დაპროგრამების ენა C–ის სწავლების ზოგიერთი ასპექტი

კ.გელაშვილი*, ნ.არჩვაძე*, თ.დავითაშვილი*, ა.ჩიტალაძე*, ი.ხუციშვილი*

ი.ჯავახიშვილის სახ. თბილისის სახელმწიფო უნივერსიტეტი,
ზუსტი და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი, კომპიუტერულ მეცნიერებათა მიმართულება

ანოტაცია

სტატიაში გაანალიზებულია ის პრობლემები, რომლებიც თავს იჩენს დაპროგრამების ენა C–ის, როგორც დაპროგრამების პირველი ენის სწავლების დროს. შემოთავაზებულია C–ის სწავლების მეთოდიკა, რომლითაც სარგებლობენ სტატიის ავტორები ბაკალავრიატრის პირველ სემესტრში C ენის სწავლებისას. ავტორთა ძირითადი შეხედულებები და იდეები რეალიზებულია C ენის ლექციათა კურსის, პრაქტიკული და ლაბორატორიული მეცადინეობების, სავარჯიშოების კრებულების სახით.

I. შესავალი

თანამედროვე საგანმანათლებლო სისტემაში მიმდინარე ცვლილებები განპირობებულია შრომის საერთაშორისო ბაზრის მოთხოვნილებებით, რომელშიც ინფორმატიკის სექტორი სწრაფად ფართოვდება. ამ ცვლილებებს გლობალური პროცესების სახე აქვთ, რის თვალსაჩინო მაგალითს წარმოადგენს კარგად ცნობილი ბოლონიის პროცესი. სხვა მაგალითი, რომელიც ჩვენს საქმიანობაზე აგრეთვე არსებით გავლენას ახდენს, არის ინფორმატიკის სწავლების სტანდარტიზაციისკენ მიმართული ღონისძიებები. ეს ღონისძიებები ხორციელდება ავტორიტეტული და გავლენიანი საერთაშორისო ორგანიზაციების ACM (Association for Computing Machinary) და IEEE Computer Society მიერ. მათ ეკუთვნით ინფორმატიკის (Computing) 5 ძირითადი მიმართულების კურიკულუმები, რომლებიც პერიოდულად ქვეყნდება განახლებული სახით დაწყებული 1991 წლიდან. ამ კურიკულუმებით ხელმძღვანელობს ფაქტიურად ყველა წარმატებული უნივერსიტეტი და ისინი წარმოადგენენ ფაქტობრივ სტანდარტს შესაბამის მიმართულებებში.

მართალია, კომპიუტერული მეცნიერების კურიკულუმი (cs 2001, cs 2005) არ აკონკრეტებს დაპროგრამების ენებს და აყალიბებს მხოლოდ ენის სწავლების კრიტერიუმებს, პირველი დაპროგრამების ენა შესაძლებელია შეირჩეს ACM –ის პატრონაჟით ჩატარებული ღონისძიებების (Topcoder, ACM International Collegiate Programming Contest, International Olimpiad in Informatics) ოფიციალური ენებიდან: C, C++, Java. რადგან ბოლო ორი C–ის ევოლუციის შედეგს წარმოადგენს, ბუნებრივია არჩევანიც მასზე გაკეთდეს.

II. ძირითადი გამოწვევები

ჩვენი აზრით, ამჟამად საქართველოში დაპროგრამების ენა **C–ის** სწავლება დგას შემდეგი ძირითადი პრობლემების წინაშე:

a) არ არსებობს დაპროგრამების ენების სწავლების ოფიციალური **სტანდარტი**;
b) არაცალსახა შესაბამისობაა ინგლისურ და ქართულ ენაში დამკვიდრებულ **ტერმინოლოგიას** შორის. ქართულ ტერმინოლოგიას დახვეწა სჭირდება, რადგან

იგი ყოველთვის ოპტიმალურად არ ასახავს შესაბამისი ინგლისური ტერმინების შინაარსს;

c) **პირველ ენად** (ბაკალავრიატის პირველ სემესტრში) C ენის სწავლების ტრადიცია არ არის ხანგრძლივი და მდიდარი., ამიტომ აუცილებელია შესაბამისი **მეთოდიკის** შემუშავება.

d) შესაქმნელია ხარისხიანი **ქართულენოვანი** სახელმძღვანელოები, სავარჯიშოთა კრებულები და მასწავლებელთათვის განკუთვნილი მეთოდური მითითებები.

