

% 3. INGLISURI ENIS SPECIALISTI;

% 4. SAOFICE PROGRAMEBIS MASCAVLEBELI.

% SAMUSHAOS MADZIEBLEBI MOCEMUL KOHKRETUL

% SHEMTXVEVASHI PASDEBIAN SHEMDEGI PARAMETREBIT:

% 1. GRAFICULI PROGRAMEBIS CODNA - COREL DRAW, PHS (KNOW);

% 2. PAGEMAKERIS AN QUARK EXPRESIS CODNA (KNOW);

% 3. DIPLOMIS AN CERTIFICATIS ARSEBOBA (CERTIFICATE);

% 4. DIPLOMIS SASHUALO NISHANI (GRADE)

% 5. SPECIALOBIT MUSHAOBIS STAGI (EXPERIENCE).

% -----

% AMOCANIS AMOXSNIS I ETAPI.

% SAMUSHAOS MADZIEBLIS MONACEMEBIS SHETANA.

% -----

%?-monacemi.

monacemi:-

write("MOGVACODET TKVENI MONACEMEBI: "),nl,nl,

read(Monacemi),write(Monacemi),nl,nl,monacemi.

```
% -----
% AMOCANIS AMOXSNIS II ETAPL.
% SAMUSHAOS MADZIEBLIS PARAMETREBIS PORMIREBA
% DA MAT BAZAZE GADACKVETILEBIS MIGEBA.
% -----
```

```
%?- tanamdeboba1.
```

```
tanamdeboba1:-
write("RA GRAFICULI PROGRAMEBI ICIT?"),nl,nl,
read(Know),write("I KNOW : "),write(Know),
nl,nl,
Know=coreldraw_photoshop->Knowledge=enough,
```

```
% -----
write("SESADZLEBELIA MIVIGOT PIRMASHI "),nl,
write("GRAFICULI DIZAINERIS TANAMDEBOBAZE. "),nl,nl,
write("The reason is :"),nl,
write("Knowledge is ="),write(Knowledge),nl,nl
```

```
else Knowledge=not_enough,
```

```
%
write("PIRMASHI MIGEBAZE UARI ETQVAS!"),nl,
write("*****"),nl,nl,
write("Knowledge is ="),write(Knowledge),nl,nl.
```

```
% *****
```

```
%?-tanamdeboba2.
```

```
    % damkabaonebeli
```

```
tanamdeboba2:-
write("Ploht PageMakers da/an QuarkExpress-s?"),nl,nl,
read(Know), write("ANSWER IS = "),write(Know),
nl,nl,write("GAQVT CERTIFICATI?"),
nl,nl,read(Certificate),write("ANSWER IS ="),
write(Certificate),nl,nl,
```

```
%
Know=yes, Certificate=yes->
Position=damkabadonebeli,
```

```
%
```

```
write("MIVIGOT PIRMASHI DAMKABADONEBLIS TANAMDEBOBAZE."),nl,
write("*****"),
nl,nl,write("POSITION="),write(Position),nl,
write("*****"),nl,nl
```

```
else Position=no,
```

```
%
write("PIRMASHI MIGEBAZE UARI ETQVAS!"),nl,
write("*****"),
nl,nl,write("Position="),write(Position),
nl,write("*****"),nl,nl.
```

```
% *****
```

```
%-tanamdeboba3.
```

```
% inglisuri enis spec.
```

```
tanamdeboba3 :-
```

```
write("GAQVT SHESABAMISI CODNA?"),
nl,nl,read(Know),write("ANSWER IS ="),
write(Know),nl,nl,
write(" DIPLOMIS SASHUALO NISHANI?"),nl,nl,
read(Grade),write("GRADE="),write(Grade),
nl,nl,
```

```
Know=yes, Grade>=4->
```

```
Position=specialist_of_English_Language,
```

```
% write("MIVIGOT PIRMASHI INGLISURI ENIS SPECIALISTAD."),
nl,write("*****"),
nl,nl,write("Position="),write(Position),nl,
write("*****"),nl,nl
```

```
else Position=no,
```

```
write("PIRMASHI MIGEBAZE UARI ETQVAS!"),nl,
write("*****"),
nl,nl,write("Position="),write(Position),nl,
write("*****"),nl,nl.
% *****
```

```
%-tanamdeboba4.
```

```
% saoffice programebis mascavlebeli.
```

```
tanamdeboba4 :-
```

```
write("GAQVT SURVILI ASCAVLOT SXVEBS?"),
nl,nl,read(Wish),write("Answer is ="),
write(Wish),nl,nl,
```


% a) Tvitoeul kmakofaze mkof pirze investors angarishze aucilebelia edos
 % 5000 Lari mainc;
 % b) Sakmaris, stabilur shemosavlal chavtvalot celicadshi 15 000 Lari;
 % g) Amastan, plius 4000 Lari tvitoeul kmakofaze mkof pirze.

% Am pirobebis gatvaliscinebit „FINANSURI MRCHEVELI" sheimushavebs
 % konkretuli investorisatvis, konkretul safinanso strategias !!!

% MIZNEBI

?-investicia(tanxa_bankshi).
 %?-investicia(akcia).
 %?-investicia(combinacia).

% DZIRITADI CESEBI

% CESI 1

investicia(tanxa_bankshi):-

write(" GTXOVT, GVACNOBOT INVESTORIS VINAoba !!!"),

read(Investori),nl,nl, write(" I N V E S T O R I A "),
 write(Investori),nl, nl,
 write("Ramdeni piria tqvens kmakofaze?"),read(G_Y),nl,
 write("-----"),nl,
 write("Tqvens kmakofazea: "),write(G_Y),write(" piri"),nl,nl,
 %-----

tanxa_bankshi(arasakmarisi)->

%-----
 write("*****"),nl,
 write("Tqven tanxa unda sheinaxot bankshi"),nl

else write("*****"),nl,
 write("Tqven unda ganaxorcielot sxva saxis investicia"),nl,fail.

% CESI 2

investicia(akcia):-

write(" GTXOVT, GVACNOBOT INVESTORIS VINAoba !!!"),
 read(Investori),nl,nl, write(" I N V E S T O R I A "),
 write(Investori),nl, nl,

write("Ramdeni piria tqvens kmayofaze?"),read(G_Y),nl,

```

write("-----"),nl,
write("Tqvens kmayofazea: "),write(G_Y),write(" piri"),nl,nl,
%-----

tanxa_bankshi(sakmarisi),shemosavali(sakmarisi)->
%-----

write("*****"),nl,
write("Tqven shegidzliat tanxis dabandeba akciebshi"),nl

else write("*****"),nl,
write("Tqven unda ganaxorcielot sxva saxis investicia"),nl,fail.

%+++++++ CESI 3 ++++++++

investicia(combinacia):-

write(" GTXOVT, GVACNOBOT INVESTORIS VINAoba !!!"),
read(Investori),nl,nl, write(" I N V E S T O R I A "),
write(Investori),nl, nl,

write("Ramdeni piria tqvens kmayofaze?"),read(G_Y),nl,
write("-----"),nl,
write("Tqvens kmayofazea: "),write(G_Y),write(" piri"),nl,nl,
%-----

tanxa_bankshi(sakmarisi),shemosavali(arasakmarisi)->
%-----

write("*****"),nl,
write("Tqven unda ganaxorcielot kombinaciuri dabandeba"),nl

else write("*****"),nl,
write("Tqven unda ganaxorcielot sxva saxis investicia"),nl,fail.

%||||||||| DAmXMARE CESEBI |||||||||||11111111

%+++++++CESI 1+++++++

tanxa_bankshi(sakmarisi):-
write("Ra tanxa gagachniat bankshi? "),read(X),nl,
write("-----"),nl,
write("Tqvens angarishzea:"),write(X),write(" lari"),nl,nl,
%-----

X>=G_Y*5000.
% X- bankis angarishze arsebuli tanxa

```

```

% G_Y      kmakopaze myofta raodemoba
%-----

```

```

%+++++++CESI 2+++++++

```

```

tanxa_bankshi(arasakmarisi):-
  write("Ra tanxa gagachniat bankshi? "),read(X),nl,
  write("-----"),nl,
  write("Tqvens angarishzea:"),write(X),write(" lari"),nl,nl,
%-----

```

```

      X<G_Y*5000.
% X-  bankis angarishze arsebuli tanxa
% G_Y      kmakopaze myofta raodemoba
%-----

```

```

%+++++++CESI 3+++++++

```

```

shemosavali(sakmarisi):-
  write("Tqveni shemosavali stabiluria? "),read(Y),nl,
  write("-----"),nl,
%-----

```

```

      Y=yes,
%-----

```

```

  write("Tqveni shemosavali stabiluria"),nl,nl,
  write("Tqveni mimdinare shemosavali? "),read(X),nl,
  write("-----"),nl,
  write("Tqveni mimdinare shemosavalia:"),write(X),write(" lari"),nl,nl,
%-----

```

```

      X>=15000+ (G_Y*4000).
% X-  mimdinare shemosavali
% G_Y      kmakopaze myofta raodemoba
%-----

```

```

%+++++++CESI 4+++++++

```

```

shemosavali(arasakmarisi):-
  write("Tqveni shemosavali stabiluria? "),read(Y),nl,
  write("-----"),nl,
  write(Y),nl,nl,
%-----

```

```

Y=no;
%-----
write("Tqveni mimdinare shemosavali ? "),read(X),nl,
write("-----"),nl,
write("Tqveni mimdinare shemosavalia:"),write(X),write(" lari"),nl,nl,
%-----

X<15000+ (G_Y*4000).
% X- mimdinare shemosavali
% G_Y kmakopaze myofa raodemoba

%-----

```

ამოცანა 3. საქარტველოს რეკინიზა

ამოცანის სრული ტექსტი:

```

?-
window(,_, win_func(, "Matareblebis Ganrigi", 200, 100, 300, 300). % mtavari panjris
-----

win_func(init) :-
    menu(normal,_,_, menu_exit(, "Gamosvla", ), % panjarashi menuebis
sheqmna
    menu(pop_up,_,_, menu_go(, "Biletis Archeva", ),
    menu(normal,_,_, menu_about(, "Programis Shesaxe"),
    edit(G_E1,_, edit_func(, "", 160, 10, 100, 25), % text-box-ebis sheqmna
    edit(G_E2,_, edit_func(, "", 160, 50, 100, 25),
    edit(G_E3,_, edit_func(, "", 160, 100, 100, 25),
    edit(G_E4,_, edit_func(, "", 160, 150, 100, 25),
    static(,_, fa(, "Danishnulebis Adgili:", 10, 10,140, 20), % Label-ebis
sheqmna
    static(,_, fa(, "Matareblis Tipi:", 10, 50,140, 20),
    static(,_, fa(, "Adgilis Katgoria:", 10, 100,140, 20),
    static(,_, fa(, "Fasi:", 10, 150,140, 20),
    button(,_,bfunc(, "Damaxsovreba",80,210,120,25). % Button-is sheqmna

win_func(close) :- not( yes_no("Darcmunebuli Xar ?", "Gtxovt Daadasturot Damosvla!", !)).
% X-ze dacherisas eg gamoaq

menu_exit(press) :- close_window(, % panjris meniushi "Gamosvla"-ze dacherisas
shesasrulebeli operacia

```

```
menu_about(press):-message("Programis Shesaxe!", "Es Programa daceres: Khatia
Khatiashvilma, Tamuna Arabidzem da Nodar Saldadzem !",n).
% "Programis Shesaxe"-veli :)
```

```
menu_go(init):-
```

```
    menu(normal, _,s1(),"Danishnulebis Adgili"),      % biletis pirobebi
chamonatvali
    menu(normal, _,s5(),"Danishnulebis adgili da Pasit"),
    menu(normal, _,s7(),"Danishnulebis Adgili, Matareblis Tipit da Pasit"),
    menu(normal, _,s8(),"Yvela Pirobit").
```

```
%=====pirveli piroba=====
```

```
s1(press):-
```

```
    read(Destination, "Sheivyvanet Danishnulebis Adgili"),nl,
    write("++++++"),nl,
    write("+      PASUXI      +"),nl,
    write("++++++"),nl,nl,
bileti(A,B,C,D),
Destination=A,
write("Danishnulebis Adgili: "),write(A),nl,
write("  Matareblis Tipi: "),write(B),nl,
write("  Adgilis Kategorია: "),write(C),nl,
write("      Sapasuri: "),write(D),nl,
write("-----"),nl,fail.
```

```
%=====mcere piroba=====
```

```
s2(press):-
```

```
    read(Price, "Sheivyvanet Tanxa"),nl,
    write("++++++"),nl,
    write("+      PASUXI      +"),nl,
    write("++++++"),nl,nl,
bileti(A,B,C,D),
Price>=D,
write("Danishnulebis Adgili: "),write(A),nl,
write("  Matareblis Tipi: "),write(B),nl,
write("  Adgilis Kategorია: "),write(C),nl,
write("      Sapasuri: "),write(D),nl,
write("-----"),nl,fail.
```

```
%=====mesame piroba=====
```

s3(press):-

```
read(Seat, "Sheiyvanet Adgilis Kategorია"),nl,
write("++++++"),nl,
write("+ PASUXI +"),nl,
write("++++++"),nl,nl,
```

bileti(A,B,C,D),

Seat=C,

write("Danishnulebis Adgili: "),write(A),nl,

write(" Matarebilis Tipi: "),write(B),nl,

write(" Adgilis Kategorია: "),write(C),nl,

write(" Sapasuri: "),write(D),nl,

write("-----"),nl,fail.

%=====meotxe piroba=====

s4(press):-

```
read(Train, "Sheiyvanet Matarebilis Tipi"),nl,
write("++++++"),nl,
write("+ PASUXI +"),nl,
write("++++++"),nl,nl,
```

bileti(A,B,C,D),

Train=B,

write("Danishnulebis Adgili: "),write(A),nl,

write(" Matarebilis Tipi: "),write(B),nl,

write(" Adgilis Kategorია: "),write(C),nl,

write(" Sapasuri: "),write(D),nl,

write("-----"),nl,fail.

%=====mexute piroba=====

s5(press):-

```
read(Destination, "Sheiyvanet Danishnulebis Adgili"),
read(Price, "Sheiyvanet Tanxa"),nl,
write("++++++"),nl,
write("+ PASUXI +"),nl,
write("++++++"),nl,nl,
```

bileti(A,B,C,D),

Destination=A.Price>=D,

write("Danishnulebis Adgili: "),write(A),nl,

write(" Matarebilis Tipi: "),write(B),nl,

write(" Adgilis Kategorია: "),write(C),nl,

write(" Sapasuri: "),write(D),nl,

write("-----"),nl,fail.

%=====meeqvsc piroba=====

s6(press):-

```
    read(Seat, "Sheiyvanet Adgilis Kategorია"),
    read(Price, "Sheiyvanet Tanxa"),nl,
    write("+++++"),nl,
    write("+ PASUXI +"),nl,
    write("+++++"),nl,nl,
bileti(A,B,C,D),
Seat=C,Price>=D,
write("Danishnulebis Adgili: "),write(A),nl,
write(" Matareblis Tipi: "),write(B),nl,
write(" Adgilis Kategorია: "),write(C),nl,
write(" Sapasuri: "),write(D),nl,
write("-----"),nl,fail.
```

%=====meshvide piroba=====

s7(press):-

```
    read(Destination, "Sheiyvanet Danishnulebis Adgili"),
    read(Train, "Sheiyvanet Matareblis Tipi"),
    read(Price, "Sheiyvanet Tanxa"),nl,
    write("+++++"),nl,
    write("+ PASUXI +"),nl,
    write("+++++"),nl,nl,
bileti(A,B,C,D),
Destination=A,Train=B,Price>=D,
write("Danishnulebis Adgili: "),write(A),nl,
write(" Matareblis Tipi: "),write(B),nl,
write(" Adgilis Kategorია: "),write(C),nl,
write(" Sapasuri: "),write(D),nl,
write("-----"),nl,fail.
```

%=====mervce' piroba=====

s8(press):-

```
    read(Destination, "Sheiyvanet Danishnulebis Adgili"),
    read(Train, "Sheiyvanet Matareblis Tipi"),
    read(Seat, "Sheiyvanet Adgilis Kategorია"),
    read(Price, "Sheiyvanet Tanxa"),nl,
    write("+++++"),nl,
    write("+ PASUXI +"),nl,
    write("+++++"),nl,nl,
bileti(A,B,C,D),
Destination=A,Train=B,Seat=C,Price>=D,
```

```

write("Danishnulebis Adgili: "),write(A),nl,
write(" Matarebilis Tipi: "),write(B),nl,
write(" Adgilis Kategorია: "),write(C),nl,
write(" Sapasuri: "),write(D),nl,
write("-----"),nl,fail.

```

%=====Codnis Bazis Shevseba=====

bfunc(press):-

```

    A is get_text(G_E1),
    B is get_text(G_E2),
    C is get_text(G_E3),
    D is get_text(G_E4),
    assert_in(bileti(A,B,C,D)),
message("Madlobt", "Monacemebi Shetania!"!),

```

```

    set_text(" ",G_E1),
    set_text(" ",G_E2),
    set_text(" ",G_E3),
    set_text(" ",G_E4).

```

%*****F A K T E B I*****

```

bileti(batumi,dizeli,saerto,12.5).
bileti(batumi,sapirmo,saerto,3.7).
bileti(batumi,sapirmo,plackarti,8).
bileti(batumi,sapirmo,kupe,13).
bileti(batumi,sapirmo,rbili,20).
bileti(batumi,samgzavro,saerto,3.4).
bileti(batumi,samgzavro,plackarti,5.1).
bileti(batumi,samgzavro,kupe,10.8).
bileti(batumi,samgzavro,rbili,15).
bileti(poti,chqari,plackarti,5.8).
bileti(poti,chqari,kupe,13.3).
bileti(poti,samgzavro,saerto,3.3).
bileti(poti,samgzavro,plackarti,4.9).
bileti(poti,samgzavro,kupe,10.4).
bileti(poti,samgzavro,rbili,14.3).
bileti(zugdidi,chqari,plackarti,5.8).
bileti(zugdidi,chqari,kupi,11.3).
bileti(zugdidi,samgzavro,saerto,3.3).
bileti(zugdidi,samgzavro,plackarti,4.9).
bileti(zugdidi,samgzavro,kupe,10.4).
bileti(zugdidi,samgzavro,rbili,14.3).
bileti(ozurgeti,samgzavro,saerto,3.3).
bileti(ozurgeti,samgzavro,plackarti,4.9).
bileti(ozurgeti,samgzavro,kupe,10.5).
bileti(ozurgeti,samgzavro,rbili,14.6).
.....

```

ამოცანა 4. კ. თბილისის სამარშრუტო ტაქსები

ამოცანის სრული ტექსტი:

```
%**** PROGRAMMA-Marshuti.*****

%  programas gamoaqvs kvela marshutkis nomeri

%  romelic migikvant mocemuli pirveli

%  punqtidan meoreshi.

%-----B E S I K I K I Z I R I A-----

%***** G O A L !!! *****

?-write("*****"),nl,fail.

?-A1="tavisuflebis moedani", A2="vaja pshavelas gamz.", write(A1),write("-"),write(A2),nl,write("dadis marshutkebi:"),nl,marshuti(A1,A2,A),write("N:"),write(A),nl,fail.

?-write("====="),nl,fail.

/*?-A1="qavtarazis q.", A2="tavisuflebis moedani", write(A1),write("-"),write(A2),nl,write("gadis marshutkebi "),nl,marshuti(A1,A2,A),write("N:"),write(A),nl,fail.*/

%?-write("====="),nl,fail.

/*?-A1="digomi", A2="vake", write(A1),write("-"),write(A2),nl,write("gadis marshrutkebi "),nl,marshuti(A1,A2,A),write("N:"),write(A),nl,fail.*/

%?-write("====="),nl,fail.

/*?-A="m. isani",B="qavtarazis q.", write(A),write("-"),write(B),nl,write("gadis marshrutkebi "),nl,marshuti(A,B,N),write("N:"),write(N),nl,fail.*/

%*****F A K T E B I*****
```

marshutka(2,"tavisuflebis moedani","vaja pshavelas gamz. ").
marshutka(2,"qavtarazis q.", "universitetis niaglini korpusi").

marshutka("2a","m. isani","qetevan zamebulis gamz. ").
marshutka("2a","tavisuflebis moedani","vaja pshavelas gamz. ").
marshutka("2a","qavtarazis q.", "").

marshutka(22,"tavisuflebis moedani","gmirta moedani").
marshutka(22,"gamsaxurdias gamz.", "mizkevichis q. ").
marshutka(22,"sairmis q.", "").

marshutka(34,"II bavshvta saavadmkofo","digomi").
marshutka(34,"samxedro gza","gagarinis moedani").
marshutka(34,"gamsaxurdias gamz.", "tavisuflebis moedani").
marshutka(34,"qetevan zamebulis gamz.", "m. isani").
marshutka(34,"m. samgori","gmirta moedani").
marshutka(34,"k/studia","eliavas bazroba").
marshutka(34,"zeretlis gamziri","digomi").

marshutka(44,"digomi","gagarinis moedani").
marshutka(44,"vaja pshavelas gamz.", "m. gozirize").
marshutka(44,"vake","republikis moedani").

marshutka(85,"gldanis 2-4 m/r","m. sarajishvili").
marshutka(85,"guramishvilis gamz.", "zeretlis gamz. ").
marshutka(85,"aglazis q.", "didubis q. ").
marshutka(85,"tamar melfis gamz.", "kostavas q. ").
marshutka(85,"tavisuflebis moedani","baratashvilis q. ").

marshutka(108,"digmis 2 kvartali","mirexulavas q. ").
marshutka(108,"gvetazis q.", "dinamo stadioni").
marshutka(108,"agmasheneblis gamz.", "zaarbrukenis moedani").
marshutka(108,"marcxena sanapiro","gulias moedani").
marshutka(108,"centraluri avtosadguri","zeretlis gamz. ").
marshutka(108,"gvetazis q.", "").

marshutka(126,"m. didube","z. dadianis q. ").
marshutka(126,"konstituziis q.", "z. chavchavazis q. ").
marshutka(126,"agmasheneblis gamz.", "m. javaxishvilis q. ").
marshutka(126,"meliqishvilis q.", "chavchavazis gamz. ").
marshutka(126,"yifshizis q.", "").

marshutka(128,"gldanis 4-8 m/r","guramishvilis gamz. ").
marshutka(128,"m. didube","robakizis gamz. ").
marshutka(128,"gagarinis moedani","vaja pshavelas gamz. ").
marshutka(128,"m. gozirize","vakis parki").
marshutka(128,"berzenishvilis q.", "gamsaxurdias zegli").

marshutka("128a","gldani","guramishvilis gamz.").
 marshutka("128a","robakizis gamz.","gagarinis moedani").
 marshutka("128a","gamsaxurdias gamz.","varazisxevi").
 marshutka("128a","barnovis q.","faliashvilis q.").
 marshutka("128a","gamsaxurdias zegli","").