ამ პრობლემების ეფექტური დაძლევა არის აუცილებელი წინაპირობა იმ ახალი და უფრო არსებითი გამოწვევების დასაძლევად, რომელთა წინაშე დგას C-ზე უფრო მაღალი დონის, თანამედროვე ენების სწავლება.

III. სტანდარტი

დაპროგრამების ენის სწავლების სიღრმე და მიზნები დამოკიდებულია სწავლების საფეხურზე (ბაკალავრიატი, მაგისტრატურა, დოქტორანტურა). ამასთან სწავლების მაღალ საფეხურებზე მეტი თავისუფლებაა ენის სწავლების მასალის შერჩევაში და შედარებით ძნელია სტანდარტის განსაზღვრა.

ამჟამად, ბაკალავრიატში სწავლების რეალურ სტანდარტს წარმოადგენს ACM-ის კურიკულუმები. ჩვენ ძირითადად ვეყრდნობით კომპიუტერული მეცნიერების კურიკულუმს (http://www.acm.org/education/curricula/ComputerScienceCurriculumUpdate2008.pdf), რომლითაც ხელმძღვანელობს თითქმის ყველა წარმატებული უნივერსიტეტი. ეს არის უაღრესად კარგად გააზრებული მრავლისმომცველი დოკუმენტი, რომლის მომზადებაშიც უშუალოდ მონაწილეობდა 150 სპეციალისტი. გარდა ძირითადი ტექსტისა, რომელიც დაწვრილებით აღწერს სტუდენტისთვის გადასაცემი ცოდნის რაოდენობრივ და შინაარსობრივ მახასიათებლებს, შეიცავს კურიკულუმების მოდელებს ნიმუშების ჩათვლით და ორ დანართს, რომლებშიც აღიწერება სტუდენტისთვის გადასაცემი ცოდნის ცალკეული ნაწილები და შესაძლო სასწავლო კურსები.

ჩვენს მიერ აირჩა საუნივერსიტეტო კურიკულუმის მოდელი, რომელიც დაპროგრამების სწავლებას იწყებს იმპერატიული პარადიგმით, რაც ჩვენი აზრით სრულ შესაბამისობაშია საქართველოში დაპროგრამების სწავლების ტრადიციებთან.

IV. ტერმინოლოგია

ტერმინოლოგიასთან მიმართებაში მდგომარეობას ართულებს ის გარემოება, რომ ქართული ტერმინების დიდი ნაწილი შემოსულია რუსულიდან, ნაწილი ინგლისურიდან, აგრეთვე ზოგიერთი სხვა ენიდან. ამის გამო ხშირია შემთხვევა, რომ ერთი ცნების აღსანიშნავად სხვადასხვა პედაგოგი სხვადასხვა ტერმინს იყენებს. თანაც, ზოგჯერ ერთ კურსთან ერთ სემესტრში. მაგალითად, ტერმინ “**statement”**-ის ქართული შესატყვისი, ჩვენი აზრით, არის “შეტყობინება”. მის ნაცვლად დამკვიდრებული იყო ტერმინები: ბრძანება, ოპერატორი, ინსტრუქცია. ამავე დროს, ის, რასაც დედანში (ინგლისურენოვან ლიტერატურაში) ეწოდება ოპერატორი, რუსულ და ქართულ თარგმანებში, ჩვეულებრივ, მოიხსენიება, როგორც ოპერაციის ნიშანი ან ოპერაცია. კიდევ ერთი მაგალითი: ქართულ თარგმანებში პროგრამული ობიექტის (ცვლადის) შემოღების და მისთვის მეხსიერების განაწილების მოთხოვნა, ჩვეულებრივ, აღინიშნება

როგორც ამა თუ ტიპის "ცვლადის გამოცხადება" ან "ცვლადის აღწერა". მიგვაჩნია, რომ შინაარსობრივად უფრო გამართულია "ცვლადზე განაცხადის გაკეთება".

ამ პრობლემის გადაჭრის გზა არის ინგლისურ ტერმინებთან შინაარსობრივად მაქსიმალურად დაახლოებული ქართული შესატყვისის შერჩევა. რა თქმა უნდა, გარკვეულ დონეზე არსებული ტრადიციის გათვალისწინება მაინც აუცილებელია. გამონაკლისს წარმოადგენს ტერმინი "პოინტერი", რომელიც იმდენად ფესვგადგმულია ინგლისურ და უკვე რუსულ ლიტერატურაშიც, რომ მას ვიყენებთ ქართული "მიმთითებლის" პარალელურად თანაბარი დატვირთვით.

ქვემოთ მოყვანილია ზოგიერთი ჩვენს მიერ შემოთავაზებული ტერმინი:

ინგლისური ტერმინი	არსებული ქართული ტერმინი	ჩვენი შემოთავაზება
Variable declaration statement	ცვლადის გამოცხადება, ცვლადის აღწერა	განაცხადი ცვლადზე
=	მინიჭების ოპერაცია	მინიჭების ოპერატორი
Assignment statements	მინიჭების ოპერატორი	მინიჭების შეტყობინება
Branching statement	განშტოების ოპერატორი	განშტოების შეტყობინება
if statement	**if** ოპერატორი, **if** ინსტრუქცია	**if** შეტყობინება
Looping statements	ციკლის ოპერატორი	განმეორების შეტყობინება
while statement	**while** ოპერატორი, **while** ინსტრუქცია	**while** შეტყობინება
Pointer	მიმთითებელი	პოინტერი, მიმთითებელი
Header file	სათავო, სათაური ფაილი	თავსართი ფაილი

V. მეთოდიკა

საწყის ეტაპზე დაპროგრამების ენის სწავლების მეთოდიკასთან დაკავშირებით (C–ენის მაგალითზე) ჩვენ ვიზიარებთ შემდეგ პრინციპებს:

- კურსში განხილული მასალა უნდა იყოს **ლოგიკურად მწყობრი და არაწინააღმდეგობრივი**;
- მასალა აგებული უნდა იყოს გასაგებად, **მარტივიდან რთულისაკენ**, კონტიგენტის ცოდნის მარაგის და უნარ-ჩვევების შესაბამისად;
- კურსის თეორიული ნაწილი სრულიად უნდა იყოს განპირობებული **პრაქტიკული** პროგრამირების საჭიროებით;
- კურსის ფორმატი სასურველია შედგებოდეს მეცადინეობის კარგად აპრობირებული და დამკვიდრებული ფორმებისაგან: **ლექცია, პრაქტიკული და ლაბორატორიული მეცადინეობები.**

სალექციო მასალის გადმოცემის თანმიმდევრულობა დამოკიდებულია პრაქტიკული და ლაბორატორიული მეცადინეობების მასალის რიგზე. შევნიშნოთ, რომ ხშირად სწავლების მეთოდს აქვს იტერაციული ხასიათი. თუ იტერაციას განვიხილავთ, როგორც დასახული მიზნისაკენ ნაბიჯ-ნაბიჯ მიახლოების გზას, მაშინ შეიძლება

იტერაციის მეთოდი გამოყენებულ იქნეს როგორც სალექციო მასალის გადმოცემისას, ისე პრაქტიკულ-ლაბორატორიული მასალის შესრულების პროცესში. მითუმეტეს, რომ ჩვენს მიერ შეთავაზებულ პრაქტიკულ-ლაბორატორიული სამუშაოები სწორედ ამ პრინციპითაა აგებული და ეფუძნება მასალის მრავალჯერად გამეორებასა და თანდათანობით გაფართოებას ცოდნის გამყარებისა და პრაქტიკული უნარ-ჩვევების ჩამოყალიბების მიზნით.