%*****C E S E B I*****

marshuti(A,B,N):-
 marshutka(N,A,_),
 marshutka(N,B,_).

marshuti(A,B,N):-
 marshutka(N_,A),
 marshutka(N_,B).

marshuti(A,B,N):-
 marshutka(N,A,_),
 marshutka(N_,B).

marshuti(A,B,N):-
 marshutka(N_,A),
 marshutka(N,B,_).

%*****

ამოცანა 5. კომპიუტერის კონფიგურაციისა და ღირებულებების შერჩევა მყიდველის მოთხოვნათა შესაბამისად

მოვიყენოთ ამოცანის სრული ტექსტი:

%***** C O M P U T E R *****

%***** CPU-procesori.*****

%***** H.D.-vinchesteri.*****

%***** M.B.-deda plata.*****

%***** Monitori.*****

%***** G O A L !!! *****

%-computer(510,F),fail.

?-computer(550,T),fail.

```
/*?-write("  C P U.  "),nl,  
write("*****"),  
nl,nl,cpu(_,Price),write(Price),  
write(" S"), nl,nl,fail.*
```

```
/* ?-write("  H.d.  "),nl,  
write("*****"),  
nl,nl,h_d(Name,_),write(Name),nl,nl,fail.*
```

```
/*?-write("  M_B  "),nl,  
write("*****"),  
nl,nl,m_b(Name,_),write(Name),nl,nl,fail.*
```

```
/*?-write("  MONITOR  "),nl,  
write("*****"),  
nl,nl.monitor(Name,_),write(Name),nl,nl,fail.*
```

,

%***** CODNIS BAZA *****

%***** FAKTEBI:*****

%-----C P U-----PRACE(S)-----

cpu("pentium233 MMX", 111).

cpu("pentiumII 400 MMX", 712).

```
cpu("pentiumII 350 MMX", 523).
```

```
cpu("pentiumII 266 MMX", 201).
```

```
cpu("pentium 233 MMX box", 112).
```

```
cpu("pentiumII 333 MMX ", 413).
```

```
%-----H.D.-----PRICE($)-----
```

```
h_d("fujitsu 2.1 Gb", 111).
```

```
h_d("fujitsu 3.2 Gb", 120).
```

```
h_d("fujitsu 5.2 Gb", 168).
```

```
h_d("quantum 3.2 Gb", 143).
```

```
h_d("fujitsu 8.4 Gb", 265).
```

```
h_d("seagate 4.3 GB", 133).
```

```
%-----M.B.-----PRICE($)-----
```

```
m_b("aristo am", 60).
```

```
m_b("asus p2b", 166).
```

```
m_b("asus p2197", 124).
```

```
m_b("mycomp a15", 90).
```

```
%-----MONITOR-----PRICE($).
```

```
monitor("panasonic 15 " , 220).
```

```
monitor("panasonic 17 " , 445).
```

```
monitor("panasonic 17pf " ,660).
```

```
monitor("view sonic 17 " , 720).
```

```
%*****CESI*****
```

```
computer(X,Y):-
```

```
cpu(A,B),
```

```
h_d(C,D),
```

```
m_b(G,H),
```

```
monitor(I,J),
```

```
X > B + D + H + J, Y = B + D + H + J,
```

```
write(" COMPUTERIS PASI="),
```

```
write(Y),write(" S"),nl,
```

write("++++++"),nl,nl,

write("C P U"),write(" "),write(A),

write(" price="),write(B),write(" S"),nl,nl,

write("H.D."),write(" "),write(C),

write(" price="),write(D),write(" S"),nl,nl,

write("M.B."),write(" "),write(G),

write(" price="),write(H),write(" S"),nl,nl,

write("MONITOR"),write(" "),write(I),

write(" price="),write(J),write(" S"),nl,

write("*****"),
nl,nl,nl.

.....

ამოცანა 6. სტრუქტურული ვახშავი

ამოცანის სრული ტექსტი:

%*****KARTULI SUPRA*****

%SSSSSSSSSSSS-COMENTARI-SSSSSSSSSSSSSSSS

% drink-alkoholuri sasmeli-----

% cooldrink-gamagrilebeli sasmeli-----

% hotfood-cxeli kerdzi-----

% cooldfood-civi kerdzi-----

%*****

%-----G O A L !!!-----

?-write(" D R I N K S:"),drinks(Q,_),
nl,nl,write(Q),point,nl,fail.

/*?-write(" C O O L D R I N K S:"),
cooldrinks(W,_),nl,nl,write(W),point,fail.*

/*?-write(" H O T F O O D:"),hotfood(E,_),
nl,nl,write(E),point,fail.*

/*?-write(" C O O L F O O D:"),coolfood(R,_),
nl,nl,write(R),point,fail.*

%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

%?-supra(4,18,Z),fail.

%?-supra(5,25,Z).

% 25 lari-is dros gvaqvs 1800-se meti biji.

%?-supra(5,50,Z).

% xolo vtqvats,50 laris dros,gvaqvs 33 biji.

%-----
% RAC UPRO DID TANXAS MIVCEM "FAIL"-is GARE-
% SHE MIZANSHI MIT UPRO MCIRDEBA BIJEBIS(STEP)
% RAODENOB "PROOF TREE"-shi!!!

% YES-is dros AUCILEBLAD VNAXOT "PROOF TREE!"
%-----

%?-supra(5,28,Z).

/*?-supra(5,27,Z),
write("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"),
nl,nl,write("KAMPANIAS GAUMARJOS!!!"),nl,
nl,nl,fail.*

%-----name-----price-----

drinks("wine kaxeti",2.5).
drinks("wine mukuzani",3.0).
drinks("wine rkatsiteli",2.5).
drinks("wine axasheni",3.0).
drinks("vodka rustavi",2.5).
drinks("vodka gudauro",3.0).
drinks("vodka gomi",1.0).


```
write(" lari."),nl,nl,  
    write(A),point,nl,  
    write(C),point,nl,  
    write(E),point,nl,  
    write(G),point,nl,nl.
```

ამოცანა 1. საქართველოს სამეფო დინასტიის გენეალოგიური ხე

ამოცანის სრული ტექსტი:

2-

window(____,win_func(), "Bagrationta Genealogiuri Xe", 200,100,520,400).

win_func(init):-

```
menu(pop_up,____,menu_window(), "&File"),
menu(pop_up,____,menu_go(), "&Asarchevi"),
menu(pop_up,____,menu_moor(), "&Mefeebi"),
menu(normal,____,menu_kidev(), "&Extend"),
menu(normal,____,menu_about(), "&About"),
```

window_brush(____,rgb(0,0,255)),

button(____,bfunc(), "Mtavari Shto", 190,170,140,25),

button(____,butfunc1(), "Kaxetis ", 150,235,100,25),

button(____,butfunc2(), "Imeretis", 280,235,100,25),

icon(____,fail(),default,45,280),

icon(____,fail(),default,75,280),

icon(____,fail(),default,105,280),

icon(____,fail(),default,135,280),

icon(____,fail(),default,165,280),

icon(____,fail(),default,195,280),

icon(____,fail(),default,225,280),

icon(____,fail(),default,255,280),

icon(____,fail(),default,45,160),

icon(____,fail(),default,45,100),

icon(____,fail(),default,65,40),

icon(____,fail(),default,435,280),

icon(____,fail(),default,115,100),

icon(____,fail(),default,165,100),

icon(____,fail(),default,215,40),

icon(____,fail(),default,265,40),

icon(____,fail(),default,315,100),

icon(____,fail(),default,365,100),

icon(____,fail(),default,415,40),

icon(____,fail(),default,435,100),

icon(____,fail(),default,435,160),

icon(____,fail(),default,435,220),

icon(____,fail(),default,450,350),

icon(____,fail(),default,50,350),

icon(____,fail(),default,285,280),

icon(____,fail(),default,315,280),

icon(____,fail(),default,345,280),

icon(____,fail(),default,375,280),

icon(____,fail(),default,45,250),

icon(____,fail(),default,45,220),

icon(____,fail(),default,45,190),

icon(____,fail(),default,45,130),

icon(____,fail(),default,45,70),

icon(____,fail(),default,405,280),

icon(____,fail(),default,90,70),

icon(____,fail(),default,140,130),

icon(____,fail(),default,190,70),

icon(____,fail(),default,240,10),

icon(____,fail(),default,290,70),

icon(____,fail(),default,340,130),

icon(____,fail(),default,390,70),

icon(____,fail(),default,435,70),

icon(____,fail(),default,435,130),

icon(____,fail(),default,435,190),

icon(____,fail(),default,435,250),

static(____,fail(), "Gmadlobt Sakartvelos Istoriisadmi Interesistvis", 100,355,330,15),

static(____,fail(), "Mefeebi", 230,210,70,20).

win_func(close):-not(yes_no("Exit", "Gsurt Gamosvla?", ?)).

```

menu_exit(press):-close_window().

menu_window(init):-menu(normal,_,menu_exit(),"Exit").

menu_about(press):-message("About", "Informacia Mopovebulia Istoriul Mecnierebata
Kandidatis Nodar Shoshitashvilis Gamokvlevidan", n).

menu_kidev(press):-
    size(_,X, Y),
    size(_,X, Y+9),
    Y<470,
    menu_kidev(press).

menu_moor(init):-menu(normal,_,fanjaral(),"Kaxetis"),
    menu(normal,_,fanjaral2(),"Imeretis").

menu_go(init):-

    menu(normal,_,s1(),"Mefe"),
    menu(normal,_,s2(),"Mefobis Clebi"),
    menu(normal,_,s3(),"Mefobis mxare").

fanjaral(press):-window(_,shto1(),"Kaxetis Mefeebi",250,150,700,500).

shto1(init):-
    window_brush(_,rgb(0,255,0)),
    static(_,fail(),"GIORGI VIII",280,15,140,15),
    static(_,fail(),"ALEKSANDRE I",280,48,140,15),
    static(_,fail(),"GIORGI II",280,81,140,15),
    static(_,fail(),"LEVAN I",280,114,140,15),
    static(_,fail(),"ALEKSANDRE II",280,147,140,15),
    static(_,fail(),"KONSTANTINE I",160,193,130,15),
    static(_,fail(),"DAVIT I",410,193,130,15),
    static(_,fail(),"TEIMURAZ I",280,236,140,15),
    static(_,fail(),"DAVITI",280,269,140,15),
    static(_,fail(),"EREKLE I",280,302,140,15),
    static(_,fail(),"TEIMURAZ II",280,345,140,15),
    static(_,fail(),"EREKLE II",280,378,140,15),
    static(_,fail(),"KONSTANTINE",80,343,140,15),
    static(_,fail(),"DAVIT II",480,343,140,15).

shto1(paint):- line(80,50,160,50),    line(540,50,620,50),

    line(80,50,80,10),    line(540,50,540,10),
    line(80,10,100,30),    line(540,10,560,30),
    line(100,30,120,5),    line(560,30,580,5),
    line(120,5,140,30),    line(580,5,600,30),
    line(140,30,160,10),    line(600,30,620,10),
    line(160,10,160,50),    line(620,10,620,50),

```

line(275,10,425,10), line(350,33,350,43),
 line(275,10,275,33),
 line(275,33,425,33),
 line(425,33,425,10),

line(275,43,425,43), line(275,76,425,76),
 line(275,43,275,66), line(275,102,275,76),
 line(275,66,425,66), line(275,102,425,102),
 line(425,66,425,43), line(425,102,425,76),
 line(350,66,350,76), line(350,102,350,112),

line(275,112,425,112),	line(275,145,425,145),
line(275,112,275,135),	line(275,145,275,168),
line(275,135,425,135),	line(275,168,425,168),
line(425,135,425,112),	line(425,168,425,145),
line(350,135,350,145),	line(350,168,350,178),
line(225,178,475,178),	
line(225,178,225,188),	
line(475,178,475,188),	
line(150,188,300,188),	line(400,188,550,188),
line(150,188,150,211),	line(400,188,400,211),
line(150,211,300,211),	line(400,211,550,211),

line(300,211,300,188),	line(550,211,550,188),
line(475,211,475,221),	line(275,231,275,254),
line(475,221,350,221),	line(275,254,425,254),
line(350,221,350,231),	line(425,254,425,231),
line(275,231,425,231),	line(350,254,350,264),
line(275,264,425,264),	line(275,264,275,287),
line(275,287,425,287),	line(425,287,425,264),
line(350,287,350,297),	line(275,297,425,297),
line(275,297,275,320),	line(275,320,425,320),
line(425,320,425,297),	line(350,320,350,330),
line(150,330,550,330),	line(150,330,150,340),
line(550,340,550,330),	line(350,330,350,340),
line(75,340,225,340),	line(75,340,75,363),
line(75,363,225,363),	line(225,363,225,340),

line(275,340,425,340),	line(275,340,275,363),
line(275,363,425,363),	line(425,363,425,340),
line(475,340,625,340),	line(475,340,475,363),
line(475,363,625,363),	line(625,363,625,340),
line(350,363,350,373),	line(275,373,425,373),
line(275,373,275,396),	line(275,396,425,396),
line(425,396,425,373).	

fanjara2(press):-window(,_,shto2(_),"Imeretis Mefeebi",250,150,700,500).

shto2(init):-
 window_brush(,rgb(0,255,0)),

```

static(,fail(),"BAGRAT II",280,15,140,15),
static(,fail(),"ALEKSANDRE II",280,48,140,15),
static(,fail(),"BAGRAT III",280,81,140,15),
static(,fail(),"KONSTANTINE",280,116,140,15),
static(,fail(),"ROSTOMI",155,160,140,15),
static(,fail(),"GIORGI III",405,160,140,15),
static(,fail(),"ALEKSANDRE III",280,203,140,15),
static(,fail(),"BAGRAT IV",280,236,140,15),
static(,fail(),"ALEKSANDRE IV",280,269,140,15),
static(,fail(),"GIORGI VI",155,312,140,15),
static(,fail(),"SIMON",405,312,140,15),
static(,fail(),"GIORGI VII",155,365,140,15),
static(,fail(),"ALEKSANDRE V",405,365,140,15),
static(,fail(),"DAVIT II",155,398,140,15),
static(,fail(),"SOLOMON I",405,398,140,15).

```

shto2(paint):-

```

line(80,50,160,50), line(540,50,620,50),
line(80,50,80,10), line(540,50,540,10),
line(80,10,100,30), line(540,10,560,30),
line(100,30,120,5), line(560,30,580,5),
line(120,5,140,30), line(580,5,600,30),
line(140,30,160,10), line(600,30,620,10),
line(160,10,160,50), line(620,10,620,50),
line(275,10,425,10), line(350,33,350,43),
line(275,10,275,33),
line(275,33,425,33),
line(425,33,425,10),

```

```

line(275,43,425,43), line(275,76,425,76),
line(275,43,275,66), line(275,102,275,76),
line(275,66,425,66), line(275,102,425,102),
line(425,66,425,43), line(425,102,425,76),
line(350,66,350,76), line(350,102,350,112),
line(275,112,425,112), line(350,135,350,145),
line(275,112,275,135),

```

```

line(275,135,425,135), line(425,135,425,112),
line(225,145,475,145),
line(225,145,225,155),
line(475,145,475,155),
line(150,155,300,155),
line(150,155,150,178),
line(150,178,300,178),
line(300,178,300,155),
line(400,155,550,155),
line(400,155,400,178),
line(400,178,550,178),
line(550,178,550,155),
line(350,221,350,231),
line(275,231,275,254),
line(475,178,475,188),
line(475,188,350,188),
line(350,188,350,198),
line(275,198,275,221),
line(275,198,425,198),
line(275,221,425,221),
line(425,198,425,221),
line(350,254,350,264),
line(275,264,425,264),

```

line(275,231,425,231),
 line(275,254,425,254),
 line(425,254,425,231),
 line(425,287,425,264),
 line(350,287,350,297),

line(275,264,275,287),
 line(275,287,425,287),
 line(225,297,475,297),
 line(225,297,225,307),
 line(475,297,475,307),

line(150,307,300,307),
 line(150,330,300,330),
 line(400,307,550,307),
 line(400,330,550,330),
 line(225,330,225,340),
 line(350,340,350,350),
 line(225,350,225,360),
 line(150,360,300,360),
 line(150,383,300,383),
 line(400,360,550,360),
 line(400,383,550,383),
 line(225,383,225,393),
 line(150,393,300,393),
 line(150,416,300,416),
 line(400,393,550,393),
 line(400,416,550,416),

line(150,307,150,330),
 line(300,330,300,307),
 line(400,307,400,330),
 line(550,330,550,307),
 line(225,340,350,340),
 line(225,350,475,350),
 line(475,350,475,360),
 line(150,360,150,383),
 line(300,383,300,360),
 line(400,360,400,383),
 line(550,383,550,360),
 line(475,383,475,393),
 line(150,393,150,416),
 line(300,416,300,393),
 line(400,393,400,416),
 line(550,416,550,393).

s1(press):-

```
read(X,"Mefe",s),nl,
  write(" Mefe"),write(" "),
  write("Mefobis Clebi"),write(" "),
  write("Mxare"), write(" "),
  nl,nl,
  mefe(X,Y,Z),
  write(X), write(" "),
  write(Y), write(" "),
  write(Z), write(" "),
  nl,fail.
```

s2(press):-

```
read(Y,"Mefe",s),nl,
  write(" Mefe"),write(" "),
  write("Mefobis Clebi"),write(" "),
  write("Mxare"), write(" "),
  nl,nl,
  mefe(X,Y,Z),
  write(X), write(" "),
  write(Y), write(" "),
  write(Z), write(" "),
  nl,fail.
```

s3(press):-

```
read(Z,"Mxare",s),nl,
write(" Mefe"),write(" "),
write("Mefobis Clebi"),write(" "),
write("Mxare"), write(" "),
nl,nl,
mefe(X,Y,Z),
write(X), write(" "),
write(Y), write(" "),
write(Z), write(" "),
nl,fail.
```

butfunc1(press):-window(,shto1(),"Kaxetis Mefeebi",250,150,700,500).

butfunc2(press):-window(,shto2(),"Imeretis Mefeebi",250,150,700,500).

bfunc(press):-

window(,drow_func1(),"Ertiani Sakartvelos Mefeebi",250,150,700,500).

drow_func1(paint):-

line(80,50,160,50), line(540,50,620,50),

line(80,50,80,10), line(540,50,540,10),

line(80,10,100,30), line(540,10,560,30),

line(100,30,120,5), line(560,30,580,5),

line(120,5,140,30), line(580,5,600,30),

line(140,30,160,10), line(600,30,620,10),

line(160,10,160,50), line(620,10,620,50),

line(275,10,425,10), line(350,33,350,43),

line(275,10,275,33),

line(275,33,425,33),

line(425,33,425,10),

line(275,43,425,43), line(275,76,425,76),

line(275,43,275,66), line(275,102,275,76),

line(275,66,425,66), line(275,102,425,102),

line(425,66,425,43), line(425,102,425,76),

line(350,66,350,76), line(350,102,350,112),

line(275,112,425,112), line(275,145,425,145),

line(275,112,275,135), line(275,145,275,168),

line(275,135,425,135), line(275,168,425,168),

line(425,135,425,112), line(425,168,425,145),

line(350,135,350,145), line(350,168,350,178),

line(275,178,425,178),

line(275,178,275,201),

line(275,201,425,201),

```

line(425,178,425,201),
line(350,201,350,211),
line(225,211,475,211),
line(225,211,225,221),
line(475,211,475,221),
line(150,221,300,221),
line(150,221,150,244),
line(150,244,300,244),
line(300,244,300,221),
line(225,244,225,254),
line(150,254,300,254),
line(150,254,150,277),
line(150,277,300,277),
line(300,277,300,254),
line(225,277,225,287),
line(150,287,300,287),
line(150,287,150,310),
line(150,310,300,310),
line(300,310,300,287),
line(540,450,620,450),
line(540,450,540,410),
line(540,410,560,430),
line(560,430,580,405),
line(400,221,550,221),
line(400,221,400,244),
line(400,244,550,244),
line(550,244,550,221),
line(80,450,80,410),
line(80,450,160,450),
line(160,450,160,410),
line(80,410,100,430),
line(100,430,120,405),
line(120,405,140,430),
line(140,430,160,410),
line(580,405,600,430),
line(600,430,620,410),
line(620,410,620,450),

```

beep.

```
drow_func1(init):- window_brush(_rgb(0,255,0)),
```

```

static(_,_fail(_),"BAGRAT III",280,15,140,15),
static(_,_fail(_),"GIORG I",280,48,140,15),
static(_,_fail(_),"BAGRAT IV",280,81,140,15),
static(_,_fail(_),"GIORG II",280,114,140,15),
static(_,_fail(_),"DAVIT AGMASHNEBELI",280,147,140,15),
static(_,_fail(_),"DEMETRE I",280,180,140,15),
static(_,_fail(_),"GIORG III",160,226,130,15),
static(_,_fail(_),"TAMARI",160,259,130,15),
static(_,_fail(_),"DAVIT V",410,226,130,15),
static(_,_fail(_),"LASHA GIORG IV",160,292,140,15),
static(_,_fail(_),"ERTIANI SAKARTVELO MEFEebi",250,340,230,20).

```

```
%*****|*****
```

```

mefe("adarnase", "779", "klarjeti").
mefe("ashot I kurapalati", "826", "klarjeti").
mefe("bagrat I", "830-876", "klarjeti").
mefe("davit I", "876-881", "klarjeti").
mefe("adarnase 2", "888-923", "klarjeti").
mefe("bagrat I", "937-945", "klarjeti").
mefe("adarnase 3", "958-961", "klarjeti").
mefe("davit 3 didi", "1001", "tao-klarjeti").
mefe("sumbat kurapalati", "954-958", "klarjeti").
mefe("davit 2", "923-937", "tao-klarjeti").