ნებისმიერი საგნის სწავლების საწყის ეტაპზე დიდი მნიშვნელობა აქვს პედაგოგის მიერ ანალოგიებისა და ანალიზის მეთოდების გამოყენებას. მით უმეტეს, რომ დაპროგრამების ენის შესწავლა ჯერ-ჯერობით არ ხდება სკოლაში და სტუდენტისთვის მნიშვნელოვანია ახალი და უცხო ცნებების ათვისება მოხდეს მათთვის ცნობილი კატეგორიების საფუძველზე. მაგალითად, პოინტერების ახსნა არსებითად მარტივდება, თუ ოპერატიულ მეხსიერებას და რიცხვით წრფეს შორის არსებულ მსგავსებას დავიხმარებთ; სტეკის ახსნა ხდება თეფშების წყების ანალოგიით, ხოლო რიგის განმარტება მისი ყოფითი ანალოგის საფუძველზე ხდება. ასევე, ჩვენ ვუხსნით აუდიტორიას რომ C არის ენა, რადგან მას ახასიათებს ბუნებრივი ენის მთელი რიგი თვისებები: აქვს ანბანი, "სიტყვები" (იდენტიფიკატორი, რიცხვები, კონსტანტები, გასაღები სიტყვები) და "წინადადებები" (შეტყობინებები, ფუნქციები). დაპროგრამების ენის ცნებების ასახსნელად მოყვანილი მაგალითები ამყარებს კავშირს ძველსა და ახალ ცნებებს შორის.

C ენის ლექციაზე მასწავლებელი ხშირად სვამს პრობლემას და მიუთითებს ამ პრობლემის გადაჭრის სტანდარტულ გზას. პრაქტიკულ მეცადინეობაზე სტუდენტებისგან მოითხოვება სავარჯიშოების სახით მიცემული მსგავსი პრობლემების გადაწყვეტა ანალოგიური მიდგომით. ხდება სხვადასხვა სტუდენტის მოსაზრებების გარჩევა და ერთმანეთთან შეჯერება. ლაბორატორიული მეცადინეობა გამოიყენება პრაქტიკული დაპროგრამების მდგრადი უნარ–ჩვევების გამომუშავებისთვის.

ჩვენი ძირითადი შეხედულებები და იდეები რეალიზებულია C ენის ლექციათა კურსის, პრაქტიკული და ლაბორატორიული მეცადინეობების სავარჯიშოების კრებულების სახით, რომელიც ბოლო ორი წლის განმავლობაში მოვამზადეთ.

VI. მეთოდური მასალები

სწავლების მეთოდიკის ზემოთ ჩამოყალიბებული პრინციპების გათვალისწინებით შეიქმნა ლექციათა კურსი “დაპროგრამების ენა C”. თეორიული მასალა აგებულია “მარტივიდან რთულისკენ” პრინციპის დაცვით, მოყვანილია თვალსაჩინოებისათვის აუცილებელი ყველა ატრიბუტი: მრავლადაა მაგალითები, პროგრამული კოდი და ილუსტრაციები; საჭიროების შემთხვევაში, საყურადღებო ტექსტი გამოყოფილია სპეციალური ჩარჩოთი.

ეს მნიშვნელოვნად უადვილებს სტუდენტებს მასალის აღქმას და ხელს უწყობს პრაქტიკული დაპროგრამების უნარ-ჩვევების გამომუშავებას. ამავე დროს დიდი ყურადღება ეთმობა პროგრამის შექმნის სტილს, რის საფუძველზეც სტუდენტებს უნდა გამოუმუშავდეთ თანამედროვე სტანდარტების მქონე პროგრამული პროდუქტის შექმნის ჩვევები. წიგნი შედგება 13 თავისგან. დანართ 1-ში მოყვანილია შესაბამისი თემები და განსახილველი საკითხების ჩამონათვალი.

შესწავლილი თეორიული მასალის განმტკიცება ხდება პრაქტიკულ და ლაბორატორიულ მეცადინეობებზე. ყოველი თემის მიხედვით მომზადდა პრაქტიკული მეცადინეობის დიდაქტიკური მასალა: იმ ამოცანების ნაკრები, რომელთა ამოხსნას

აუცილებლად მივიჩნევთ თემის ღრმად შესასწავლად. ამოცანების გარკვეული ნაწილი სტუდენტებს ეძლევათ ამოხსნებიანად, სათანადოდ კომენტირებული და დაწერილი დაპროგრამების კარგი სტილის დაცვით. ერთ პრაქტიკულ მეცადინეობაზე ირჩევა სირთულის მიხედვით დახარისხებული 3-4 ამოცანა. ყოველი ამოხსნილი ამოცანის შემდეგ მოცემულია მსგავსი ამოცანების ჩამონათვალი დამოუკიდებლად შესრულებისათვის. თითოეულ თემაზე გათვალისწინებულია 15-20 ამოცანა. დანართ 2-ში საილუსტრაციოდ მოყვანილია ერთ-ერთი პრაქტიკული მეცადინეობის დიდაქტიკური მასალა.

ასევე მომზადდა ლაბორატორიული მეცადინეობების ჩატარების დაწვრილებითი გეგმა-დავალებები. ყოველ ლაბორატორიულ მეცადინეობაზე გათვალისწინებულია სულ ცოტა 3-4 ამოცანის გააზრება (ამოცანები ამოხსნილი სახით მიეწოდებათ სტუდენტებს), შესრულება და შედეგების გააანალიზება. ყოველი ამოცანის შემდეგ მოცემულია მასთან დაკავშირებული დავალებები. ზოგ შემთხვევაში დავალება მდგომარეობს პროგრამის ტესტირებაში სხვა საწყისი მონაცემებით, რამაც, თავის მხრივ, შეიძლება გამოიწვიოს სათანადო ტიპის შემოღების აუცილებლობა. უმეტეს შემთხვევაში დავალება მოითხოვს პროგრამის მოდიფიცირებას. ეს კი სტუდენტმა უკვე დამოუკიდებლად უნდა შესძლოს. გეგმა ითვალისწინებს საშინაო დავალებისათვის ამოცანების ჩამონათვალსაც. მაგალითისთვის იხილეთ დანართი 3, რომელშიც მოცემულია ერთ-ერთი ლაბორატორიული მეცადინეობის დიდაქტიკური მასალა.

კრიტერიუმები, რომლებსაც, ჩვენი აზრით, უნდა აკმაყოფილებდეს კარგი სახელმძღვანელო., საკმაოდ კარგადაა ცნობილი. ესენია: მასალის მოტივაცია, პრაქტიკაში გამოყენების ალბათურობა, აქცენტების გაკეთება პრაქტიკული დაპროგრამების ჩვევებზე და ა. შ. რამდენიმე გამომცემლობა, მათ შორის O'Reilly, ხელმძღვანელობს ამ კრიტერიუმებით და ბეჭდავს კარგად მოტივირებულ და წარმატებულ სახელმძღვანელოებს. ჩვენ ბევრი სასარგებლო იდეა ვნახეთ და გავიზიარეთ მათ მიერ გამოცემულ Steve Oualline-ის C ენის სახელმძღვანელოში [1]. ყურადღების გარეშე არ დაგვიტოვებია მეთოდური თვალსაზრისით ისეთი გამართული სახელმძღვანელო, როგორიცაა [2]. მაგალითების მრავალფეროვნებისთვის სასარგებლო აღმოჩნდა Stephen Prata -ს სახელმძღვანელო [3].