```

mefe("bagrat 2","958-994","tao-klarjeti").
 mefe("adarnase 4","961-983","tao-klarjeti").
 mefe("gurgun 2","1008","tao-klarjeti").
 mefe("bagrat 3","957-1014","ertiani sakartvelo").
 mefe("giorgi 1","1014-1027","ertiani sakartvelo").
 mefe("Davit Agmashenebeli","1089-1125","ertiani sakartvelo").
 mefe("bagrat 4","1027-1072","ertiani sakartvelo").
 mefe("giorgi 2","1072-1089","ertiani sakartvelo").
 mefe("demetre 1","1125-1156","ertiani sakartvelo").
 mefe("giorgi 3","1156-1184","ertiani sakartvelo").
 mefe("tamari","1184-1213","ertiani sakartvelo").
 mefe("davit soslani","1189-1207","ertiani sakartvelo").
 mefe("rusudani","1223-1245","sakartvelo").
 mefe("davit 5","1155","ertiani sakartvelo").
 mefe("davit narini","1247-1293","sakartvelo").
 mefe("demetre 2","1270-1289","sakartvelo").
 mefe("giorgi 5 brcyinvale","1318-1346","sakartvelo").
 mefe("lasha giorgi","1213-1223","ertiani sakartvelo").

mefe("davit 7 ulu","1247-1270","sakartvelo").
 mefe("vaxtang 3","1302-1308","sakartvelo").
 mefe("davit 8","1293-1311","sakartvelo").
 mefe("giorgi 6 mcire","1308-1318","sakartvelo").
 mefe("davit 9","1346-1360","sakartvelo").
 mefe("bagrat 5","1360-1393","sakartvelo").
 mefe("konsantine 1","1407-1412","sakartvelo").
 mefe("giorgi 7","1393-1407","sakartvelo").
 mefe("aleksandre 1","1412-1442","sakartvelo").
 mefe("bagrat 6","1466-1478","kartl-kaxeti").
 mefe("demetre","1453","provincia").
 mefe("vaxtang 6","1442-1446","sakartvelo").
 mefe("giorgi 8","1446-1466","sakartvelo").
 mefe("giorgi 8","1466-1476","kaxeti").
 mefe("konstantine 2","","kartli").
 mefe("aleksandre 1","1476-1511","kaxeti").
 mefe("giorgi 2","1511-1513","kaxeti").
 mefe("levan 1","1518-1574","kaxeti").
 mefe("aleksandre 2","1574-1605","kaxeti").
 mefe("davit 1","1601-1602","kaxeti").
 mefe("konstantine 1","1605","kaxeti").

mefe("teimuraz 1","1606-1648","kaxeti").
 mefe("teimuraz 1","1625-1633","kaxeti").
 mefe("daviti","16??-16??","kaxeti").
 mefe("erekle 1","1703-1709","kaxeti").
 mefe("erekle 1","1688-1703","kartli").
 mefe("konstantine","1722-1732","kaxeti").
 mefe("teimuraz 2","1732-1744","kaxeti").
 mefe("teimuraz 2","1744-1762","kartli").
 mefe("davit 2","1703-1722","kaxeti").
 mefe("erekle 2","1744-1762","kaxeti").
 mefe("bagrat 6","1463-1478","imereti").
 mefe("aleksandre 2","1484-1510","imereti").

```

mefe("bagrat 3","1510-1565","imereti").
mefe("giorgi 2","1565-1585","imereti").
mefe("levan","1585-1590","imereti").
mefe("konstantine","","imereti").
mefe("rostop","1590-1605","imereti").
mefe("giorgi 3","1605-1639","imereti").
mefe("aleksandre 3","1639-1660","imereti").
mefe("bagrat 4","1660-1681","imereti").
mefe("aleksandre 4","1683-1695","imereti").
mefe("giorgi 6","1703-1720","imereti").
mefe("simon","1698-1701","imeretis mefe").
mefe("giorgi 7","1741","imereti").
mefe("davit 2","1784-1789","imereti").
mefe("aleksandre 5","1720-1752","imereti").
mefe("solomon 1","1752-1784","imereti").
mefe("archil ","","imereti").
mefe("solomon 2","1789-1810","imereti").
mefe("konstantine 2","1478-1505","kartli").
mefe("davit 10","1505-1525","kartli").
mefe("giorgi 9","1525-1527","kartli").
mefe("bagrat muxranbatoni","","kartli").
mefe("luarsab 1","1527-1556","kartli").
mefe("simon 1","1556-1600","kartli").
mefe("davit 11","1569-1578","kartli").
mefe("giorgi 10","1600-1606","kartli").
mefe("luarsab 2","1606-1615","kartli").
mefe("bagrat 7","1616-1619","kartli").
mefe("simon 2","1619-1631","kartli").
mefe("rostop","1633-1658","kartli").
mefe("vaxtang 5","1658-1675","kartli").
mefe("giorgi 11","1676-1688","kartli").
mefe("levan","","kartli").
mefe("vaxtang 6","1716-1724","kartli").
mefe("qaixosro","1706-1711","kartli").
mefe("iese","1714-1716","kartli").
mefe("baqari","1717-1719","kartli").

```

.....

ამოცანა 2. შემსრულებელი (exeCutor).

***This program will only work correctly on MS Windows XP platform

ამოცანის სრული ტექსტი:

```

?-
window_n(,win_func,"exeCutor v.1.4beta",200,200,430,430,
_,rgb(125,225,0)).

win_func(init):-
menu(pop_up,_,menu_file(),"File"),

```

```

menu(pop_up,_,_,menu_about(_),"About"),

static(G_E1,_,stf1(_),"Executed in ",300,355,80,20),
static(G_E2,_,stf2(_),"",380,355,40,20),

button(,_,bf1(_),"Winamp",10,85,70,20),
button(,_,bf2(_),"Explorer",90,85,70,20),
button(,_,bf3(_),"Console",170,85,70,20),
button(,_,bf4(_),"DirectX",250,85,70,20),
button(,_,bf5(_),"Calculator",330,85,70,20),
button(,_,bf6(_),"MPlayer",10,285,70,20),
button(,_,bf7(_),"Notepad",90,285,70,20),
button(,_,bf8(_),"Freecell",170,285,70,20),
button(,_,bf9(_),"MSHearts",250,285,70,20),
button(,_,bf10(_),"WinMine",330,285,70,20),

beep("C:\\WINDOWS\\Media\\tada.wav"),
%beep("C:\\My Music\\super musik\\Universiteti\\1.wav"),
%beep("C:\\My Music\\super musik\\nato gelashvili\\kuku.wav"),
%beep("C:\\My Music\\bubu.wav"),
%beep("C:\\My Music\\kviteli.wav"),
%beep("C:\\My Music\\nino.wav"),

X is print("C:\\Documents and Settings\\All Users\\Application Data\\Microsoft\\User
Account Pictures\\Default Pictures\\"),

bitmap(,_,f1(_), X + print("drip.bmp"),20,30),
bitmap(,_,f2(_), X + print("airplane.bmp"),100,30),
bitmap(,_,f3(_), X + print("guitar.bmp"),180,30),
bitmap(,_,f4(_), X + print("frog.bmp"),260,30),
bitmap(,_,f5(_), X + print("chess.bmp"),340,30),
bitmap(,_,f6(_), X + print("butterfly.bmp"),20,230),
bitmap(,_,f7(_), X + print("horses.bmp"),100,230),
bitmap(,_,f8(_), X + print("ball.bmp"),180,230),
bitmap(,_,f9(_), X + print("cat.bmp"),260,230),
bitmap(,_,f10(_), X + print("beach.bmp"),340,230).

win_func(paint):-
pen(2,rgb(0,0,0)),
line(0,40,430,40),
line(0,95,430,95),
line(0,240,430,240),
line(0,295,430,295),
line(0,352,430,352).

win_func(close):- not(yes_no("Exit","Are You Sure? :)",?)).

stf1(init):-
color_text_back(G_E1, rgb(0,225,225)).

stf2(init):-
color_text_back(G_E2, rgb(0,225,225)).

```

```

menu_file(init):-
menu(normal_1_,opf(_),"Open File..."),
menu(normal_1_,rmf(_),"Delete Record"),
menu(separator_1_,spf(_),_),
menu(normal_1_,exf(_),"Exit").

opf(press):-
X is select_file(o_,"*.pro"),
execute("C:\\Program Files\\Strawberry Prolog\\Prolog.exe"
+print(" ") +print(X)).

rmf(press):-
retract_in(file(X,T)).

exf(press):- close_window(_).

menu_about(init):-
menu(normal_1_,af(_),"Author"),
menu(normal_1_,df(_),"Dedication"),
menu(normal_1_,pf(_),"About").

af(press):-
message("We Made It:", "Program was created by
M.Belmasova aka Banshee
with assistance of
A.Moskalenko :) (2005)",i).

df(press):-
beep("C:\\WINDOWS\\Media\\chimes.wav"),
message("Whom for?", "Dedicated to our lector
Besarion Kiziria :)"),i).

pf(press):-
message("About", "This program
can execute programs,
store their names and paths
in the database and count
periods of their
execution.",i).

bfl(press):-
X is ("C:\\Program Files\\Winamp\\winamp.exe"),
chronometer(_),
execute(X),
chronometer(T),
assert_in(file(X,T)),nl,
Y is print(T),
set_text(Y, G_E2).

bf2(press):-
X is ("C:\\WINDOWS\\explorer.exe"),

```

```
chronometer(_),
execute(X),
chronometer(T),
assert_in(file(X,T)),nl,
Y is print(T),
set_text(Y, G_E2).
```

```
bf3(press):-
X is ("C:\\WINDOWS\\system32\\cmd.exe"),
chronometer(_),
execute(X),
chronometer(T),
assert_in(file(X,T)),nl,
Y is print(T),
set_text(Y, G_E2).
```

```
bf4(press):-
X is ("C:\\WINDOWS\\system32\\dxdiag.exe"),
chronometer(_),
execute(X),
chronometer(T),
assert_in(file(X,T)),nl,
Y is print(T),
set_text(Y, G_E2).
```

```
bf5(press):-
X is ("C:\\WINDOWS\\system32\\calc.exe"),
chronometer(_),
execute(X),
chronometer(T),
assert_in(file(X,T)),nl,
Y is print(T),
set_text(Y, G_E2).
```

```
bf6(press):-
X is ("C:\\WINDOWS\\system32\\mplay32.exe"),
chronometer(_),
execute(X),
chronometer(T),
assert_in(file(X,T)),nl,
Y is print(T),
set_text(Y, G_E2).
```

```
bf7(press):-
X is ("C:\\WINDOWS\\notepad.exe"),
chronometer(_),
execute(X),
chronometer(T),
assert_in(file(X,T)),nl,
Y is print(T),
set_text(Y, G_E2).
```

```
bf8(press):-
X is ("C:\\WINDOWS\\system32\\freecell.exe"),
chronometer(_),
execute(X),
chronometer(T),
assert_in(file(X,T)),nl,
Y is print(T),
set_text(Y, G_E2).
```

```
bf9(press):-
X is ("C:\\WINDOWS\\system32\\mshearts.exe"),
chronometer(_),
execute(X),
chronometer(T),
assert_in(file(X,T)),nl,
Y is print(T),
set_text(Y, G_E2).
```

```
bf10(press):-
X is ("C:\\WINDOWS\\system32\\winmine.exe"),
chronometer(_),
execute(X),
chronometer(T),
assert_in(file(X,T)),nl,
Y is print(T),
set_text(Y, G_E2).
```

.....

ამოცანა 3. საქართველოს ისტორიული ძეგლები

საკურსო სამუშაოს აღწერის ტექსტი:

საძარტველოს ისტორიული ძეგლები

პროგრამა წარმოადგენს მონაცემთა ბაზას, რომელიც შეიცავს ინფორმაციას საქართველოს ისტორიული ძეგლების შესახებ. მასში შესაძლებელია სასურველი ძეგლის ძიება სხვა და სხვა პარამეტრის მიხედვით : სახელი, ტიპი (ეკლესია, მონასტერი, სასახლე, ციხესიმაგრე), წელი (აშენების თარიღი), საუკუნე (აშენების საუკუნე), ქალაქი (თუ რომელ ქალაქში თუ რაიონშია აგებული), კუთხე (საქართველოს თუ რომელ ისტორიულ კუთხეშია აგებული), ქვეყანა, დამაარსებელი. ძიების შედეგია არჩეული ძეგლის ან ძეგლების მოკლე აღწერა, რომელსაც თან ერთვის ფოტოსურათი.

პროგრამა იწყება შემდეგი მისიით:

```
?-window(_w_main(),"istoriuli Zeglebi",200,200,355,290).
```

wndow – არის პრედიკატი, რომელიც გამოიყენება ფანჯრის შესაქმნელად. მისი ზოგადი სახეა :

window(Handle,Parent,Win_func(),Title,X,Y,Width,Height).

Handle – არის ფანჯრის სახელი, რომლის მეშვეობითაც შემდგომში შესაძლებელია ამ ფანჯარასთან მუშაობა. ძირითადად გამოიყენება გლობალური ცვლადები, მაგრამ ჩვენს შემთხვევაში იგი არაა საჭირო და ვიყენებთ ფიქტიურ ცვლადს **()**.

Parent – არის ე.წ. ძირითადი ფანჯრის „მშობელი“ ფანჯრის სახელი. ჩვენ აქაც ვიყენებთ ფიქტიურ პარამეტრს.

Title – არის ფანჯრის სათაური. ჩვენს შემთხვევაში **"istoriuli Zeglebi"**.

X,Y – წარმოადგენენ ფანჯრის მარცხენა ზედა კუთხის კოორდინატებს. ხოლო **Width** და **Height** შესაბამისად ფანჯრის სიგრძე და სიმაღლე პიქსელებში.

Win_func() – არის მოცემული ფანჯრის ფუნქციის სახელი. იგი შეიძლება იყოს ნებისმიერი, ჩვენს შემთხვევაში **w_main()**. მისი მეშვეობით შესაძლებელია ფანჯარაზე სხვა და სხვა სახის მანიპულაციების განხორციელება. ფუნქციას შეიძლება აქონდეს პარამეტრები, მაგრამ ჩვენ მათ არ ვიყენებთ და შესაბამისად ვუთითებთ ფიქტიურ პარამეტრს. ქვემოთ მოყვანილია ამ ფუნქციის რეალიზაცია ჩვენს პროგრამაში:

w_main(init):-

```
    bitmap(_,_bitmap_func(),"C:\istoriuli Zeglebi\SYSTEM\ menu.bmp",0,0),
    menu(pop_up_._,way_out(),"File"),
    menu(pop_up_._,view_func(),"Windows"),
    menu(normal_._,info(),"Info"),
    button(_,_dat_func(),"DATVALIEREBA",5,3,120,30),
    button(_,_dam_func(),"DAMATEBA",220,3,120,30),
    button(_,_p_func(),"!",3,50,20,20).
```

ამ ფუნქციას ვიყენებთ ფანჯრის სხვა და სხვა კომპონენტების შესაქმნელად. ესენია **bitmap**, **menu** და **button**. ფუნქცია ღებულობს პარამეტრს **init**, რომელიც ინიციალიზებას უკეთებს ფანჯრის ელემენტებს.

Bitmap–ის ანუ გამოსახულების განსათავსებლად ფანჯარაზე გამოიყენება პრედიკატი **bitmap**, რომლის ზოგადი სახეც შემდეგია:

bitmap(Handle,Parent,Bitmap_func(),Bitmap,X,Y).

მისი ზოგიერთი პარამეტრი ემთხვევა **wndow** პრედიკატის პარამეტრებს: **Handle,Parent,X,Y**. მათ თავიდან არ განვიხილავთ.

Bitmap–პარამეტრი წარმოადგენს გზას გამოსახულებიან ფაილამდე. ჩვენს მაგალითში "C:\istoriuli Zgleb\SYSTEM\ menu.bmp". გასათვალისწინებელია, რომ პროლოგი მუშაობს მხოლოდ .BMP გაფართოების გამოსახულებიან ფაილებთან.

Bitmap_func() – პარამეტრი წარმოადგენს ფუნქციას, რომლის მეშვეობითაც შესაძლებელია მინიპულაციების განხორციელება გამოსახულებაზე.

ფანჯრის მენიუს შესაქმნელად გამოიყენება პრედიკატი **menu** :

menu(Type,Handle,Parent,Menu_func(),Text),

Handle Parent პარამეტრები ზემოთგანიხილულის ანალოგიურია.

Type განსაზღვრავს მენიუს ტიპს. პროლოგში გამოიყენება რამოდენიმე ტიპის მენიუ, ესენია : **pop_up** და **normal**, და ა.შ. პირველი მათგანი გამოიყენება ქვემენიუების, ხოლო მეორე კი ცალკეული მენიუების შესაქმნელად. **Text** პარამეტრი განსაზღვრავს მენიუს სახელს. მაგალითად ჩვენს მიერ შერჩეულია შემდეგი სახელები : "File","Windows" და "Info".

ჩვენს შემთხვევაში ფუნქცია **Menu_func** წარმოდგენილია **info()**, **view_func()** და **way_out()** ფუნქციების სახით. ისინი გამოიყენება იმისათვის რომ ყოველ მენიუს მიეანიჭოთ შესაბამისი ფუნქცია. მათი დაწკრილებითი განმარტება მოცემულია მოგვიანებით.

button(Handle,Parent,Button_func(),Title,X,Y,Width,Height) – პრედიკატი გამოიყენება ფანჯარაში ღილაკის შესაქმნელად. მისი პარამეტრები უკვე აღწერილია. ჩვენს პროგრამაში იქმნება სამი ასეთი ღილაკი : "DATVALIEREBA", "DAMATEBA" და "".

განიხილოთ ის ფუნქციები, რომლებიც გამოიყენებოდა აღნიშნულ პრედიკატებში.

w_main(close):-

not(yes_no("Exit","Do you want to Exit ?",?)).

მოცემული წესი გამოიყენება იმისათვის, რომ შემთხვევით არ მოხდეს პროგრამიდან გამოსვლა. ანუ. თუკი თქვენ შემთხვევით დააჭირეთ პროგრამიდან გამოსვლის ღილაკს, პროგრამა არ გაითიშება, არამედ შეგვკითხვბათ: ნამდვილად გსურთ თუ არა ამის გაკეთება ? ეს მიიღწევა **yes_no** პრედიკატის გამოყენებით.

yes_no(Title,Text,Icon).

Title – არის დიალოგური ფანჯრის სათაური, ხოლო **Text** კი – მისი ტექსტი. ჩვენს შემთხვევაში, შესაბამისად "Exit" და "Do you want to Exit ?".

Icon პარამეტრის ადგილას შესაძლებელია რამოდენიმე ეარიანტის ჩასმა.

ესენია: ? – კოთხეა. **s** – STOP, **i** –ინფორმაცია, **n** – ცარიელი შეტყობინება.

way_out(init):-

menu(normal,_,menu_exit(),"Exit").

menu_exit(press):-close_window().

way_out ფუნქცია გამოიყენება File მენიუს Exit ქვემენიუს შესაქმნელად. თავის მხრივ Exit მენიუს ფუნქცია ხურავს ძირითად ფანჯარას, რადგანაც მის პარამეტრად გამოიყენებულია ფიქტიური პარამეტრი, რაც გულისხმობს ძირითად ფანჯარას.

```
view_func(init):-  
    menu(normal_ც,dat_func(),"DATVALIEREBA"),  
    menu(normal_ც,dam_func(),"DAMATEBA").
```

ეს ფუნქცია ინიციალიზებას უკეთებს ორ ქვემენიუს "DATVALIEREBA" "DAMATEBA", რომელთაც თავისმხრივ გააჩნიათ თავიანთი ფუნქციები: dat_func() და dam_func().

dat_func ფუნქცია გამოიყენება, როგორც "DATVALIEREBA" მენიუს მიერ, ასევე ამავე სახელწოდების ლილაკის მიერაც. ასევე dam_func ფუნქციას იყენებენ "DAMATEBA" სახელის მქონე მენიუ და ლილაკი. ფუნქცია

```
dat_func(press):-  
    window(ც,w_dat(),"DATVALIEREBA",5,5,410,350)
```

გამოიყენება ახალი ფანჯრის შესაქმნელად სახელწოდებით "DATVALIEREBA", ხოლო dam_func(press):-window(ც,w_dam(),"DAMATEBA",410,5,410,550). კი ფანჯრის სახელად "DAMATEBA".

```
info(press):-window(ც,w_info(),"info",150,150,500,400).
```

ამ ფუნქციას გამოაქვს ფანჯარა სახელად "info", რომელსაც ინიციალიზაციას უკეთებს ფუნქცია :

```
w_info(init):-  
    bitmap(ც,fail(),"C:\istoriuli Zegleb\SYSTEM\info_menu.bmp",0,0).
```

მასზე განთავსებულია გამოსახულება, ინფორმაციით პროგრამის შემქმნელთა შესახებ. მასაც იყენებს ძირითადი ფანჯრის ორი კომპონენტი : "!" კლავიში და "Info" მენიუ.

შემდეგი ფუნქცია ინიციალიზებას უკეთებს ფანჯარას "DAMATEBA":

```
w_dam(init):-  
    bitmap(ც,fail(),"C:\istoriuli Zegleb\SYSTEM\menu DAMATEBA.bmp",0,0),  
    menu(normal_ც,help1(),"help"),  
    edit(G_P1_ც,edit_func(),"",200,10,200,25),  
    edit(G_P2_ც,edit_func(),"",200,60,200,25),  
    edit(G_P3_ც,edit_func(),"",200,110,200,25),  
    edit(G_P4_ც,edit_func(),"",200,160,200,25),  
    edit(G_P5_ც,edit_func(),"",200,210,200,25),  
    edit(G_P6_ც,edit_func(),"",200,260,200,25),  
    edit(G_P7_ც,edit_func(),"",200,310,200,25),  
    edit(G_P8_ც,edit_func(),"",200,360,200,25),
```

```
edit(G_P9,edit_func(),"",200,410,200,25),
button(_,_damat(),"SAVE",150,450,100,30).
```

ამ ფანჯარასე განთავსებულია ერთი გამოსახულება, ერთი მენიუ, ერთი ლილაკი და ტექსტის შესაყვანი რვა სპეციალური უბანი, რომელიც იქმნება edit პრედიკატის მიერ. ამ პრედიკატის პარამეტრები იდენტურია window პრედიკატის პარამეტრებისა, რომელიც უკვე განვიხილეთ.

```
edit(Handle.Parent,Edit_func(),Title,X,Y,Width,Height).
```

ყოველ ასეთ უბანს ეანჭებთ შესაბამის გლობალურ ცვლადს G_P1–GP9, რადგანაც მომავალში ისინი აუცილებლად დაგეჭირდება.

"help" მენიუს help1 ფუნქცია კმნის ახალ –"help" ფანჯარას, რომელზეც w_help1 ფუნქცია განთავსებს გამოსახულებას ამ ფანჯარასთან მუშაობის წესებით :

```
help1(press):-
```

```
window(_,_w_help1(),"help",100,10,510,550).
```

```
w_help1(init):-
```

```
bitmap(_,_fail(),"C:\istoriuli Zegleb\SYSTEM\help1.bmp",0,0).
```

განვიხილოთ "SAVE" ლილაკის ფუნქცია, რომელიც საშუალებას გვაძლევს პროგრამის ცოდნის ბაზაში დაეამატოთ ახალი მონაცემები :

```
damat(press):-
```

```
A is get_text(G_P1),
```

```
B is get_text(G_P2),
```

```
C is get_text(G_P3),
```

```
D is get_text(G_P4),
```

```
E is get_text(G_P5),
```

```
F is get_text(G_P6),
```

```
G is get_text(G_P7),
```

```
H is get_text(G_P8),
```

```
I is "C:\istoriuli Zegleb\PHOTO\"+ get_text(G_P9)+ ".bmp",
```

```
assert_in(dzegli(A,B,C,D,E,F,G,H,I),
```

```
message("Congratulations !","Tqven daamatet informacia monacemta bazaSi !",i),
```

```
set_text("",G_P1),
```

```
set_text("",G_P2),
```

```
set_text("",G_P3),
```

```
set_text("",G_P4),
```

```
set_text("",G_P5),
```

```
set_text("",G_P6),
```

```
set_text("",G_P7),
```

```
set_text("",G_P8),
```

```
set_text("",G_P9).
```

გლობალური ცვლადების მეშვეობით, რომლებიც თავიდან მიეანიჭეთ edit კომპონენტებს და set_text ფუნქციის გამოყენებით A,B,C,D,E,F,G,H,I ცვლადებს ენიჭებათ შესაბამის edit კომპონენტში ჩაწერილი ტექსტი. ამის შემდეგ ეს ახალი მონაცემები ბაზაში ჩაიწერება assert_in ფუნქციის მეშვეობით. მოცემული ოპერაციების წარმატების შემთხვევაში გამოდის დიალოგური ფანჯარა (message), რომელიც გეატყობინებს, რომ მოხდა ცოდნის ბაზის შევსება.

set_text ფუნქციით ხორციელდება edit კომპონენტის გასუფთავება, ცარიელი ტექსტის ("") შეყვანიტ.

შემდეგ ხდება "DATVALIEREBA" ფანჯრის ინიციალიზება, რომელიც მსგავსია უკვე განხილული ფანჯრების ინიციალიზებისა, ამიტომ მას არ განვიხილავთ, მხოლოდ მოვიყვანო მის ფრაგმენტს:

```
w_dat(init):-
```

```
    bitmap( fail(), "C:\istoriuli Zglebi\SYSTEM\menu DATVALIEREBA.bmp",0,0),
```

```
    enu(pop_up, zeb(), "Search"),
```

```
    enu(normal, help(), "help"),
```

```
    button( saxel_func(), "saxeli", 3, 3, 100, 25),
```

```
    button( tipi_func(), "tipi", 3, 100, 70, 25),
```

```
    button( weli_func(), "weli", 3, 160, 70, 25),
```

```
    button( sauk_func(), "saukune", 3, 260, 100, 25),
```

```
    button( qalaq_func(), "qalaqi", 290, 3, 100, 25),
```

```
    button( kuTxe_func(), "kuTxe", 320, 100, 70, 25),
```

```
    button( qvey_func(), "qveyana", 320, 160, 70, 25),
```

```
    button( damaars_func(), "damaarsebeli", 290, 260, 100, 25).
```

```
zeb(init):-
```

```
    enu(normal, saxel_func(), "saxeli"),
```

```
    enu(normal, tipi_func(), "tipi"),
```

```
    enu(normal, weli_func(), "weli"),
```

```
    enu(normal, sauk_func(), "saukune"),
```

```
    enu(normal, qalaq_func(), "qalaqi"),
```

```
    enu(normal, kuTxe_func(), "kuTxe"),
```

```
    enu(normal, qvey_func(), "qveyana"),
```

```
    enu(normal, damaars_func(), "damaarsebeli").
```

```
help(press):-
```

```
    window( w_help(), "help", 100, 10, 510, 500).
```

```
w_help(init):-
```

bitmap(ღ, fail(ღ, "C:\istoriuli Zegleb\SYSTEM\help.bmp", 0, 0).

"DATA\ALIEREBA" ფანჯარაში იქმნება 8 დილაკი და იგივე ფუნქციების მქონე 8 მენიუ, რომელთა მეშვეობითაც შესაძლებელია სასურველი ისტორიული ძეგლის ძიება სხვა და სხვა პარამეტრებით: "saxeli", "tipi", "weli", "saukune", "qalaqi", "kuTxe", "qveyana", "damaarsebeli". ყოველი პარამეტრისათვის რეალიზებულია ცალკე ფუნქცია, რომელთა მუშაობის პრინციპიც ერთნაირია, ამიტომ განვიხილავთ მხოლოდ ერთს :

```
qalaq_func(press):-  
    read(G_Saxeli,"saxeli",s),  
    dzeqli(G_Saxeli,A,B,C,D,E,F,G,G_P),  
    write("      "),write(G_Saxeli),nl,  
    write("-----"),nl,  
    write(" tipi          : "),write(A),nl,  
    write(" weli          : "),write(B),nl,  
    write(" saukune        : "),write(C),nl,  
    write(" qalaqi         : "),write(D),nl,  
    write(" kuTxe          : "),write(E),nl,  
    write(" qveyana        : "),write(F),nl,  
    write(" damaarsebeli : "),write(G),nl,nl,nl,  
    far_way(ღ, fail.
```

მოცემული ფუნქციის პარამეტრია **press**, რაც იმაზე მიუთითებს, რომ იგი აქტიურდება შესაბამის დილაკზე თუ მენიუზე დაჭერის შედეგად. ფუნქციის ამუშავებისთანავე გამოდის სტანდარტული ფანჯარა ინფორმაციის შექცევისათვის. მასში შეტანილი ინფორმაცია თავსდება **G_Saxeli** გლობალურ ცვლადში.

```
read(G_Saxeli,"saxeli",s),
```

read პრედიკატის მეორე პარამეტრი წარმოადგენს **read** ფანჯრის სათაურს, ხოლო მესამე პარამეტრის ადგილას შეიძლება იყოს ორი სიმბოლოდან ერთ-ერთი : **s** ან **f**. **s** აღნიშნავს ტექსტს და ამიტომ **read**-ში წყენს მიერ შეტანილ ინფორმაციას არ სჭირდება ბრჭყალები. შემდეგ მოდის მიზანი : **dzeqli(G_Saxeli,A,B,C,D,E,F,G,G_P)**, რომელშიც გამოიყენება ორი გლობალური ცვლადი : **G_Saxeli** ისტორიული ძეგლის სახელი და **G_P** – სრული გზა შესაბამისი ისტორიული ძეგლის ფოტო ფაილამდე. ეს ცვლადები გამოიყენება ფუნქციაში – **far_way(ღ)**. თავიდან **dzeqli** პრედიკატის მხოლოდ **G_Saxeli** პარამეტრია განსაზღვრული, შემდეგ ხორციელდება შესაბამისი ფაქტის ძებნა ცოდნის ბაზაში. მას შემდეგ, რაც მოიძებნება შესაბამისი სახელის მქონე ძეგლი, დანარჩენ ცვლადებსაც მიენიჭებათ მნიშვნელობები. შემდეგ ეს მნიშვნელობები გამოგვაქვს ეკრანზე **write** ფუნქციების მეშვეობით. შემდეგ საჩქეში ერთდება ფუნქცია **far_way(ღ)**, რომე

ლსაც ეკრანზე გამოაქვს შესაბამისი ისტორიული ძეგლის ფოტოსურათი, ამისათვის იგი იყენებს 'ზემოთაღნიშნულ ორ გლობალურ ცვლადს:

```
far_way():-  
    window(⌒,fan_func(),G_Saxeli,50,50,405,375).  
fan_func(init):-  
    bitmap(⌒,fail(),G_P,0,0).
```

.....

აქვე მოვიყვანოთ ამავე პროგრამის სრული ტექსტი:

```
?-window(⌒,w_main(),"Istoriuli dzegebi",200,200,355,290).  
w_main(init):-  
    bitmap(⌒,bitmap_func(),"C:/istoriuli Zeglebi/SYSTEM/I menu.bmp",0,0),  
    menu(pop_up,⌒,way_out(),"File"),  
    menu(pop_up,⌒,view_func(),"Windows"),  
    menu(normal,⌒,info(  
    ),"Info"),  
    button(⌒,dat_func(),"DATVALIEREBA",5,3,120,30),  
    button(⌒,dam_func(),"DAMATEBA",220,3,120,30),  
    button(⌒,info(),"!" ,3,50,20,20).  
w_main(close):-  
    not(yes_no("Exit","Do you want to Exit ?","?)).  
way_out(init):-  
    menu(normal,⌒,menu_exit(),"Exit").  
menu_exit(press):-  
    close_window().  
view_func(init):-  
    menu(normal,⌒,dat_func(),"DATVALIEREBA"),  
    menu(normal,⌒,dam_func(),"DAMATEBA").  
info(press):-  
    window(⌒,w_info(),"info",150,150,500,400).  
dat_func(press):-  
    window(⌒,w_dat(),"DATVALIEREBA",5,5,410,350).  
dam_func(press):-  
    window(⌒,w_dam(),"DAMATEBA",410,5,410,550).
```

```

w_info(init):-
    bitmap(,_,fail(), "C:/istoriuli Zeglebi/SYSTEM/info_menu.bmp",0,0).

w_dam(init):-
    bitmap(,_,fail(), "C:/istoriuli Zeglebi/SYSTEM/menu DAMATEBA.bmp",0,0),
    menu(normal,_,_,help1(), "help"),
    edit(G_P1,_,edit_func(), "",200,10,200,25),
    edit(G_P2,_,edit_func(), "",200,60,200,25),
    edit(G_P3,_,edit_func(), "",200,110,200,25),
    edit(G_P4,_,edit_func(), "",200,160,200,25),
    edit(G_P5,_,edit_func(), "",200,210,200,25),
    edit(G_P6,_,edit_func(), "",200,260,200,25),
    edit(G_P7,_,edit_func(), "",200,310,200,25),
    edit(G_P8,_,edit_func(), "",200,360,200,25),
    edit(G_P9,_,edit_func(), "",200,410,200,25),
    button(,_,damat(), "SAVE",150,450,100,30).

help1(press):-
    window(,_,w_help1(), "help",100,10,510,550).

w_help1(init):-
    bitmap(,_,fail(), "C:/istoriuli Zeglebi/SYSTEM/help1.bmp",0,0).

damat(press):-
    A is get_text(G_P1),
    B is get_text(G_P2),
    C is get_text(G_P3),
    D is get_text(G_P4),
    E is get_text(G_P5),
    F is get_text(G_P6),
    G is get_text(G_P7),
    H is get_text(G_P8),
    I is "C:/istoriuli Zeglebi/PHOTO/" + get_text(G_P9) + ".bmp",
    assert_in(dzegli(A,B,C,D,E,F,G,H,I)),
    message("Congratulations !", "Tqven daamatet informacia monacemta bazaSi !", i),
    set_text("", G_P1),
    set_text("", G_P2),
    set_text("", G_P3),
    set_text("", G_P4),
    set_text("", G_P5),
    set_text("", G_P6),
    set_text("", G_P7),
    set_text("", G_P8),
    set_text("", G_P9).

w_dat(init):-
    bitmap(,_,fail(), "C:/istoriuli Zeglebi/SYSTEM/menu DATVALIEREBA.bmp",0,0),
    menu(pop_up,_,_,zeb(), "Search"),
    menu(normal,_,_,help(), "help"),
    button(,_,saxel_func(), "dasaxeleba",3,3,100,25),
    button(,_,tipi_func(), "tipi",3,100,70,25),

```

```

button(_,_celi_func(),"celi",3,160,70,25),
button(_,_sauk_func(),"saukune",3,260,100,25),
button(_,_qalaq_func(),"qalaqi",290,3,100,25),
button(_,_kutxe_func(),"kutxe",320,100,70,25),
button(_,_qvey_func(),"qvekana",320,160,70,25),
button(_,_damaars_func(),"damaarsebeli",290,260,100,25).

```

zeb(init):-

```

    menu(normal,_,_saxel_func(),"dasaxeleba"),
    menu(normal,_,_tipi_func(),"tipi"),
    menu(normal,_,_celi_func(),"celi"),
    menu(normal,_,_sauk_func(),"saukune"),
    menu(normal,_,_qalaq_func(),"qalaqi"),
    menu(normal,_,_kutxe_func(),"kutxe"),
    menu(normal,_,_qvey_func(),"qvekana"),
    menu(normal,_,_damaars_func(),"damaarsebeli").

```

help(press):-

```

window(_,_w_help(),"help",100,10,510,500).

```

w_help(init):-

```

    bitmap(_,_fail(),"C:/istoriuli Zeglebi/SYSTEM/help.bmp",0,0).

```

saxel_func(press):-

```

read(G_Saxeli,"dasaxeleba",s),
dzegli(G_Saxeli,A,B,C,D,E,F,G,G_P),
write("      "),write(G_Saxeli),nl,
write("_____"),nl,
write("      tipi      :"),write(A),nl,
write("      celi       :"),write(B),nl,
write("      saukune    :"),write(C),nl,
write("      qalaqi : "),write(D),nl,
write("      kutxe      :"),write(E),nl,
write("      qvekana    :"),write(F),nl,
write("      damaarsebeli: "),write(G),nl,nl,nl,
far_way(_),fail.

```

tipi_func(press):-

```

read(A,"tipi",s),
dzegli(G_Saxeli,A,B,C,D,E,F,G,G_P),
write("      "),write(G_Saxeli),nl,
write("_____"),nl,
write("      tipi      :"),write(A),nl,
write("      celi       :"),write(B),nl,
write("      saukune    :"),write(C),nl,
write("      qalaqi : "),write(D),nl,
write("      kutxe      :"),write(E),nl,
write("      qvekana    :"),write(F),nl,
write("      damaarsebeli: "),write(G),nl,nl,nl,
far_way(_),fail.

```

celi_func(press):-

```

read(B,"celi",s),
dzegli(G_Saxeli,A,B,C,D,E,F,G,G_P),
write("      "),write(G_Saxeli),nl,
write("-----"),nl,
write("      tipi      : "),write(A),nl,
write("      celi       : "),write(B),nl,
write("      saukune     : "),write(C),nl,
write("      qalaqi : "),write(D),nl,
write("      kutxe      : "),write(E),nl,
write("      qvekana     : "),write(F),nl,
write("      damaarsebeli: "),write(G),nl,nl,nl,
far_way().fail.

```

```

sauk_func(press):-
read(C,"saukune",s),
dzegli(G_Saxeli,A,B,C,D,E,F,G,G_P),
write("      "),write(G_Saxeli),nl,
write("-----"),nl,
write("      tipi      : "),write(A),nl,
write("      celi       : "),write(B),nl,
write("      saukune     : "),write(C),nl,
write("      qalaqi : "),write(D),nl,
write("      kutxe      : "),write(E),nl,
write("      qvekana     : "),write(F),nl,
write("      damaarsebeli: "),write(G),nl,nl,nl,
far_way().fail.

```

```

qalaq_func(press):-
read(D,"qalaqi",s),
dzegli(G_Saxeli,A,B,C,D,E,F,G,G_P),
write("      "),write(G_Saxeli),nl,
write("-----"),nl,
write("      tipi      : "),write(A),nl,
write("      celi       : "),write(B),nl,
write("      saukune     : "),write(C),nl,
write("      qalaqi : "),write(D),nl,
write("      kutxe      : "),write(E),nl,
write("      qvekana     : "),write(F),nl,
write("      damaarsebeli: "),write(G),nl,nl,nl,
far_way().fail.

```

```

kutxe_func(press):-
read(E,"kutxe",s),
dzegli(G_Saxeli,A,B,C,D,E,F,G,G_P),
write("      "),write(G_Saxeli),nl,
write("-----"),nl,
write("      tipi      : "),write(A),nl,
write("      celi       : "),write(B),nl,
write("      saukune     : "),write(C),nl,
write("      qalaqi : "),write(D),nl,
write("      kutxe      : "),write(E),nl,
write("      qvekana     : "),write(F),nl,

```

```
write("      damaarsebeli: "),write(G),nl,nl,nl,
far_way(),fail.
```

```
qvey_func(press):-
read(F,"qvekana",s),
dzegli(G_Saxeli,A,B,C,D,E,F,G,G_P),
write("      "),write(G_Saxeli),nl,
write("-----"),nl,
write("      tipi      : "),write(A),nl,
write("      celi      : "),write(B),nl,
write("      saukune     : "),write(C),nl,
write("      qalaqi : "),write(D),nl,
write("      kutxe      : "),write(E),nl,
write("      qvekana     : "),write(F),nl,
write("      damaarsebeli: "),write(G),nl,nl,nl,
far_way(),fail.
```

```
damaars_func(press):-
read(G,"qalaqi",s),
dzegli(G_Saxeli,A,B,C,D,E,F,G,G_P),
write("      "),write(G_Saxeli),nl,
write("-----"),nl,
write("      tipi      : "),write(A),nl,
write("      celi      : "),write(B),nl,
write("      saukune     : "),write(C),nl,
write("      qalaqi : "),write(D),nl,
write("      kutxe      : "),write(E),nl,
write("      qvekana     : "),write(F),nl,
write("      damaarsebeli: "),write(G),nl,nl,nl,
far_way(),fail.
```

```
far_way():-
window(,fan_func(),G_Saxeli,50,50,405,375).
```

```
fan_func(init):-
bitmap(,fail(),G_P,0,0).
```

% IIIIIIIIIII CODNIS BAZA IIIIIIIIIII

```
dzegli("alaverdi","monasteri","1010-1014","XI","","kaxeti","saqartvelo","","C:/istoriuli
Zeglebi/PHOTO/alaverdi.BMP").
dzegli("amagleba","tadzari","","XX","tbilisi","qartli","saqartvelo","","C:/istoriuli
Zeglebi/PHOTO/amaRleba.BMP").
dzegli("ananuri","cixesimagre","1660-
1676","XVII","","tianeti","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/ananuri.BMP").
dzegli("anchisxati","tadzari","","VI","tbilisi","qartli","saqartvelo","vaxtang
gorgasali","C:/istoriuli Zeglebi/PHOTO/anCisxati.BMP").
dzegli("antioqia","tadzari","","IV_V","mcxeta","qartli","saqartvelo","arCil
mefe","C:/istoriuli Zeglebi/PHOTO/antioqia.BMP").
dzegli("artanuji","cixesimagre","","V","","tao-klarjeti","turqeti","vaxtang
gorgasali","C:/istoriuli Zeglebi/PHOTO/artanuji.BMP").
```

dzepli("atenis sioni","tadzari","","VII","gori","qartli","saqartvelo","","C:/istoriuli
 Zeglebi/PHOTO/atenis sioni.BMP").
 dzepli("ackuri","cixesimagre","","IV","","","saqartvelo","","C:/istoriuli
 Zeglebi/PHOTO/awyuri.BMP").
 dzepli("axali shuamta","monasteri","1550-
 1565","XVI","telavi","kaxeti","saqartvelo","dedofali TinaTini","C:/istoriuli
 Zeglebi/PHOTO/axali SuamTa.BMP").
 dzepli("bagratis tadzari","tadzari","1001-
 1008","XI","qutaisi","imereti","saqartvelo","bagrat III","C:/istoriuli
 Zeglebi/PHOTO/bagratis taZari.BMP").
 dzepli("bana","tadzari","881-923","IX-X","","tao-klarjeti","turqeti","adarnese
 kurapalati","C:/istoriuli Zeglebi/PHOTO/bana.BMP").
 dzepli("bebris cixe","cixesimagre","","","tbilisi","qartli","saqartvelo","","C:/istoriuli
 Zeglebi/PHOTO/bebris cixe.BMP").
 dzepli("betania","monasteri","1185-1190","XII","tbilisi","qartli","saqartvelo","tamar
 mefe","C:/istoriuli Zeglebi/PHOTO/beTania.BMP").
 dzepli("bolnisis sioni","tadzari","478-
 493","V","bolnisi","qartli","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/bolnisis
 sioni.BMP").
 dzepli("dadianebis
 sasaxle","sasaxle","1849","XIX","zugdidi","samegrelo","saqartvelo","davit
 dadiani","C:/istoriuli Zeglebi/PHOTO/dadianebis sasaxle.BMP").
 dzepli("ertacminda","tadzari","","XIII","","","saqartvelo","","C:/istoriuli
 Zeglebi/PHOTO/erTawminda.BMP").
 dzepli("fitareti","monasteri","1213-1222","XIII","","","saqartvelo","lasha
 giorgi","C:/istoriuli Zeglebi/PHOTO/fitareTi.BMP").
 dzepli("goris cixe","cixesimagre","","","gori","qartli","saqartvelo","","C:/istoriuli
 Zeglebi/PHOTO/goris cixe.BMP").
 dzepli("geguTi","sasaxle","","","qutaisi","imereti","saqartvelo","","C:/istoriuli
 Zeglebi/PHOTO/geguTi.BMP").
 dzepli("gelati","monasteri","1106-1123","XII","qutaisi","imereti","saqartvelo","davit
 agmashenebeli","C:/istoriuli Zeglebi/PHOTO/gelaTi.BMP").
 dzepli("gergeti","tadzari","","IV","kazbegi","xevi","saqartvelo","davit
 agmashenebeli","C:/istoriuli Zeglebi/PHOTO/gergeTi.BMP").
 dzepli("gremi","cixesimagre","1565","XVI","","kaxeti","saqartvelo","","C:/istoriuli
 Zeglebi/PHOTO/gremi.BMP").
 dzepli("ierusalimis jvari","monasteri","","XI","ierusalimi","","israeli","","C:/istoriuli
 Zeglebi/PHOTO/ierusalimis jvari.BMP").
 dzepli("ishxani","tadzari","","IX","","tao-klarjeti","turqeti","","C:/istoriuli
 Zeglebi/PHOTO/iSxani.BMP").
 dzepli("ikalto","monasteri","","VII","telavi","kaxeti","saqartvelo","","C:/istoriuli
 Zeglebi/PHOTO/iyalTo.BMP").
 dzepli("jumati","monasteri","","","ozurqeti","guria","saqartvelo","","C:/istoriuli
 Zeglebi/PHOTO/jumaTi.BMP").
 dzepli("kaxtubani","tadzari","","VIII_IX","gurjaani","kaxeti","saqartvelo","","C:/istoriuli
 Zeglebi/PHOTO/kaxTubani.BMP").
 dzepli("kveteri","tadzari","","X","","","saqartvelo","","C:/istoriuli
 Zeglebi/PHOTO/kveteri.BMP").
 dzepli("mamadavit","tadzari","","VI","tbilisi","qartli","saqartvelo","","C:/istoriuli
 Zeglebi/PHOTO/mamadaviTi.BMP").
 dzepli("mcxetis jvari","monasteri","584-605","VI-
 VII","mcxeta","qartli","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/mcxetis jvari.BMP").