ლიტერატურა

1. **Practical C Programming**. 3rd Edition, By Steve Oualline, Publisher: O'Reilly, Pages 451, 1997.
2. **C How to programm,** Second Edition By H. M. Deitel - Deitel & Associates, Inc., P. J. Deitel - Deitel & Associates, Inc, Publisher: Prentice-Hall, Pages 1008, 2000.
3. **C Primer Plus**, Fifth Edition By Stephen Prata, Publisher: Sams, Pages 984, 2004.

დანართი 1.

სახელმძღვანელოში “დაპროგრამების ენა C” განხილული თემებისა და საკითხების ჩამონათვალი:

1. რას წარმოადგენს C
   - რა არის დაპროგრამება
   - როგორ მუშაობს C
   - როგორ შევისწავლოთ C
   - პროგრამები, ამოცანის დასმიდან შესრულებამდე
   - პროგრამის შექმნა ინტეგრებულ გარემოში

2. სტილი
   - რას წარმოადგენს სტილი
   - კომენტარები
   - პროგრამის კოდის წერა: გავრცელებული მიდგომები
   - სტილი თუ კულტი?
   - კოდის ფორმატირება წანაცვლების საშუალებით
   - სიცხადე და სიმარტივე

3. ძირითადი განაცხადები და გამოსახულებები
   - პროგრამის ელემენტები
   - პროგრამის ძირითადი სტრუქტურა
   - მარტივი გამოსახულებები
   - ცვლადები და მეხსიერება
   - განაცხადი ცვლადებზე
   - მთელი რიცხვები
   - მინიჭების შეტყობინება
   - მცოცავი წერტილი
   - რიცხვების გაყოფა
   - სიმბოლოები.

4. მასივები და რიცხვების წაკითხვა
   - მასივები
   - სტრიქონები
   - სტრიქონების წაკითხვა
   - მრავალგანზომილებიანი მასივები
   - რიცხვების წაკითხვა
   - ცვლადების ინიციალიზება
   - მთელი რიცხვების ტიპები
   - ნამდვილი რიცხვების ტიპები
   - მუდმივები
   - შემოკლებული ოპერატორები
   - ჩრდილოვანი ეფექტები
   - ინფორმაციის შეტანა-გამოტანის გადამისამართება. ფუნქცია `freopen`

5. მართვის შეტყობინებები

- **if** შეტყობინება
- **if-else** შეტყობინება
- როგორ არ უნდა გამოვიყენოთ **strcmp**
- განმეორების(**looping**) შეტყობინება
- **while** შეტყობინება
- **break** შეტყობინება
- **continue** შეტყობინება
- გვერდითი ეფექტები

6. სხვა მმართველი შეტყობინებები
   - **for** შეტყობინება
   - **switch** შეტყობინება
   - **switch, break** და **continue**
7. ცვლადების ხილვადობის არე და ფუნქციები
   - ხილვადობის არე და მეხსიერების კლასი
   - სტეკი
   - ფუნქციები
   - უპარამეტრო ფუნქციები
   - სტრუქტურული პროგრამირება
   - რეკურსია
8. C-ს პრეპროცესორი
   - #define შეტყობინება
   - პირობითი კომპილაცია
   - include ფაილები
   - პარამეტრიანი მაკროსი
9. სტრუქტურები, გაერთიანებები და ზოგიერთი სხვა ტიპი
   - სტრუქტურები
   - გაერთიანებები (**Unions**)
   - ტიპის განმსაზღვრელი **typedef**
   - ტიპი **enum** (ჩამონათვალი)
   - ტიპის გარდაქმნა (**typecast**)
   - ბიტური ველები, შეკუმშული (packed) სტრუქტურები
   - სტრუქტურების მასივები
   - რეზიუმე
10. პოინტერები
    - მისამართები. მოქმედებები მისამართებზე
    - პოინტერი. პოინტერის გამოყენება ფუნქციის არგუმენტად
    - მუდმივი პოინტერი
    - პოინტერები და მასივები
    - პოინტერის გამოყენება სტრიქონის გასახლეჩად
    - პოინტერები და სტრუქტურები
11. მცოცავწერტილიანი რიცხვები
    - მცოცავწერტილიანი ფორმატი
    - შეკრება /გამოკლება ნამდვილ რიცხვებზე
    - გამრავლება