dzeqli("metexi","tadzari","1278-1289","XIII","tbilisi","qartli","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/metexi.BMP").
 dzeqli("mocameta","tadzari","","XI","qutaisi","imereti","saqartvelo","bagrat III","C:/istoriuli Zeglebi/PHOTO/mowameTa.BMP").
 dzeqli("narikala","cixesimagre","","","tbilisi","qartli","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/nariyala.BMP").
 dzeqli("nekresi","tadzari","","IV","","","","C:/istoriuli Zeglebi/PHOTO/nekresi.BMP").
 dzeqli("niko dadianis sasaxle","sasaxle","","XIX","zugdidi","samegrelo","saqartvelo","davit dadiani","C:/istoriuli Zeglebi/PHOTO/niko dadianis sasaxle.BMP").
 dzeqli("nikorcminda","tadzari","1014","XI","","racha","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/nikorwminda.BMP").
 dzeqli("ninocminda","tadzari","1550-1565","XVI","tbilisi","qartli","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/ninowminda.BMP").
 dzeqli("norio","tadzari","","","qartli","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/norio.BMP").
 dzeqli("oshki","tadzari","958-961","X","","tao-klarjeti","turqeti","adarnese kurapalati","C:/istoriuli Zeglebi/PHOTO/oSki.BMP").
 dzeqli("otxta","monasteri","","X","","tao-klarjeti","turqeti","davit kurapalati","C:/istoriuli Zeglebi/PHOTO/oTxTa.BMP").
 dzeqli("parxali","tadzari","","X","","tao-klarjeti","turqeti","davit kurapalati","C:/istoriuli Zeglebi/PHOTO/parxali.BMP").
 dzeqli("qashueti","tadzari","1753","XVIII","tbilisi","qartli","saqartvelo","givi amilaxvari","C:/istoriuli Zeglebi/PHOTO/qaSueTi.BMP").
 dzeqli("qurmuxa","tadzari","","","hereTi","azerbaijani","","C:/istoriuli Zeglebi/PHOTO/qurmuxa.BMP").
 dzeqli("qvataxevi","monasteri","1200-1220","XIII","","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/qvaTaxevi.BMP").
 dzeqli("rkoni","tadzari","","","qartli","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/rkoni.BMP").
 dzeqli("safara","tadzari","1268-1285","XIII","axalcixe","mesxeti","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/safara.BMP").
 dzeqli("sameba","tadzari","1989-2005","XX-XXI","tbilisi","qartli","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/sameba.BMP").
 dzeqli("samtavisi","tadzari","1030","XI","","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/samTavisi.BMP").
 dzeqli("samtavro","tadzari","1001-1010","XI","","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/samTavro.BMP").
 dzeqli("samcevrissi","tadzari","","VII","","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/samwevrissi.BMP").
 dzeqli("shatberdi","tadzari","836","IX","","tao-klarjeti","turqeti","aSot kurapalati","C:/istoriuli Zeglebi/PHOTO/Satberdi.BMP").
 dzeqli("shiomgvime","monasteri","","VI","","qartli","saqartvelo","","C:/istoriuli Zeglebi/PHOTO/SiomRvime.BMP").
 dzeqli("sioni","tadzari","","VI_VII","tbilisi","qartli","saqartvelo","adarnese kurapalati","C:/istoriuli Zeglebi/PHOTO/sioni.BMP").
 dzeqli("sno","cixesimagre","","","","C:/istoriuli Zeglebi/PHOTO/sno.BMP").
 dzeqli("stefancminda","tadzari","","IV-V","mcxeta","qartli","saqartvelo","archil mefe","C:/istoriuli Zeglebi/PHOTO/stefanwminda.BMP").

dzegli("sveticxoveli","tadzari","1010-1029","XI","mexeta","qartli","saqartvelo","", "C:/istoriuli Zeglebi/PHOTO/sveticxoveli.BMP").
 dzegli("sxvilo","cixesimagre","", "", "", "", "", "", "C:/istoriuli Zeglebi/PHOTO/sxvilo.BMP").
 dzegli("tmogvis cixe","cixesimagre","", "", "", "", "", "C:/istoriuli Zeglebi/PHOTO/Tmogvis cixe.BMP").
 dzegli("tbeti","tadzari","", "X","", "tao-klarjeti","turqeti","", "C:/istoriuli Zeglebi/PHOTO/tbeTi.BMP").
 dzegli("timotesubani","monasteri","", "X","borjomi","qartli","saqartvelo","", "C:/istoriuli Zeglebi/PHOTO/timoTesubani.BMP").
 dzegli("ubisi","tadzari","1141","XII","", "", "saqartvelo","", "C:/istoriuli Zeglebi/PHOTO/ubisi.BMP").
 dzegli("ufliscixe","tadzari","", "IX-X","gori","qartli","saqartvelo","", "C:/istoriuli Zeglebi/PHOTO/ufliscixe.BMP").
 dzegli("ujarma","cixesimagre","", "III","axmeta","kaxeti","saqartvelo","", "C:/istoriuli Zeglebi/PHOTO/ujarma.BMP").
 dzegli("vardzia","", "", "XII","aspinZa","mesxeti","saqartvelo","giorgi III","C:/istoriuli Zeglebi/PHOTO/varZia.BMP").
 dzegli("ciklani","tadzari","", "", "qartli","saqartvelo","", "C:/istoriuli Zeglebi/PHOTO/wilkani.BMP").
 dzegli("cugrugasheni","tadzari","1200-1220","XIII","", "", "saqartvelo","lasha giorgi","C:/istoriuli Zeglebi/PHOTO/wuRruRaSeni.BMP").
 dzegli("xandzta","monasteri","", "IX","", "tao-klarjeti","turqeti","grigol xandzeli","C:/istoriuli Zeglebi/PHOTO/xanZTa.BMP").
 dzegli("xaxuli","monasteri","", "X","", "tao-klarjeti","turqeti","davit kurapalati","C:/istoriuli Zeglebi/PHOTO/xaxuli.BMP").
 dzegli("xirsi","monasteri","", "VI","siRnaRi","kaxeti","saqartvelo","", "C:/istoriuli Zeglebi/PHOTO/xirsi.BMP").
 dzegli("kvelacminda","tadzari","", "VIII","gurjaani","kaxeti","saqartvelo","", "C:/istoriuli Zeglebi/PHOTO/yvelawminda.BMP").
 dzegli("zarzma","monasteri","1300-1320","XIV","", "mesxeti","saqartvelo","", "C:/istoriuli Zeglebi/PHOTO/zarzma.BMP").

პროგრამა 4. ქართველი მხატვრები

მოიყვანოთ პროგრამის აღწერა:

პროგრამა წარმოადგენს მონაცემთა ბაზას, რომელიც შეიცავს ინფორმაციას ქართველი მხატვრების ნახატებსზე. მასში შესაძლებელია სასურველი ნახატის ძიება სხვადასხვა პარამეტრის მიხედვით: მხატვარი(ნახელი და გვარი), ნახატი, წელი(ნახატის შექმნის), მასალა(თუ რომელი მასალითაა შესრულებული ნახატი), ზომა. ძიების შედეგია არჩეული ნახატის ან ნახატების მოკლე აღწერა, რომელსაც თან ერთვის ფოტოსურათი.

პროგრამა იწეობა შემდეგი მიზნით:

?-window(„w_main()„QARTVELI MXATVREBI",200,200,400,400).

wndow –არის პრედიკატი, რომელიც გამოიყენება ფანჯრის შესაქმნელად და მისი ზოგადი სახეა :

window(Handle,Parent,Win_func(),Title,X,Y,Width,Height).

Handle არის ფანჯრის სახელი, რომლის მეშვეობითაც შემდგომში შესაძლებელია ამ ფანჯარასთან მუშაობა. ძირითადად გამოიყენება გლობალური ცვლადები, მაგრამ ჩვენს შემთხვევაში იგი არაა საჭირო და ვიყენებთ ფიქტიურ ცვლადს („-“).

Parent არის ე.წ. “მშობელი” ფანჯრის სახელი. ჩვენ აქაც ვიყენებთ ფიქტიურ პარამეტრს.

Title არის ფანჯრის სათაური. ჩვენს შემთხვევაში „**QARTVELI MXATVREBI**”.

X,Y წარმოადგენენ ფანჯრის მარცხენა ზედა კუთხის კოორდინატებს. ხოლო **Width** და **Height** შესაბამისად ფანჯრის სიგრძე და სიმაღლე პიქსელებში.

Win_func() არის მოცემული ფანჯრის ფუნქციის სახელი. იგი შეიძლება იყოს ნებისმიერი, ჩვენს შემთხვევაში **w_main()**. მისი მეშვეობით შესაძლებელია ფანჯარაზე სხვა და სხვა სახის მანიპულაციების განხორციელება. ფუნქციას შეიძლება აქონდეს პარამეტრები, მაგრამ ჩვენ მათ არ ვიყენებთ და შესაბამისად ეუთოთებთ ფიქტიურ პარამეტრს. ქვემოთ მოყვანილია ამ ფუნქციის რეალიზაცია ჩვენს პროგრამაში:

w_main(Init):-

```
    bitmap(____,fail(),"C://qartveli mxatvreb/main menu.bmp",0,0),
    menu(pop_up____,way_out(),"File"),
    menu(pop_up____,s_func(),"Search By"),
    menu(normal____,info_func(),"About"),
    button(____,mxatvari(),"MXATVARI",10,10,100,30),
    button(____,naxati(),"NAXATI",10,50,100,30),
    button(____,celi(),"CELI",10,100,100,30),
    button(____,masala(),"MASALA",10,150,100,30),
    button(____,zoma(),"ZOMA",10,200,100,30),
    button(____,chamateba(),"CHAMATEBA",10,270,120,60).
```

ამ ფუნქციას ვიყენებთ ფანჯრის სხვა და სხვა კომპონენტების შესაქმნელად. ესენია: **bitmap**, **menu** და **button**. ფუნქცია ღებულობს პარამეტრს **init**, რომელიც ინიციალიზებას უკეთებს ფანჯრის ელემენტებს.

Bitmap–ის ანუ გამოსახულების განსათავსებლად ფანჯარაზე გამოიყენება პრედიკატი **bitmap**, რომლის ზოგადი სახეც შემდეგია:

bitmap(Handle,Parent,Bitmap_func(),Bitmap,X,Y).

მისი ზოგიერთი პარამეტი ემთხვევა **wndow** პრედიკატის პარამეტრებს: **Handle,Parent,X,Y**. მათ თავიდან არ განვიხილავთ.

Bitmap–პარამეტრი წარმოადგენს გამოსახულებიან ფაილამდე გზას. ჩვენს მაგალითში : "C://QARTVELI MXATVREBI/main menu.bmp". გახატვალისწინებელია, რომ პროლოგი მუშაობს მხოლოდ .BMP გაფართოების გამოსახულებიან ფაილებთან.

Bitmap_func() – პარამეტრი წარმოადგენს ფუნქციას, რომლის მეშვეობითაც შესაძლებელია მანიპულაციების განხორციელება გამოსახულებაზე.

ფანჯრის მენიუს შესაქმნელად გამოიყენება პრედიკატი **menu** :

menu(Type,Handle,Parent,Menu_func(),Text),

Handle Parent პარამეტრები ზემოთ განხილულის ანალოგიურია.

Type განსაზღვრავს მენიუს ტიპს. სულ, პროლოგში გამოიყენება ორი სახის მენიუ ესენია : **pop_up** და **normal**. პირველი მათგანი გამოიყენება ქვემენიუების, ხოლო მეორე კი (ჯალკეული მენიუების შესაქმნელად. **Text** პარამეტრი განსაზღვრავს მენიუს სახელს, მაგალითად ჩვენს მიერ შერჩეულია შემდეგი სახელები : "File", "Search By" და "About".

ჩვენს შემთხვევაში ფუნქცია **Menu_func** წარმოდგენილია **way_out()**, **s_func()** და **info_func()** ფუნქციების სახით. ისინი გამოიყენება იმისათვის რომ ყოველ მენიუს მიუანაბლო შესაბამისი ფუნქცია. მათი დაწვრილებითი განმარტება მოცემულია მოგვიანებით.

button(Handle,Parent,Button_func(),Title,X,Y,Width,Height) – პრედიკატი გამოიყენება ფანჯარაში ღილაკის შესაქმნელად. მისი პარამეტრები უკვე აღწერილია. ჩვენს პროგრამაში იქმნება ექვსი ასეთი ღილაკი "MXATVARI", "NAXATI", "CELI", "MASALA", "ZOMA", "CHAMATEBA".

განვიხილოთ ის ფუნქციები, რომლებიც გამოიყენებოდა აღნიშნულ პრედიკატებში.

w_main(close):-

not(yes_no("Exit","Do you want to Exit ?",?)).

მოცემული წესი გამოიყენება იმისათვის, რომ შემთხვევით არ მოხდეს პროგრამიდან გამოსვლა, ანუ, თუკი თქვენ შემთხვევით დააჭირეთ პროგრამიდან გამოსვლის ღილაკს, პროგრამა არ გაითიშება, არამედ შეგვეითებთ: ნამდვილად გსურთ თუ არა ამის გაკეთება ? ეს მიიღწევა **yes_no** პრედიკატის გამოყენებით.

yes_no(Title,Text,Icon).

Title – არის დიალოგური ფანჯრის სათაური, ხოლო **Text** კი – მისი ტექსტი. ჩვენს შემთხვევაში, შესაბამისად "Exit" და "Do you want to Exit ?".

Icon პარამეტრის ადგილას შესაძლებელია რამოდენიმე ვარიანტის ჩასმა. ესენია : ? – კითხვა, s – STOP, i –ინფორმაცია, n – (კარგი) შეტყობინება.

way_out(init):-

menu(normal,_,menu_exit(),"Exit").

menu_exit(press):-close_window().

way_out ფუნქცია გამოიყენება File მენიუს Exit ქვემენიუს შესაქმნელად. თავის მხრივ Exit მენიუს ფუნქცია ხურავს ძირითად ფანჯარას, რადგანაც მის პარამეტრად გამოიყენებულია ფიქტიური პარამეტრი, რაც გულისხმობს ძირითად ფანჯარას.

```
s_func(init):-  
menu(normal,_,mxtatvari(), "MXATVARI"),  
menu(normal,_,naxati(), "NAXATI"),  
menu(normal,_,celi(), "CELI"),  
menu(normal,_,masala(), "MASALA"),  
menu(normal,_,zoma(), "ZOMA").
```

ეს ფუნქცია ინიციალიზებას უკეთებს ხუთ ქვემენიუს: "MXATVARI", "NAXATI", "CELI", "MASALA", "ZOMA", რომელთაც თავის მხრივ გააჩნიათ თავიანთი ფუნქციები: mxtatvari(), naxati(), celi(), masala(), zoma().

```
Info_func(press):-window(,w_info(), "About", 100, 100, 492, 400).
```

ამ ფუნქციას გამოაქვს ფანჯარა სახელად "About", რომელსაც ინიციალიზაციას უკეთებს ფუნქცია :

```
w_info(Init):-  
    bitmap(,fail(), "C://qartveli mxtatvreb/cham menu.bmp", 0, 0).
```

```
chamateba(press):-
```

```
    window(, cham_main(), "CHAMATEBA", 250, 250, 400, 400)
```

ამ ფუნქციას გამოაქვს ფანჯარა სახელად "CHAMATEBA", რომელსაც ინიციალიზაციას უკეთებს შემდეგი ფუნქცია:

```
cham_main(Init):-
```

```
    bitmap(,fail(), "C://qartveli mxtatvreb/cham menu.bmp", 0, 0),  
    edit(G_P1, edit_func(), "", 150, 10, 200, 25),  
    edit(G_P2, edit_func(), "", 150, 60, 200, 25),  
    edit(G_P3, edit_func(), "", 150, 110, 200, 25),  
    edit(G_P4, edit_func(), "", 150, 160, 200, 25),  
    edit(G_P5, edit_func(), "", 150, 210, 200, 25),  
    edit(G_P6, edit_func(), "", 150, 260, 200, 25),  
    button(, save_func(), "SAVE", 100, 300, 100, 30).
```

ამ ფანჯარაზე განთავსებულია ერთი ღილაკი და ტექსტის შესაყვანი ექვსი ხპეციალური უბანი. რომელიც იქმნება edit პრედიკატის მიერ. ამ პრედიკატის პარამეტრები ოდენტურია window პრედიკატის პარამეტრებისა, რომელიც უკვე განვიხილო.

edit(Handle,Parent,Edit_func(),Title,X,Y,Width,Height).

ყოველ ასეთ უბანს ეანიჭებთ შესაბამის გლობალურ ცვლადს G_P1-G_P6, რადგანაც მომავალში აუცილებლად დაგეჭირდება.

განვიხილოთ "SAVE" ღილაკის ფუნქცია, რომელიც საშუალებას გეძლევს პროგრამის ცოდნის ბაზაში დაეამატოთ ახალი მონაცემები :

```
save_func(press):-  
A is get_text(G_P1),  
B is get_text(G_P2),  
C is get_text(G_P3),  
D is get_text(G_P4),  
E is get_text(G_P5),  
F is "C://qartveli mxatvreb/"+ get_text(G_P6)+ ".bmp",  
assert_in(baza(A,B,C,D,E,F)),  
message("Info !","INFORMACIA SHENAXULIA !",i),  
set_text("",G_P1),  
set_text("",G_P2),  
set_text("",G_P3),  
set_text("",G_P4),  
set_text("",G_P5),  
set_text("",G_P6).
```

გლობალური ცვლადების მეშეობით, რომლებიც თავიდან მივანიჭეთ edit კომპონენტებს და set_text ფუნქციის გამოყენებით A,B,C,D,E,F ცვლადებს ენიჭებათ შესაბამის edit კომპონენტში ჩაწერილი ტექსტი. ამის შემდეგ ეს ახალი მონაცემები ბაზაში ჩაწერება assert_in ფუნქციის მეშეობით. მოცემული ოპერაციების წარმატების შემთხვევაში გამოდის დიალოგური ფანჯარა (message), რომელიც გეატყობინებს, რომ მოხდა ცოდნის ბაზის შექცება.

set_text ფუნქციით ხორციელდება edit კომპონენტის გასუფთავება, ცარიელი ტექსტის (") შეყვანით.

"Search By" მერიუში იქმნება 5 უბანი, რომელთა მეშეობითაც შესაძლებელია სასურველი ნახატის ძებნა სხვადასხვა პარამეტრებით: "mxatvari", "naxati", "celi", "masala", "zoma". ყოველი პარამეტრისათვის რეალიზებულია ცალკე ფუნქცია, რომელთა მუშაობის პრინციპიც ერთნაირია, ამიტომ განვიხილავთ მხოლოდ ერთს :

mxatvari(press):-

```

read(A,"mxatvari",s),
baza(A,G_N,B,C,D,G_P),
write("      "),write(G_N),nl,
write("*****"),nl,
write("    MXATVARI : "),write(A),nl,
write("    CELI : "),write(B),nl,
write("    MASALA: "),write(C),nl,
write("    ZOMA: "),write(D),nl,nl,
far_way(),fail.

```

მოცემული ფუნქციის პარამეტრია **press**, რაც იმაზე მიუთითებს, რომ იგი აქტიურდება შესაბამის ლიდაქზე თუ შენიუზე დაჭერის შედეგად. ფუნქციის ამუშავებისთანავე გამოდის სტანდარტული ფანჯარა ინფორმაციის შეყვანისათვის. მასში შეტანილი ინფორმაცია თავსდება **A** ცვლადში.

```

read(A,"mxatvari",s),

```

read პრედიკატის მეორე პარამეტრი წარმოადგენს **read** ფანჯრის სათაურს, ხოლო მესამე პარამეტრის ადგილას შეიძლება იყოს ორი სიმბოლოდან ერთ-ერთი : **s** ან **f**. **s** აღნიშნავს ტექსტს და ამიტომ **read**-ში ჩვენს მიერ შეტანილ ინფორმაციას არ სჭირდება ბრჭყალები. შემდეგ მოდის მიზანი : **baza(A,G_N,B,C,D,G_P)**, რომელშიც გამოიყენება ორი გლობალური ცვლადი : **G_N** ნახატი(სახელი) და **G_P** – სრული გზა შესაბამისი ნახატის ფოტო ფაილამდე. ეს ცვლადები გამოიყენება ფუნქციაში – **far_way()**. თავიდან **baza** პრედიკატის მხოლოდ **A** პარამეტრია განსაზღვრული, შემდეგ ხორციელდება შესაბამისი ფაქტის ძებნა ცოდნის ბაზაში, მას შემდეგ, რაც მოიძებნება შესაბამისი სახელის მქონე ნახატი, დანარჩენ ცვლადებსაც მიენიჭებათ მნიშვნელობები. შემდეგ ეს მნიშვნელობები გამოგეაქვს ეკრანზე **write** ფუნქციების მეშვეობით. შემდეგ საქმეში ერთვება ფუნქცია **far_way()**, რომელსაც ეკრანზე გამოაქვს შესაბამისი ნახატის ფოტოსურათი, ამისათვის იგი იყენებს ზემოთაღნიშნულ ორ გლობალურ ცვლადს:

```

far_way():-
window(,fan_func(),G_N,50,405,375).

fan_func(init):-
bitmap(,fail(),G_P,0,0).

```

.....

ამოცანის პროლოგ-პროგრამის სრული ტექსტი:

```

?-window(,w_main(),"QARTVELI MXATVREBI",200,200,400,400).

```

```

w_main(init):-

```

```

        bitmap(_,_ ,fail(),"C://qartveli mxatvrebi/main menu.bmp",0,0),
menu(pop_up,_ ,way_out(),"File"),
menu(pop_up,_ ,s_func(),"Search By"),
menu(normal,_ ,info_func(),"About"),
        button(_ ,mxatvari(),"MXATVARI",10,10,100,30),
        button(_ ,naxati(),"NAXATI",10,50,100,30),
        button(_ ,celi(),"CELI",10,100,100,30),
        button(_ ,masala(),"MASALA",10,150,100,30),
        button(_ ,zoma(),"ZOMA",10,200,100,30),
        button(_ ,chamateba(),"CHAMATEBA",10,270,120,60).

```

```

w_main(close):-
not(yes_no("Exit","Do you want to Exit ?",?)).

```

```

way_out(init):-
menu(normal,_ ,menu_exit(),"Exit").

```

```

menu_exit(press):-
close_window().

```

```

s_func(init):-
menu(normal,_ ,mxatvari(),"MXATVARI"),
menu(normal,_ ,naxati(),"NAXATI"),
menu(normal,_ ,celi(),"CELI"),
menu(normal,_ ,masala(),"MASALA"),
menu(normal,_ ,zoma(),"ZOMA").

```

```

info_func(press):-
window(_ ,w_info(),"About",100,100,492,400).

```

```

w_info(init):-
bitmap(_ ,fail(),"C://qartveli mxatvrebi/about.bmp",0,0).

```

```

chamateba(press):-
window(_ ,cham_main(),"CHAMATEBA",250,250,400,400).

```

```

cham_main(init):-
        bitmap(_ ,fail(),"C://qartveli mxatvrebi/cham menu.bmp",0,0),
edit(G_P1,_ ,edit_func(),"",150,10,200,25),
edit(G_P2,_ ,edit_func(),"",150,60,200,25),
edit(G_P3,_ ,edit_func(),"",150,110,200,25),
edit(G_P4,_ ,edit_func(),"",150,160,200,25),
edit(G_P5,_ ,edit_func(),"",150,210,200,25),
edit(G_P6,_ ,edit_func(),"",150,260,200,25),
        button(_ ,save_func(),"SAVE",100,300,100,30).

```

```

save_func(press):-
A is get_text(G_P1),
B is get_text(G_P2),
C is get_text(G_P3),
D is get_text(G_P4),

```

```

E is get_text(G_P5),
F is "C://qartveli mxatvreb/."+ get_text(G_P6)+ ".bmp",
assert_in(baza(A,B,C,D,E,F)),
message("Info !", "INFORMACIA SHENAXULIA !",i),
set_text("",G_P1),
set_text("",G_P2),
set_text("",G_P3),
set_text("",G_P4),
set_text("",G_P5),
set_text("",G_P6).

```

```

mxatvari(press):-
read(A,"mxatvari",s),
baza(A,G_N,B,C,D,G_P),
write("      "),write(G_N),nl,
write("*****"),nl,
write("      MXATVARI : "),write(A),nl,
write("      CELI : "),write(B),nl,
write("      MASALA: "),write(C),nl,
write("      ZOMA: "),write(D),nl,nl,nl,
far_way(),fail.

```

```

naxati(press):-
read(G_N,"naxati",s),
baza(A,G_N,B,C,D,G_P),
write("      "),write(G_N),nl,
write("*****"),nl,
write("      MXATVARI : "),write(A),nl,
write("      CELI : "),write(B),nl,
write("      MASALA: "),write(C),nl,
write("      ZOMA: "),write(D),nl,nl,nl,
far_way(),fail.

```

```

celi(press):-
read(B,"celi",s),
baza(A,G_N,B,C,D,G_P),
write("      "),write(G_N),nl,
write("*****"),nl,
write("      MXATVARI : "),write(A),nl,
write("      CELI : "),write(B),nl,
write("      MASALA: "),write(C),nl,
write("      ZOMA: "),write(D),nl,nl,nl,
far_way(),fail.

```

```

masala(press):-
read(C,"masala",s),
baza(A,G_N,B,C,D,G_P),
write("      "),write(G_N),nl,
write("*****"),nl,
write("      MXATVARI : "),write(A),nl,
write("      CELI : "),write(B),nl,
write("      MASALA: "),write(C),nl,

```

```
write("      ZOMA: "),write(D),nl,nl,nl,
far_way().fail.
```

```
zoma(press):-
read(D,"zoma",s),
baza(A,G_N,B,C,D,G_P),
write("      "),write(G_N),nl,
write("*****"),nl,
write("      MXATVARI : "),write(A),nl,
write("      CELI : "),write(B),nl,
write("      MASALA: "),write(C),nl,
write("      ZOMA: "),write(D),nl,nl,nl,
far_way().fail.
```

```
far_way():-
window(,_,fan_func(),G_N,50,50,405,375).
```

```
fan_func(init):-
    bitmap(,_,fail(),G_P,0,0).
```

```
baza("firosmani","meczove","19??","http://www.baiagallery.ge/art.html","100x100","C:/qartveli mxatvrebi/meczove.bmp").
baza("firosmani","aktrisa margarita","19??","http://www.baiagallery.ge/art.html","90x100","C:/qartveli mxatvrebi/aktrisa margarita.bmp").
baza("Kirill Zdanevich","Mountains","1960","Gouache on paper","35X67","C:/qartveli mxatvrebi/kza.bmp").
baza("Kirill Zdanevich","Mountains","1960","Gouache on paper","72X80","C:/qartveli mxatvrebi/kzb.bmp").
baza("Kirill Zdanevich","Paris","1962","Gouache on paper","50X45","C:/qartveli mxatvrebi/kzc.bmp").
baza("Kirill Zdanevich","Still Life","","Gouache on paper","56X42","C:/qartveli mxatvrebi/kzd.bmp").
baza("Avto Varazi","Sleeping Woman","1972","Mixed technique on cardboard","86X52","C:/qartveli mxatvrebi/ata.bmp").
baza("Avto Varazi","Violin","1975","Mixed technique on cardboard","75X53","C:/qartveli mxatvrebi/atb.bmp").
baza("Korneli Sanadze","Tea Plantation","1968","Oil on paper","40X65","C:/qartveli mxatvrebi/ksa.bmp").
baza("Korneli Sanadze","Achara","1966","oil on canvas","40,5X52","C:/qartveli mxatvrebi/ksb.bmp").
baza("Korneli Sanadze","Landscape","1960","Oil on paper","44X55","C:/qartveli mxatvrebi/ksc.bmp").
baza("Korneli Sanadze","Kobuleti","1950","Oil on cardboard","37X63","C:/qartveli mxatvrebi/ksd.bmp").
baza("Korneli Sanadze","The Tree","1960","Oil on cardboard","48X59","C:/qartveli mxatvrebi/kse.bmp").
baza("Valentin Sherpilov","Leselidze. Sea Side","1943","oil on canvas","100X150","C:/qartveli mxatvrebi/vsa.bmp").
baza("Vasili Shukhaev","Natanebi","1950","tempera on canvas","93X73","C:/qartveli mxatvrebi/vsaa.bmp").
```

baza("Vasili Shukhaev", "Landscape", "1951", "tempera on canvas", "67X83", "C:/qartveli mxatvrebi/vsb.bmp").
 baza("Helene Akhvediani", "Kvishkheti Way", "1961", "pastel on paper", "50X115", "C:/qartveli mxatvrebi/haa.bmp").
 baza("Sergo Kobuladze", "Legend of Tariel", "1937", "mixed technique on paper", "", "C:/qartveli mxatvrebi/ska.bmp").
 baza("Sergo Kobuladze", "Woman with Grapes", "1962", "tempera on cardboard", "70X90", "C:/qartveli mxatvrebi/skb.bmp").
 baza("Korneli Sanadze", "The Wood", "1960", "Oil on paper", "47X51", "C:/qartveli mxatvrebi/ksf.bmp").
 baza("Vano Meliashvili", "Queen Tamara", "1950", "oil on cardboard", "70X52", "C:/qartveli mxatvrebi/vma.bmp").
 baza("Helene Akhvediani", "Vachevi", "19??", "Oil on canvas", "50X70", "C:/qartveli mxatvrebi/hab.bmp").
 baza("Helene Akhvediani", "Landscape", "", "pencil on paper", "", "C:/qartveli mxatvrebi/hac.bmp").
 baza("Irakli Pardjiani", "Argonauts", "1984", "mixed technique on paper", "120X150", "C:/qartveli mxatvrebi/ipa.bmp").
 baza("Irakli Pardjiani", "Annunciation", "1989", "sanguine on paper", "30X41", "C:/qartveli mxatvrebi/ipb.bmp").
 baza("Lado Gudishvili", "Abesalom and Eteri", "1940", "Oil on canvas", "133X105", "C:/qartveli mxatvrebi/lga.bmp").
 baza("Sergo Kobuladze", "King Lire", "1946", "gouache on paper", "27X18", "C:/qartveli mxatvrebi/skc.bmp").
 baza("Sergo Kobuladze", "Taming a horse", "1928", "Indian ink on paper", "25X21", "C:/qartveli mxatvrebi/skd.bmp").
 baza("Sergo Kobuladze", "King Lire", "1946", "gouache on paper", "26X22", "C:/qartveli mxatvrebi/ske.bmp").
 baza("Sergo Kobuladze", "Dante", "1943", "pencil on paper", "26X16", "C:/qartveli mxatvrebi/skf.bmp").
 baza("Helene Akhvediani", "Winter", "1966", "Gouache on paper", "18X18", "C:/qartveli mxatvrebi/hae.bmp").
 baza("Ketevan Maghalashvili", "Still Life", "1959", "Oil on canvas", "38X46", "C:/qartveli mxatvrebi/kma.bmp").
 baza("Peter Bliotkin", "Nude", "1970", "Gouache on paper", "18X26", "C:/qartveli mxatvrebi/pba.bmp").
 baza("Peter Bliotkin", "Nude", "1970", "Oil on cardboard", "51X40", "C:/qartveli mxatvrebi/pbb.bmp").
 baza("Peter Bliotkin", "Maternity", "1973", "Oil on cardboard", "46X40", "C:/qartveli mxatvrebi/pbc.bmp").
 baza("Peter Bliotkin", "Abstraction", "1949", "Gouache on paper", "32X27", "C:/qartveli mxatvrebi/pbe.bmp").
 baza("Peter Bliotkin", "Tbilisi", "1962", "Gouache on paper", "44X48", "C:/qartveli mxatvrebi/pbf.bmp").
 baza("Peter Bliotkin", "Abstraction", "1950", "Gouache on paper", "32X27", "C:/qartveli mxatvrebi/pbg.bmp").
 baza("Peter Bliotkin", "Abstraction", "1950", "Gouache on paper", "30X42", "C:/qartveli mxatvrebi/pbh.bmp").
 baza("Peter Bliotkin", "Maternity", "1974", "Oil on cardboard", "50X40", "C:/qartveli mxatvrebi/pbl.bmp").
 baza("Mose Toidze", "Oriental Motif", "1910", "Oil on cardboard", "21X29", "C:/qartveli mxatvrebi/mta.bmp").

baza("Gigo Gabashvili", "Still Life with Fruits", "beginning of XX century", "oil on canvas", "46X70", "C:/qartveli mxatvrebi/gga.bmp").

baza("Lado Gudishvili", "Singing Bird", "1961", "Gouache on paper", "26X19", "C:/qartveli mxatvrebi/lgb.bmp").

baza("David Kakabadze", "Ninoshvili's Guria", "1934", "Watercolour on paper", "15X24", "C:/qartveli mxatvrebi/dka.bmp").

baza("Ioseb Gabashvili", "Woman with the Apples", "1930", "Gouache on p", "", "C:/qartveli mxatvrebi/iga.bmp").

baza("Ioseb Gabashvili", "Woman Worn in Red Dress", "1930", "Gouache on paper", "", "C:/qartveli mxatvrebi/igc.bmp").

baza("Ioseb Gabashvili", "Woman with the Basket of Apples", "1930", "Gouache on paper", "", "C:/qartveli mxatvrebi/igc.bmp").

baza("Sergo Kobuladze", "Georgian Nobleman", "1934", "Mixed media on paper", "31.5X21", "C:/qartveli mxatvrebi/skh.bmp").

baza("Irakli Parjiani", "", "1986", "Oil on canvas", "130X160", "C:/qartveli mxatvrebi/ipc.bmp").

baza("Alexsander Bandzeladze", "Abstraction", "1982", "Gouache on paper", "65X90", "C://qartveli mxatvrebi/aba.bmp").

baza("Helene Akhvdiani", "Landscape", "1940", "Oil on canvas", "75X100", "C://qartveli mxatvrebi/haf.bmp").

baza("Lado gudiashvili", "On a balcony", "1956", "Gouache on paper", "36X25", "C://qartveli mxatvrebi/lgc.bmp").

baza("Helene Akhvdiani", "Winter", "1973", "Oil on canvas", "92X66", "C://qartveli mxatvrebi/hag.bmp").

baza("Lado gudiashvili", "Kinto", "1925", "Gouache on paper", "30X20", "C://qartveli mxatvrebi/lgb.bmp").

baza("Lado gudiashvili", "Bruniquell", "1924", "Watercolor on paper", "30X20", "C://qartveli mxatvrebi/lge.bmp").

baza("Helene Akhvdiani", "Landscape", "1970", "Oil on canvas", "100X80", "C://qartveli mxatvrebi/haaa.bmp").

ამოცანა 5. მობილური ტელეფონების მონაცემები

მოუყვანოთ ამოცანის სრული ტექსტი:

```
% **MOBILURI TELEFONEBIS MONACEMTA BAZA**
```

```
?-window(,win_func(),"Mobiles",320,120,400,500).
```

```
win_func(init):-
```

```
bitmap(,fail(),"c://mobiles//mobile/menu.bmp",30,10),
```

```
menu(pop_up,_,way_out(),"File"),
```

```
menu(normal,_,_,menu_brush(),"&Brush"),
```

```
menu(pop_up,_,search(),"Mobile"),
```

```
menu(normal,_,_,menu_window(),"&New window"),
```

```
menu(pop_up,_,_,info(),"Help"),
```

```

edit(G_P1,_edit_func(),"",120,10,200,25),
edit(G_P2,_edit_func(),"",120,60,200,25),
edit(G_P3,_edit_func(),"",120,110,200,25),
edit(G_P4,_edit_func(),"",120,160,200,25),
edit(G_P5,_edit_func(),"",120,210,200,25),
edit(G_P6,_edit_func(),"",120,270,200,25),
edit(G_P7,_edit_func(),"",120,330,200,25),

static(_,_st_func(),"Name",10,15,38,15),
static(_,_st_func(),"Model",10,60,43,15),
static(_,_st_func(),"Color",10,115,35,15),
static(_,_st_func(),"Neto",10,165,35,15),
static(_,_st_func(),"Memory",10,215,55,18),
static(_,_st_func(),"Kamera",10,275,55,15),
static(_,_st_func(),"Path to the file",10,335,94,15),

button(_,_bfunc(),"Save New",173,410,80,20).

```

win_func(init):-

```

    bitmap(_,_fail(), default, 50, 50),
    bitmap(_,_fail(), default, 250, 50),
    bitmap(_,_fail(), default, 100, 100),
    bitmap(_,_fail(), default, 200, 100),
    bitmap(_,_fail(), default, 50, 150),
    bitmap(_,_fail(), default, 150, 150),
    bitmap(_,_fail(), default, 250, 150),
    bitmap(_,_fail(), default, 100, 200),
    bitmap(_,_fail(), default, 200, 200),
    bitmap(_,_fail(), default, 150, 250),

```

```

    icon(_,_fail(), default, 165, 215),
    icon(_,_fail(), default, 165, 115),
    icon(_,_fail(), default, 110,70),
    icon(_,_fail(), default, 220, 70),
    icon(_,_fail(), default, 30,112),
    icon(_,_fail(), default, 300,110),
    icon(_,_fail(), default, 112,170),
    icon(_,_fail(), default, 70,115),
    icon(_,_fail(), default, 265,115),
    icon(_,_fail(), default, 210,170),
    icon(_,_fail(), default, 23,76),
    icon(_,_fail(), default, 305,72),

```

G_Br=0.

menu_brush(press) :-

```

    new_brush(G_Br),
    update_window(),
    G_Br:= (G_Br+1) mod 10.

```

new_brush(0) :- window_brush(_ ,rgb(0, 0, 0), x).

new_brush(1) :- window_brush(_ ,rgb(255, 0, 255), ()).

```

new_brush(2) :- window_brush(, "\\Wall.bmp").
new_brush(3) :- window_brush(, "Nature_14.bmp").
new_brush(4) :- window_brush(, "Berry.bmp").
new_brush(5) :- window_brush(, "2686.bmp").
new_brush(6) :- window_brush(, "paris.bmp").
new_brush(7) :- window_brush(, rgb(127, 0, 127)).
new_brush(8) :- window_brush(, "2234.bmp").
new_brush(9) :- window_brush(, "2218.bmp").
new_brush(10) :- window_brush(, ).

menu_window(press) :-
    position(,X, Y),
    window(, 0, win_func(, "Window", X+40, Y+30, 400, 500).

win_func(close):-
    not(yes_no("Exit", "Do you want to exit?",?)).

way_out(init):-
    menu(normal,_,menu_exit(, "Exit").

menu_exit(press):-
    close_window(, ).

info(init):-
    menu(normal,_,creator(, "About"),
    menu(normal,_,internet(, "Information").

creator(press):-
    message(" Author", "This Software was built by Natia Phirkulashvili 4 June 2005",i).

internet(press):-
    message("Web Sites", "For more detailed information please visit
    the following sites:
        http://www.nokia.com
        http://www.siemens.com
        http://www.panasonic.com
        http://www.motorola.com
        http://www.sony_ericsson
        , http://www.samsung.com
        http://www.lg.com",!).

edit_func(init):-
    color_text(,rgb(200,24,10)),
    color_text_back(,rgb(255,230,123)).

st_func(init):-
    color_text(,rgb(255,224,100)).
    color_text_back(,rgb(25,120,255)).

search(init):-
    menu(normal,_,t1(, "Name"),
    menu(normal,_,t2(, "Model"),

```

```

menu(normal,_,t3(),"Color"),
menu(normal,_,t4(),"Neto"),
menu(normal,_,t5(),"Memory"),
menu(normal,_,t6(),"Kamera").

```

t1(press):-

```

read(X,"Name of mobile",s),
write("Name"),write(" "),
write("Model"),write(" "),
write("Color"),write(" "),
write("Neto"),write(" "),
write("Memory"),write(" "),
write("Kamera"),nl,nl,
mobile(X,G_Y,Z,A,B,C,G_P),
write(X),write(" "),
write(G_Y),write(" "),
write(Z),write(" "),
write(A),write(" "),
write(B),write(" "),
write(C),nl,nl,
far_way(),fail.

```

t2(press):-

```

read(G_Y,"Model of mobile",s),
write("Name"),write(" "),
write("Model"),write(" "),
write("Color"),write(" "),
write("Neto"),write(" "),
write("Memory"),write(" "),
write("Kamera"),nl,nl,
mobile(X,G_Y,Z,A,B,C,G_P),
write(X),write(" "),
write(G_Y),write(" "),
write(Z),write(" "),
write(A),write(" "),
write(B),write(" "),
write(C),nl,nl,
far_way(),fail.

```

t3(press):-

```

read(Z,"Color of mobile",s),
write("Name"),write(" "),
write("Model"),write(" "),
write("Color"),write(" "),
write("Neto"),write(" "),
write("Memory"),write(" "),
write("Kamera"),nl,nl,
mobile(X,G_Y,Z,A,B,C,G_P),
write(X),write(" "),
write(G_Y),write(" "),
write(Z),write(" "),
write(A),write(" "),

```

```

write(B),write("  "),
write(C),nl,nl,
far_way(),fail.

```

t4(press):-

```

read(A,"Neto of mobile",s),
write("Name"),write("  "),
write("Model"),write("  "),
write("Color"),write("  "),
write("Neto"),write("  "),
write("Memory"),write("  "),
write("Kamera"),nl,nl,
mobile(X,G_Y,Z,A,B,C,G_P),
write(X),write("  "),
write(G_Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),
write(C),nl,nl,
far_way(),fail.

```

t5(press):-

```

read(B,"Memory of mobile ",s),
write("Name"),write("  "),
write("Model"),write("  "),
write("Color"),write("  "),
write("Neto"),write("  "),
write("Memory"),write("  "),
write("Kamera"),nl,nl,
mobile(X,G_Y,Z,A,B,C,G_P),
write(X),write("  "),
write(G_Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),
write(C),nl,nl,
far_way(),fail.

```

t6(press):-

```

read(C,"Kamera of mobile",n),
write("Name"),write("  "),
write("Model"),write("  "),
write("Color"),write("  "),
write("Neto"),write("  "),
write("Memory"),write("  "),
write("Kamera"),nl,nl,
mobile(X,G_Y,Z,A,B,C,G_P),
write(X),write("  "),nl,
write(G_Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),

```

```
write(C),nl,nl,  
far_way(),fail.
```

```
far_way():-  
    window(,wfunc(),G_Y,40,170,400,500).
```

```
wfunc(init):-  
    window_brush(,rgb(255,255,255)),  
    bitmap(,fail(),G_P,30,10).
```

```
bfunc(press):-  
    X is get_text(G_P1),  
    Y is get_text(G_P2),  
    Z is get_text(G_P3),  
    A is get_text(G_P4),  
    B is get_text(G_P5),  
    C is get_text(G_P6),  
    D is get_text(G_P7),  
    assert_in(mobile(X,Y,Z,A,B,C,D)),  
    message("Congratulations!", "You have successfully refilled database!",!),  
    set_text(,G_P1),  
    set_text(,G_P2),  
    set_text(,G_P3),  
    set_text(,G_P4),  
    set_text(,G_P5),  
    set_text(,G_P6),  
    set_text(,G_P7).
```

```
mobile("Nokia","E50","silver","89","8","no","c://mobiles/e50.bmp").  
mobile("Nokia","E61","silver","100","10","1.2","c://mobiles/pm_nokia_e61.bmp").  
mobile("Nokia","E70","silver","110","12","1.4","c://mobiles/pm_nokia_e70.bmp").  
mobile("Nokia","N70","silver","90","10","1.3","c://mobiles/pm_n70.bmp").  
mobile("Nokia","N71","silver","83","11","1.2","c://mobiles/n71.bmp").  
mobile("Nokia","N72","black","80","11","1.2","c://mobiles/n72_pm.bmp").  
mobile("Nokia","N73","silver","103","12","1.3","c://mobiles/n73_pm.bmp").  
mobile("Nokia","N80","silver","85","11","1.2","c://mobiles/n80.bmp").  
mobile("Nokia","7210","blue","89","8","no","c://mobiles/pm_7210.bmp").  
mobile("Nokia","7600","yellow","100","12","1.2","c://mobiles/pm_7600.bmp").  
mobile("Nokia","6250","gray","98","11","1.3","c://mobiles/pm_6250.bmp").  
mobile("Nokia","6111","silver","92","12","1.4","c://mobiles/pm_6111.bmp").  
mobile("Nokia","6070","silver","92","8","no","c://mobiles/6070_pm.bmp").  
mobile("Nokia","6270","silver","100","12","1.2","c://mobiles/pm_6270.bmp").  
mobile("Nokia","8910i","black","105","12","no","c://mobiles/pm_8910i.bmp").  
mobile("Nokia","9210","silver","140","20","1.4","c://mobiles/pm_9210.bmp").  
mobile("Nokia","9300i","gray","135","18","1.4","c://mobiles/pm_9300i_phone.bmp").  
mobile("Nokia","6060","black","91.7","14","1.2","c://mobiles/pm_6060.bmp").  
mobile("Nokia","6150","red","102","8","no","c://mobiles/pm_6150.bmp").  
mobile("Nokia","6280","silver","115","16","1.2","c://mobiles/pm_6280.bmp").  
mobile("Nokia","6110","","100","8","no","c://mobiles/pm_6110.bmp").  
mobile("Nokia","5100","green","91","10","no","c://mobiles/pm_5100.bmp").  
mobile("Nokia","5500","","90","10","no","c://mobiles/pm_5100.bmp").
```

mobile("Nokia","N92","black","75","12","1.2","c://mobiles/n92.bmp").
 mobile("Nokia","N93","black","70","12","1.4","c://mobiles/n93_pm.bmp").
 mobile("Nokia","N1112","silver","80","8","no","c://mobiles/n1112_pm.bmp").
 mobile("Nokia","3330","red","101","6","no","c://mobiles/pm_3330.bmp").
 mobile("Nokia","3300","","125","14","no","c://mobiles/pm_3300.bmp").
 mobile("Motorola","C139","black","95","4","no","c://mobiles/c139_smnw.bmp").
 mobile("Motorola","C261","black","100","4","no","c://mobiles/c261_phoneselect.bmp").
 mobile("Motorola","E816","black","102","10","no","c://mobiles/E816_phoneselect.bmp").
 mobile("Motorola","V3","silver","100","11","","c://mobiles/V3_phoneselect.bmp").
 mobile("Motorola","V3c","gray","98","11","no","c://mobiles/V3c_phoneselect.bmp").
 mobile("Motorola","V3m","silver","95","","","c://mobiles/V3m_phoneselect.bmp").
 mobile("Motorola","V173","silver","100","14","no","c://mobiles/V173_phoneselect.bmp").
 mobile("Motorola","V188","black","90","15","1.2","c://mobiles/V188_phoneselect.bmp").
 mobile("Motorola","V360","blue","100","16","1.2","c://mobiles/V360_phoneselect.bmp").
 mobile("Motorola","Q","silver","120","16","1.3","c://mobiles/Q_phoneselect.bmp").
 mobile("Motorola","ROKR","silver","115","16","1.4","c://mobiles/ROKR_phoneselect.bmp").

mobile("Samsung","x507","black","108","15","1.3","c://mobiles/x507web_lg.bmp").
 mobile("Samsung","x507","black","105","16","1.4","c://mobiles/x507web_med.bmp").
 mobile("Samsung","p777","silver","116","14","1.2","c://mobiles/p777duo.bmp").
 mobile("Samsung","T809","gray","125","18","1.3","c://mobiles/t809duo.bmp").
 mobile("Samsung","E316","silver","120","16","1.2","c://mobiles/e316duo.bmp").
 mobile("Samsung","E635","gray","114","16","1.3","c://mobiles/e635duo.bmp").
 mobile("Samsung","T300","silver","90","8","no","c://mobiles/b2c_l_sch_t300.bmp").

mobile("Philips","7360","yellow","104","10","1.2","c://mobiles/pm_7360.bmp").
 mobile("Philips","7370","black","100","12","1.2","c://mobiles/pm_7370.bmp").
 mobile("Philips","7380","yellow","90","14","1.3","c://mobiles/pm_7380.bmp").
 mobile("Philips","sa7","blue","100","12","1.2","c://mobiles/sa7.bmp").
 mobile("Philips","sc3","black","105","14","1.3","c://mobiles/sc3.bmp").

mobile("LG","AX5000","red","91","12","1.4","c://mobiles/AX5000_l.f.bmp").
 mobile("LG","AX4270","black","85","11","1.2","c://mobiles/AX4270_m.f.bmp").
 mobile("LG","AX5000","silver","90","12","1.3","c://mobiles/AX5000_m.o.bmp").

mobile("SonyEricsson","P990","silver","155","14","1.1","c://mobiles/p990-1.bmp").

.....
 ,
ამოცანა 6. ელექტრო გიტარების ცოლნის ბაზის ფორმირება გრაფიკული
 პროდუქტების გამოყენებით

შედეგადად პროდუქტ-პროგრამის სრული ვარიანტი:

?-window(_win_func(),"Guitar Catalogue",0,0,400,500).

win_func(init):-

```

bitmap(_,_ ,fail(),"c://Guitars/menu.bmp",0,0),

menu(pop_up,_ ,way_out(),"File"),
menu(pop_up,_ ,search(),"Guitars"),
menu(pop_up,_ ,info(),"Help"),

edit(G_P1,_ ,edit_func(),"",120,10,200,25),
edit(G_P2,_ ,edit_func(),"",120,60,200,25),
edit(G_P3,_ ,edit_func(),"",120,110,200,25),
edit(G_P4,_ ,edit_func(),"",120,160,200,25),
edit(G_P5,_ ,edit_func(),"",120,210,200,25),
edit(G_P6,_ ,edit_func(),"",120,270,200,25),
edit(G_P7,_ ,edit_func(),"",120,330,200,25),

static(_ ,_ ,st_func(),"Brand",10,15,38,15),
static(_ ,_ ,st_func(),"Model",10,65,38,15),
static(_ ,_ ,st_func(),"Color",10,115,35,15),
static(_ ,_ ,st_func(),"Frets Number",10,165,89,15),
static(_ ,_ ,st_func(),"Wood",10,215,38,15),
static(_ ,_ ,st_func(),"Bridge Type",10,275,80,15),
static(_ ,_ ,st_func(),"Path to the file",10,335,94,15),

button(_ ,_ ,bfunc(),"Save",310,420,70,25).

win_func(close):-
not(yes_no("Exit","Do you want to exit?",?)).

way_out(init):-
menu(normal,_ ,_ ,menu_exit(),"Exit").

menu_exit(press):-
close_window().

info(init):-
menu(normal,_ ,_ ,creator(),"About"),
menu(normal,_ ,_ ,internet(),"Information").

creator(press):-
message("Author","This Software was built by Vitali Aivazov
14 May 2005",n).

internet(press):-
message("Web Sites","For more detailed information please visit
the following sites:
http://www.ibanez.ru
http://www.jacksonsguitars.com
http://www.gibson.com
http://www.fender.com",n).

edit_func(init):-
color_text(_ ,rgb(0,255,10)),
color_text_back(_ ,rgb(0,0,0)).

```

```

st_func(init):-
color_text(_,rgb(255,255,255)),
color_text_back(_,rgb(0,0,0)).

```

```

search(init):-
    menu(normal,_,t1(),"Brand"),
    menu(normal,_,t2(),"Model"),
    menu(normal,_,t3(),"Color"),
    menu(normal,_,t4(),"Frets"),
    menu(normal,_,t5(),"Wood"),
    menu(normal,_,t6(),"Bridge").

```

```

t1(press):-
read(X,"Brand of Guitar",s),
write("Brand"),write(" "),
write("Model"),write(" "),
write("Color"),write(" "),
write("Frets"),write(" "),
write("Wood"),write(" "),
write("Bridge"),nl,nl,
guitar(X,G_Y,Z,A,B,C,G_P),
write(X),write(" "),
write(G_Y),write(" "),
write(Z),write(" "),
write(A),write(" "),
write(B),write(" "),
write(C),nl,
far_way(),fail.

```

```

t2(press):-
read(G_Y,"Model of Guitar",s),
write("Brand"),write(" "),
write("Model"),write(" "),
write("Color"),write(" "),
write("Frets"),write(" "),
write("Wood"),write(" "),
write("Bridge"),nl,nl,
guitar(X,G_Y,Z,A,B,C,G_P),
write(X),write(" "),
write(G_Y),write(" "),
write(Z),write(" "),
write(A),write(" "),
write(B),write(" "),
write(C),nl,
far_way(),fail.

```

```

t3(press):-
read(Z,"Color of Guitar",s),
write("Brand"),write(" "),
write("Model"),write(" "),
write("Color"),write(" "),

```

```

write("Frets"),write(" "),
write("Wood"),write(" "),
write("Bridge"),nl,nl,
guitar(X,G_Y,Z,A,B,C,G_P),
write(X),write(" "),
write(G_Y),write(" "),
write(Z),write(" "),
write(A),write(" "),
write(B),write(" "),
write(C),nl,
far_way(),fail.

```

```

t4(press):-
read(A,"Number of Frets",s),
write("Brand"),write(" "),
write("Model"),write(" "),
write("Color"),write(" "),
write("Frets"),write(" "),
write("Wood"),write(" "),
write("Bridge"),nl,nl,
guitar(X,G_Y,Z,A,B,C,G_P),
write(X),write(" "),
write(G_Y),write(" "),
write(Z),write(" "),
write(A),write(" "),
write(B),write(" "),
write(C),nl,
far_way(),fail.

```

```

t5(press):-
read(B,"Wood of Body",s),
write("Brand"),write(" "),
write("Model"),write(" "),
write("Color"),write(" "),
write("Frets"),write(" "),
write("Wood"),write(" "),
write("Bridge"),nl,nl,
guitar(X,G_Y,Z,A,B,C,G_P),
write(X),write(" "),
write(G_Y),write(" "),
write(Z),write(" "),
write(A),write(" "),
write(B),write(" "),
write(C),nl,
far_way(),fail.

```

```

t6(press):-
read(C,"Bridge",s),
write("Brand"),write(" "),
write("Model"),write(" "),
write("Color"),write(" "),
write("Frets"),write(" "),

```

```

write("Wood").write(" "),
write("Bridge").nl,nl,
guitar(X,G_Y,Z,A,B,C,G_P),
write(X).write(" "),
write(G_Y).write(" "),
write(Z).write(" "),
write(A).write(" "),
write(B).write(" "),
write(C).nl,
far_way(_),fail.

```

```

far_way(_):-
window(_,_wfunc(_),G_Y,100,50,250,600).

```

```

wfunc(init):-
window_brush(_rgb(0,0,0)),
bitmap(_,_fail(_),G_P,0,0).

```

```

bfunc(press):-
X is get_text(G_P1),
Y is get_text(G_P2),
Z is get_text(G_P3),
A is get_text(G_P4),
B is get_text(G_P5),
C is get_text(G_P6),
D is get_text(G_P7),
assert_in(guitar(X,Y,Z,A,B,C,D)),
message("Congratulations!", "You have successfully refilled database!",!),

```

```

set_text("",G_P1),
set_text("",G_P2),
set_text("",G_P3),
set_text("",G_P4),
set_text("",G_P5),
set_text("",G_P6),
set_text("",G_P7).

```

```

guitar("berich","acrylic","green","24","basswood","tune-o-
matic","c://guitars/acrylic.bmp").
guitar("berich","beast","red","24","alder","floyd rose","c://guitars/beast.bmp").
guitar("berich","bich","violet","24","alder","floyd rose","c://guitars/bich.bmp").
guitar("berich","bronze","red","24","beech","static bridge","c://guitars/bronze.bmp").
guitar("berich","mb","light red","24","lime","double locking FR","c://guitars/mb.bmp").
guitar("berich","plat","platinum","24","oak","synchronized
tremolo","c://guitars/plat.bmp").
guitar("berich","warlock","brown","24","basswood","edge
pro","c://guitars/warlock.bmp").
guitar("fender","cyclon","yellow","22","alder","synchronized
tremolo","c://guitars/cyclon.bmp").
guitar("fender","esquire","grey","22","lime","static bidge","c://guitars/esquire.bmp").

```

```

guitar("fender","jaguar","light blue","22","ohm","tune-o-
matic","c://guitars/jaguar.bmp").
guitar("fender","mustang","yellow","22","oak","static
bridge","c://guitars/mustang.bmp").
guitar("fender","showmaster","red","22","lime","synchronized
tremolo","c://guitars/showmaster.bmp").
guitar("fender","stratocaster","blue","22","oak","synchronized
tremolo","c://guitars/stratocaster.bmp").
guitar("fender","telecaster","brown","22","alder","static
bridge","c://guitars/telecaster.bmp").
guitar("gibson","blueshawk","red","22","mahogany","static
bridge","c://guitars/blueshawk.bmp").
guitar("gibson","explorer","black","22","ohm","static bridge","c://guitars/explorer.bmp").
guitar("gibson","firebird5","brown","22","alder","static
bridge","c://guitars/firebird5.bmp").
guitar("gibson","firebird7","red","22","beech","tune-o-
matic","c://guitars/firebird7.bmp").
guitar("gibson","flying","white","22","basswood","static bridge","c://guitars/flying.bmp").
guitar("gibson","les paul","orange","22","mahogany","static
bridge","c://guitars/lespaul.bmp").
guitar("gibson","sg","brown","22","oak","floyd rose","c://guitars/sg.bmp").
guitar("gibson","sgs","red","22","mahogany","tune-o-matic","c://guitars/sgs.bmp").
guitar("ibanez","af75","brown","22","lime","static bridge","c://guitars/af75.bmp").
guitar("ibanez","dmm1","painted","22","basswood","strings-through-
body","c://guitars/dmm1.bmp").
guitar("ibanez","jem77","painted","24","basswood","double locking
FR","c://guitars/jem77.bmp").
guitar("ibanez","k7","grey","24","basswood","double locking FR","c://guitars/k7.bmp").
guitar("ibanez","rg2620","light blue","24","mahogany","double locking
FR","c://guitars/rg2620.bmp").
guitar("ibanez","rg370dx","black","24","basswood","edge pro
2","c://guitars/rg370dx.bmp").
guitar("ibanez","rga121","red","24","ohm","edge pro","c://guitars/rga121.bmp").
guitar("ibanez","sa120","black","22","alder","edge pro 2","c://guitars/sa120.bmp").
guitar("jackson","js1","blues","22","oak","edge pro","c://guitars/js1.bmp").
guitar("jackson","dk2","red","24","beech","double locking FR","c://guitars/dk2.bmp").
guitar("jackson","dkmg","brown","24","ohm","strings-through-
body","c://guitars/dkmg.bmp").
guitar("jackson","firestorm","red","24","mahogany","floyd
rose","c://guitars/firestorm.bmp").
guitar("jackson","dk1","green","24","alder","edge pro 2","c://guitars/dk1.bmp").
guitar("jackson","ke3","painted","24","basswood","edge pro","c://guitars/ke3.bmp").
guitar("jackson","kvx10","black","24","lime","strings-through-
body","c://guitars/kvx10.bmp").
guitar("jackson","rr3","red","21","alder","synchronized tremolo","c://guitars/rr3.bmp").

```

პროცესი 1. ტესტ-პროგრამა საბავშვებელის ისტორიაში

მოყვანილი ტესტ-პროგრამის სრული ტექსტი:

%*****TESTIREBA SAQARTVELOS ISTORIASHI*****

%*****SAQARTVELOS TEQNIKURI UNIVERSITETI****

?-kitxva1.

%?-kitxva2.

%?-kitxva3.

%?-kitxva4.

%?-kitxva5.

%?-kitxva6.

%?-kitxva7.

%?-kitxva8. ,

%?-kitxva9.

%?-kitxva10.

%***** CODNIS BAZA *****

%PIRVELI KITXVA

```

kitxva1:- write("*****KITXVA PIRVELI*****"),
nl,nl,nl,

write("ROMEL CELS MONDA DIDGORIS BRDZOLA?"),nl,nl,nl,

write("a) 1452 b)1121 c)1089 d)1745"),nl,nl,nl,
read(Pasuxi).Pasuxi=1121->G_A=1,nl,nl,nl,

write("SCORIA!"),nl,nl,nl,write("*****"),
nl,nl,nl,

write("TQVEN MIIGET "),write(G_A),write(" QULA!"),nl,nl,nl,kitxva2

    else G_A=0, write("MCDARIA!"),nl,nl,nl,
write("*****"),nl,

nl,nl,write("TQVEN MIIGET "),write(G_A),write(" QULA!"),nl,nl,
nl,kitxva2,nl,nl,nl.

%MEORE KITXVA

kitxva2:-write("*****KITXVA MEORE*****"),nl,nl,
nl,write("VIN IKO TAMAR MEPIS MAMA?"),nl,nl,nl,

write("a) giorgi_3 b)davit2 c)erkle d)simoni "),nl,nl,nl,read(Pasuxi),
Pasuxi=giorgi_3->G_B=1.write("SCORIA!"),nl,nl,nl,

write("*****"),nl,nl,nl,

write(" TQVEN MIIGET "),write(G_B),write(" QULA!"),nl,

nl,nl,kitxva3

else G_B=0,write("MCDARIA!"),nl,nl,nl,

write("*****"),nl,nl,nl,write("TQVEN MIIGET "),

```

```

write(G_B),write(" QULA!"),nl,nl,nl,kitxva3.

%MESAME KITXVA

kitxva3:-write("*****MESAME KITXVA*****"),nl,nl,nl,
write("ROMEL CELS MOXDA ASPINDZIS BRDZOLA?"),nl,nl,nl,

write("a)1245 b) 1770 c)1851 c)4511"),nl,nl,nl,read(Pasuxi),
Pasuxi=1770->G_C=1,

write("SCORIA!"),nl,nl,nl,write("*****"),nl,nl,nl,
write("TQVEN MIIGET "),

write(G_C),write(" QULA!"),nl,nl,nl,kitxva4

    else G_C=0,write("MCDARIA!"),nl,nl,nl,
write("*****"),nl,nl,nl,
write("TQVEN MIIGET "),write(G_C),write(" QULA!"),nl,nl,nl,kitxva4.

%MEOTXE KITXVA

kitxva4:-write("*****MEOTXE KITXVA*****"),nl,nl,nl,

write("VIN DACERA SHUSHANIKIS CAMEBA"),nl,nl,nl,

write("a) iakob_xucesi b) chavchavadze c) iovane_sababisdze d)
vaga_pshavela"),nl,nl,nl,read(Pasuxi),

Pasuxi=iakob_xucesi->G_D=1,write(" SCORIA!"),nl,nl,nl,

write("*****"),nl,nl,nl,write("TQVEN MIIGET "),

write(G_D),write(" QULA! "),nl,nl,nl,kitxva5

,

else
G_D=0,write("MCDARIA!"),nl,nl,nl,write("*****"),nl,nl,nl,
write("TQVEN MIIGET "),write(G_D),write(" QULA!"),nl,nl,nl,kitxva5.

%MEXUTE KITXVA

kitxva5:-write("*****MEXUTE KITXVA*****"),nl,nl,
nl,write("VIN IKO SAQARTVELOS UKANASKNELI MEPE?"),nl,nl,nl,

```

```

write("a)giorgi_12 b)simon_1 c)davit_4 d)alexandre_1"),nl,nl,nl,

read(Pasuxi).Pasuxi=giorgi_12->G_E=1,write("SCORIA!"),nl,nl,nl,
write("*****"),nl,nl,nl,

write("TQVEN MIIGET "),write(G_E),write(" QULA!"),nl,nl,nl,kitxva6

    else G_E=0,write("MCDARIA!"),nl,nl,nl,write("*****"),
nl,nl,nl,

write("TQVEN MIIGET "),write(G_E),write(" QULA!"),nl,nl,nl,kitxva6.

%MEEQVSE KITXVA

kitxva6:-write("*****KITXVA MEEQVSE*****"),nl,nl,nl,

write("ROMEL CELS SHEUERTDA SAQARTVELO RUSETS?"),nl,
nl,nl,

write("a)1945 b)1801 c)1745 d)975"),nl,nl,nl,

read(Pasuxi).Pasuxi=1801->G_F=1,write("SCORIA!"),nl,nl,nl,
write("*****"),nl,nl,nl,
write("TQVEN MIIGET "),write(G_F),write(" QULA!"),nl,nl,nl,kitxva7

    else G_F=0,

write("MCDARIA!"),nl,nl,nl,write("*****"),nl,nl,nl,

write("TQVEN MIIGET "),write(G_F),write(" QULA!"),nl,nl,nl,kitxva7.

%MESHVIDE KITXVA

kitxva7:-write("*****MESHVIDE KITXVA*****"),nl,nl,nl,

write("ROMEL CELS GARDAICVALA TAMAR-MEPE?"),nl,nl,nl,

write("a)1213 b)1475 c)1241 d)1741"),nl,nl,nl,

read(Pasuxi).Pasuxi=1213->G_G=1,write("SCORIA!"),nl,nl,nl,

write("*****"),nl,nl,nl,

write("TQVEN MIIGET ").write(G_G).write(" QULA!"),nl,nl,nl,kitxva8

else G_G=0,

```

write("MCDARIA!"),nl,nl,nl,write("*****"),nl,nl,nl,

write("TQVEN MIIGET "),write(G_G),write(" QULA!"),nl,nl,nl,kitxva8.

%MERVE KITXVA

kitxva8:-write("*****MERVE KITXVA*****"),nl,nl,nl,

write("VIN IKO SAQARTVELOS PIRVELI PREZIDENTI?"),nl,nl,nl,

write("a)gamsaxurdia b)shevardnadze c)jordanian d)saakashvili"),nl,nl,nl,

read(Pasuxi),Pasuxi=gamsaxurdia->G_H=1,write("SCORIA!"),nl,nl,nl,

write("*****"),nl,nl,nl,

write("TQVEN MIIGET "),write(G_H),write(" QULA!"),nl,nl,nl,kitxva9

else G_H=0,

write(" MCDARIA"),nl,nl,nl,write("*****"),nl,nl,nl,

write("TQVEN MIIGET ").write(G_H),write("QULA!"),nl,nl,nl,kitxva9.

%MECXRE KITXVA

kitxva9:-write("*****KITXVA MECXRE*****"),nl,nl,nl,

write("ROMEL CELS MOXDA KRCANISIS BRDZOLA?"),nl,nl,nl,

write("a)1213 b)1795 c)1241 d)1741"),nl,nl,nl,

read(Pasuxi),Pasuxi=1795->G_I=1,write("SCORIA!"),nl,nl,nl,

write("*****"),nl,nl,nl,

write("TQVEN MIIGET ").write(G_I),write("QULA!"),nl,nl,nl,kitxva10

else G_I=0,

write("MCDARIA!"),nl,nl,nl,write("*****"),nl,nl,nl,

```

write("TQVEN MIIGET "),write(G_I),write(" QULA!"),nl,nl,nl,kitxva10.

%MEATE KITXVA

kitxva10:-write("*****MEATE KITXVA*****"),nl,nl,nl,
write("ROMEL CELS GAXDA SAQARTVELO DAMOUKIDEBELI?"),nl,nl,nl,
write("a)1213 b)1475 c)1991 d)1741"),nl,nl,nl,
read(Pasuxi),Pasuxi=1991->G_J=1,write("SCORIA!"),nl,nl,nl,
write("*****"),nl,nl,nl,
write("TQVEN MIIGET "),write(G_J),write(" QULA!"),nl,nl,nl,testireba

else G_J=0.

write("MCDARIA!"),nl,nl,nl,write("*****"),nl,nl,nl,
write("TQVEN MIIGET "),write(G_J),write(" QULA!"),nl,nl,nl,testireba.

testireba:-write("TQVENS MIER DAGROVILI QULATA JAMIA: "),
G_Z=G_A+G_B+G_C+G_D+G_E+G_F+G_G+G_H+G_I+G_J,write(G_Z),write("
QULA!").

```

პროგრამა 2. მონქოვოს ავტომობილები „MERCEDES-ის ცენტრი

მედიკინის ცენტრ-პროგრამის სრული ტექსტი:

```

%      MERCEDES CENTRE. AVTOHOUSE OF MOSCOW.
%      -----
%-

G_Sum=0.
G_Num=1.
question(G_Num.Q|G_L),
G_Q:=Q.
window( _ , _ , win_func(_ , "Question " + print(G_Num), 38, 110, 350, 615).

win_func(init):-
    menu(pop_up, _ , _ , menu_window(_ , "File"),
    menu(pop_up, _ , _ , menu_go(_ , "Avtomobiles"),

```

```

        menu(normal,_,_, menu_j(), "Go on"),
        menu(pop_up,_,_, menu_about(), "About"),
        window_brush(_, rgb(0, 0, 0)),
        icon(_, fail(), icon, 373, 15),
        window_brush(_, rgb(0, 0, 0)),

```

```

button(_,b1(),"M Player" 360,525,70,20),

```

```

        X is print("C://mercedes//"),

```

```

bitmap(_,f1(), X +print("My M.bmp") ,65,2),

```

```

bitmap(_,f2(), X +print("My CLS.bmp") ,65,250),

```

```

        static(_,_, inf(), "St.

```

```

P
r
o
l
o
g", 380, 60,20, 144),
        window_brush(_, rgb(0, 0, 0)),
        color_text(_, rgb(180, 0, 0)),
        color_text_back(_, rgb(0, 0, 0)),

```

```

        static(_,_, inf(), "AUCILEBELIA TU ARA HQONDES —", 60, 10,220, 35),
        window_brush(_, rgb(0, 0, 0)).
        color_text(_, rgb(250, 0, 0)),
        color_text_back(_, rgb(0, 0, 0)).

```

```

        static(_,_, inf(), G_Q,20,50,300,20),
        window_brush(_, rgb(0, 0, 0)),
        color_text(_, rgb(0, 170, 0)),
        color_text_back(_, rgb(0, 0, 0)).

```

```

        static(_,_, inf(), " MOSKOVIS AVTOBAZARZE IKIDEBA MERSEDESIS
SXVADASXVA MODELEBI: ",25,240,300,40),
        window_brush(_, rgb(0, 0, 0)).
        color_text(_, rgb(0, 133, 255)).
        color_text_back(_, rgb(0, 0, 0)).

```

```

        static(_,_, inf(), "I

```

```

2
3
4
5
6
7
8
9

```

```

10
11
12
13
14", 380, 290, 20, 235),
    window_brush(_, rgb(0, 0, 0)),
    color_text(_, rgb(0, 170, 0)),
    color_text_back(_, rgb(0, 0, 0)),

    make_but(G_L, 85).

menu_window(init) :-
    menu( normal, _, _, menu_win(_), "Wish"),
    menu( normal, _, _, dr(_), "Delete Record"),
    menu( normal, _, _, menu_exit(_), "Exit").

menu_win(press) :-
    %beep("C:\\mercedes\\Windows XP Exclamation.wav"),
    %beep("C:\\My Music\\super musik\\Universiteti\\5.wav"),
    window(_, _, menu_func(_), "Wish", 43, 160, 339, 270).

menu_func(init) :-
    menu(normal, _, _, menu_exit(_), "&Exit"),

    static(_, _, inf(_), "Welcome!

ketili ikos tkveni mobrdzaneba
Mercedes-is samkaroshi.

shedzenis shemtxvevashi:

gisurvebt bednierebasa da carmatebebs!", 1, 1, 339, 240),

    window_brush(_, rgb(167, 167, 167)),
    color_text(_, rgb(0, 0, 138)),
    color_text_back(_, rgb(167, 167, 167)).
dr(press):-
retract_in(file(X,T)).

menu_exit(press) :- close_window(_). !, fail.

menu_j(press) :-
    size(_, X, Y),
    size(_, X+2, Y).
    X<450.
menu_j(press).

```

```

menu_about(init):-
    menu( normal, _, _, menu_about_inf(), "Info"),
    menu( normal, _, _, menu_about_author(), "By Author").

menu_about_inf(press):-
    /*beep("C:\\mercedes\\Windows XP Exclamation.wav"),*/
    window( _, _, m_func(), "Info", 43, 160, 500, 350).

m_func(init) :-
    menu(normal, _, _, menu_exit(), "&Exit"),

    static( _, _, inf(), "My Program Help

90 -- sruli kompleqtit.(CD audiommartvelit da DU saketit).
85 -- kondincioneris, CD da DU gareshe.
80 -- kondincionerisa, luqis, CD da DU gareshe.
75 -- kondincionerisa, luqis, tkavis salonis, CD da DU gareshe.
70 -- kondincioneris, klimat--maregulirebelis, tkavis salonis,
    kresloebs elektro amdzravis, luqis, CD da DU gareshe.
69-1 -- am qulebistvis shesabamisi avtomanqana bazarze ar aris.
4-0 -- tqveni mosvlis mizezi gaugebaria.
    ", 1, 1, 500, 350),

    window_brush( _, rgb(167, 167, 167)),
    color_text( _, rgb(0, 0, 138)),
    color_text_back( _, rgb(167, 167, 167)).

menu_about_author(press):-
    message("By Author", " This program is created by Irakli Lemonjava <--- 13 / 06 /
2005 ---> ", n).

b1(press):-
    K is ("C:\\Program Files\\Windows Media Player\\wmplayer.exe"),
    chronometer( _),
    execute(K),
    chronometer(T),
    assert_in(file(K,T)), nl,
    Y is print(T),
    set_text(Y, G_J).

make_but([[_A, _]|T], N):-
    button( _, _, but_func( _), A, 20, N, 300, 20),
    N1 is N+40,
    make_but(T, N1).

but_func(press):-
    N is get_text( _),
    add_sum(N, G_L),
    close_window(parent( _)),
    G_Num:=G_Num+1,

```

```

G_L:=L,
(question(G_Num,Q|G_L)->
G_Q:=Q,

window( _,_, win_func(_), "Question " + print(G_Num), 38, 110, 350, 615)
else
    message("result", "

    meti informaciiisatvis shedit 'Avtomobiles'--shi)
im shemtxxevashi, roca qula naklebia 70-ze, meti informaciiisatvis naxet: 'About-->Info' .

    tqveni shedegia {Your result is} – "+print(G_Sum),                !)).

add_sum(N, [|N,P|_]):-
    G_Sum:=G_Sum+P.

add_sum(N, |_|L):-
    add_sum(N, L).

win_func(paint):-
    pen(2, rgb(200, 200, 200)), line(-200000,0,200000,440).

menu_go(init):-
    menu(normal, _,s1(_),"90 - Qula "),
    menu(normal, _,s2(_),"85 - Qula "),
    menu(normal, _,s3(_),"80 - Qula "),
    menu(normal, _,s4(_),"75 - Qula "),
    menu(normal, _,s5(_),"70 - Qula ").

s1(press):-
    read(X, "90 - Qula",s),nl,

write("Qula"),                write("  "),
write("Model"),                write("  "),
write("Dzaris Tipi"),          write("  "),
write("Karis da Adgilis Raod."), write("  "),
write("Dzravis Tipi"),          write("  "),
write("Cilindris Raod."),      write("  "),
write("Samushao Mocoloba"),     write("  "),
write("Simdžlavre (c.dz)"),     write("  "),
write("Amdzravi"),              write("  "),
write("Gadacemis Kolophi"),     write("  "),
write("Sichqare"),              write("  "),
write(" Phasi"),                write("  "),
write(" Gamkidveli Phirma"),    write("  "),
write("Sakontaqto Telephoni"),nl,nl.

avtomobile(X,Y,Z,A,B,C,D,E,F,G,H,I,J,K).
write(X),write("  ").
write(Y),write("  ").
write(Z),write("  ").

```

```

write(A),write("  "),
write(B),write("  "),
write(C),write("  "),
write(D),write("  "),
write(E),write("  "),
write(F),write("  "),
write(G),write("  "),
write(H),write("  "),
write(I),write("  "),
write(J),write("  "),
write(K),nl,fail.

```

s2(press):-

```

    read(X, "85 - Qula",s),nl,
write("Qula"),           write("  "),
write("Model"),          write("  "),
write("Dzaris Tipi"),    write("  "),
write("Karis da Adgilis Raod."), write("  "),
write("Dzravis Tipi"),   write("  "),
write("Cilindris Raod."), write("  "),
write("Samushao Mocoloba"), write("  "),
write("Simdzlavre (c.dz)", write("  "),
write("Amdzravi"),        write("  "),
write("Gadacemis Kolophi"), write("  "),
write("Sichqare"),        write("  "),
write(" Phasi"),          write("  "),
write(" Gamlkidveli Phirma"), write("  "),
write("Sakontaqto Telephoni"),nl,nl,

```

avtomobile(X,Y,Z,A,B,C,D,E,F,G,H,I,J,K),

```

write(X),write("  "),
write(Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),
write(C),write("  "),
write(D),write("  "),
write(E),write("  "),
write(F),write("  "),
write(G),write("  "),
write(H),write("  "),
write(I),write("  "),
write(J),write("  "),
write(K),nl,fail.

```

s3(press):-

```

    read(X, "80 - Qula",s),nl,
write("Qula"),           write("  "),
write("Model"),          write("  "),
write(" Dzaris Tipi"),   write("  "),
write("Karis da Adgilis Raod."). write("  "),
write("Dzravis Tipi"),   write("  "),

```

```

write("Cilindris Raod."),      write("  "),
write("Samushao Mocoloba"),    write("  "),
write("Simdzlavre (c.dz)"),    write("  "),
write("Amdzravi"),             write("  "),
write("Gadacemis Kolophi"),    write("  "),
write("Sichqare"),             write("  "),
write("Phasi"),                write("  "),
write(" Gamkidveli Phirma"),    write("  "),
write("Sakontaqto Telephoni").nl,nl,

```

```

avtomobile(X,Y,Z,A,B,C,D,E,F,G,H,I,J,K),

```

```

write(X),write("  "),
write(Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),
write(C),write("  "),
write(D),write("  "),
write(E),write("  "),
write(F),write("  "),
write(G),write("  "),
write(H),write("  "),
write(I),write("  "),
write(J),write("  "),
write(K).nl,fail.

```

```

s4(press):-

```

```

    read(X, "75 - Qula",s).nl,
write("Qula"),      write("  "),
write("Model"),     write("  "),
write("Dzaris Tipi"), write("  "),
write("Karis da Adgilis Raod."), write("  "),
write("Dzravisi Tipi"), write("  "),
write("Cilindris Raod."), write("  "),
write("Samushao Mocoloba"), write("  "),
write("Simdzlavre (c.dz)"), write("  "),
write("Amdzravi"),       write("  "),
write("Gadacemis Kolophi"), write("  "),
write("Sichqare"),       write("  "),
write("Phasi"),          write("  "),
write(" Gamkidveli Phirma"), write("  "),
write(" Sakontaqto Telephoni").nl,nl,

```

```

avtomobile(X,Y,Z,A,B,C,D,E,F,G,H,I,J,K).

```

```

write(X),write("  "),
write(Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),
write(C),write("  "),
write(D),write("  "),
write(E),write("  "),

```

```

write(F),write("  "),
write(G),write("  "),
write(H),write("  "),
write(I),write("  "),
write(J),write("  "),
write(K),nl,fail.

```

s5(press):-

```

read(X, "70 - Qula",s),nl,
write("Qula"), write("  "),
write("Model"), write("  "),
write("Dzaris Tipi"), write("  "),
write("Karis da Adgilis Raod."), write("  "),
write("Dzravis Tipi"), write("  "),
write("Cilindris Raod."), write("  "),
write("Samushao Mocoloba"), write("  "),
write("Simdzlavre (c.dz)"), write("  "),
write("Amdzravi"), write("  "),
write("Gadacemis Kolophi"), write("  "),
write("Sichqare"), write("  "),
write(" Phasi"), write("  "),
write("Gamkidveli Phirma"), write("  "),
write("Sakontaqto Telephoni"),nl,nl,

```

avtomobile(X,Y,Z,A,B,C,D,E,F,G,H,I,J,K),

```

write(X),write("  "),
write(Y),write("  "),
write(Z),write("  "),
write(A),write("  "),
write(B),write("  "),
write(C),write("  "),
write(D),write("  "),
write(E),write("  "),
write(F),write("  "),
write(G),write("  "),
write(H),write("  "),
write(I),write("  "),
write(J),write("  "),
write(K),nl,fail.

```

%@@@@@@@@@@@@@@@@ K I T X V E B I : @@@@@@@@@@@@@@@@@@

```

question(1,"ABS SISTEMA ?", ["No", 0], ["Yes", 5], ["Next -->", 0]).
question(2,"USAFRTXOEBIS BALISHEBI ?", ["Yes", 5], ["No", 0], ["Next -->", 0]).
question(3,"SACHIS HIDROMADZLIEREBELI ?", ["No", 0], ["Yes", 5], ["Next -->", 0]).
question(4,"SACHIS KALONKIS MAREGULIREBELI ?", ["Yes", 5], ["No", 0], ["Next -->", 0]).
question(5,"DAMCAVI SISTEMA ?", ["No", 0], ["Yes", 5], ["Next -->", 0]).
question(6,"IMOBILLAIZERI ?", ["Yes", 5], ["No", 0], ["Next -->", 0]).
question(7,"CENTRALURI SAKETI ?", ["No", 0], ["Yes", 5], ["Next -->", 0]).

```


avtomobile("75","E 240"," SEDANI"," 4/5","	B PRISKIT"," 6","
2597 sm/kubi"," 177(c.dz.)"," UKANA"," A","	260 km/h","57 568 EURO","
AVTOFORUMI"," 730-11-11").	
avtomobile("75","E 200 Komp.,"SEDANI"," 4/5","	B PRISKIT"," 4","
1796 sm/kubi"," 163(c.dz.)"," UKANA"," A","	255 km/h","46 900 EURO","
AVTOFORUMI"," 730-11-11").	
avtomobile("70"," A 140","HECHBEKI"," 5/5","	B PRISKIT"," 4","
1397 sm/kubi"," 82(c.dz.)"," CINA"," A","	220 km/h","29 533 EURO","
AVTOFORUMI"," 730-11-11").	

ლიტერატურა

1. Н. Братко. Программирование на языке Пролог для искусственного интеллекта: Москва. „Мир“. 1990. 560 стр.
2. Л.Стерлинг, Э. Шапиро. Искусство программирования на языке Пролог. Москва. „Мир“. 1990. 235 стр.
3. Дж. Уотерман. Экспертные системы. Москва. „Мир“. 1990. 250 стр.
4. Люгер, Джордж, Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. 4-ое издание. Москва. „Вильямс“. 2003. 864 стр.
5. Лорьер Ж.Л. Системы искусственного интеллекта. Москва. „Мир“. 1993. 568 стр.
6. Клоксин У. , Меллиш К. Программирование на языке Пролог. Москва. „Мир“. 1987. 336 стр.
7. ბ. კიზირია. ხელოვნური ინტელექტის საფუძვლები. თბილისი. „ტექნიკური უნივერსიტეტი“. 2005. 215 გვ.

შესავალი	3
თავი 1. ხელოვნური ინტელექტის სისტემების დაპროექტების პროგრამული საშუალებები	5
1.1. „Strawberry Prolog“ იმიტატორის სამუშაო გარემოს აღწერა	7
1.2. მუშაობა „Strawberry Prolog“ იმიტატორის ფანჯარაში	10
1.3. „Strawberry Prolog“ იმიტატორის ბიჯობრივი რეჟიმი	11
თავი 2. რელაციური ცოდნის ბაზების პროექტირება. მათი გამოყენება ხელოვნური ინტელექტის ამოცანებში	16
2.1. ამოცანა 1. ქართული მარანი	16
2.2. ამოცანა 2. ევროპის ჩემპიონთა ლიგა ფეხბურთში	24
2.3. ამოცანა 3. ავტომატური კატალოგი	31
2.4. ამოცანა 4. მსოფლიოს კურორტები	32
2.5. ამოცანა 5. მსოფლიოს ქვეყნები	35
თავი 3. ალტერნატივების გადარჩევის მართვა. გადაწყვეტილების მიღების პროცესის ავტომატიზირება	40
3.1. ჩაჭრის (cut) მექანიზმის გამოყენების მაგალითები	40
3.2. გადაწყვეტილების მიღების პროცესის დაპროგრამების მაგალითები პროლოგზე	48
3.2.1. ამოცანა 1. ფირმაში ვაკანტური თანამდებობის შერჩევა	48
3.2.2. ამოცანა 2. ფინანსური მრჩეველი	54
3.2.3. ამოცანა 3. საქართველოს რკინიგზა	58
3.2.4. ამოცანა 4. ქ. თბილისის სამარშრუტო ტაქსები	64
3.2.5. ამოცანა 5. კომპიუტერის კონფიგურაციისა და ღირებულების შერჩევა მყიდველის მოთხოვნათა შესაბამისად	66
თავი 4. „Strawberry Prolog“ იმიტატორის გრაფიკული შესაძლებლობები	72
4.1. პროლოგ-პროგრამების შედგენა ე.წ. „ჰილასიკური“ და გრაფიკული ოპერატორების გამოყენებით. ამ მეთოდების შედარებითი ანალიზი	72
4.2. პროლოგის იმიტატორის გრაფიკული ოპერატორების ზოგადი მიმოხილვა	90
4.3. რელაციური (ჩაშენებული) ბაზების პროექტირების ამოცანებში გრაფიკული პრედიკატების გამოყენება	95
4.3.1. ამოცანა 1. საქართველოს სამეფო დინასტიის გენეალოგიური ხე	96
4.3.2. ამოცანა 2. შემსრულებელი (exeCutor)	100
4.3.3. ამოცანა 3. საქართველოს ისტორიული ძეგლები	105

4.3.4. ამოცანა 4. ქართველი მხატვრები	115
4.3.5. ამოცანა 5. მობილური ტელეფონების მოდელები	120
4.3.6. ამოცანა 6. ელექტრო გიტარების ც/ბ-ის ფორმირება გრაფიკული პრედიკატების ბაზაზე	124
თავი 5. ტესტ-პროგრამების შედგენა „Strawberry Prolog“ იმიტატორის ბაზაზე	128
5.1. სტუდენტთა და მაგისტრთა რეიტინგული შეფასების ტესტ-პროგრამის (გარსის) აღწერა	130
5.2. ტესტ-პროგრამების დაპროგრამება ალტერნატიული მეთოდით	138
5.2.1. ამოცანა 1. ტესტ-პროგრამა საქართველოს ისტორიაში	138
5.2.2. ამოცანა 2. ქ. მოსკოვის ავტობაზარი. „Mercedes“ ცენტრი.	140
დავალებები 1.	146
დავალებები 2.	170
დავალებები 3.	196
დავალებები 4.	244
ლიტერატურა	259

იგეჭდება ავტორთა მიერ წარმოდგენილი სახით

გადაეცა წარმოებას 01.11.2006 წ. ხელმოწერილია დასაბუჯდად 13.11.2006 წ. ქალაქის
ზომა 60X84 1/8. პირობითი ნაბეჭდი თაბახი 16,5. ტირაჟი 100 ეგზ. შეკვეთა №

გამომცემლობა „ტექნიკური უნივერსიტეტი“, თბილისი, კოსტავას 77