- გაყოფა
- გადავსება
- დამრგვალების ცდომილება
- სიზუსტე
- დამრგვალების ცდომილების მინიმიზება
- სიზუსტის განსაზღვრა
- სიზუსტე და სისწრაფე

12. შეტანა / გამოტანა ფაილებში

- ფაილის გახსნა და დახურვა
- სიმბოლოების და სტრიქონების შეტანა/გამოტანა ფაილებში
- გარდაქმნის ფუნქციები
- ბინარული და ASCII ფაილები
- ბინარული შეტანა/გამოტანა

13. მეხსიერების დინამიკური მართვა

- ერთგანზომილებიანი მასივისთვის მეხსიერების დინამიკური გამოყოფა და გათავისუფლება
- არამართკუთხა ფორმის ორგანზომილებიანი მასივების შექმნა მეხსიერების დინამიკური მართვის საშუალებების გამოყენებით
- ორგანზომილებიანი მასივისთვის მეხსიერების დინამიკური გამოყოფა და გათავისუფლება
- ავტორეფერატული სტრუქტურები
- ბმული სიები

დანართი 2.

პრაქტიკული მეცადინეობა №12

გამოყენებული ტიპები: სტრუქტურების მასივი. სტრუქტურის ტიპზე პოინტერი.

განსახილველი საკითხები: სტრუქტურების მასივის ინიციალიზება, მასივის ელემენტების ბეჭდვა. მუშაობა სტრუქტურების მასივთან: ელემენტის ძიება ველის (ველების) მნიშვნელობის მიხედვით, ელემენტის პოვნა ერთ-ერთი ველის მაქსიმალური (მინიმალური) მნიშვნელობის მიხედვით.

მოკლე მიმოხილვა:

მასივი შეიძლება შეიცავდეს რთული ტიპის ელემენტებს. კერძოდ, მისი ელემენტი შეიძლება იყოს სტრუქტურის ტიპის. მაგალითად, განვიხილოთ სტრუქტურის ტიპი

```
typedef struct{
    int n;
    char c;
    float f;
    char s[30];
} info;
```

info ტიპის N ელემენტის შემცველ array მასივზე გამაცხადს აქვს სახე:

```
info array[N];
```

array მასივის i- ურ ელემენტზე მიმართვა ხდება array[i] ინდექსირებული ცვლადის გამოყენებით. ყოველი array[i] წარმოადგენს info ტიპის სტრუქტურას, ამიტომ მის ველებზე წვდომისთვის უნდა ვისარგებლოთ ოპერატორით " . " მაგალითად, შემდეგი ფრაგმენტი ჩაწერს მასივის ელემენტებში კლავიატურიდან შემოტანილ მონაცემებს:

```
int i;
for(i=0; i<N; i++)
    scanf("%d%c%f%s",
&array[i].n, &array[i].c, &array[i].f, array[i].s);
```

სტრუქტურების მასივის ელემენტზე წვდომისთვის შეგვიძლია გამოვიყენოთ პოინტერი. ჩვენს შემთხვევაში პოინტერზე განაცხადს ექნება სახე:

```
info *ptr;
```

მაშინ array[i] ელემენტზე წვდომისთვის ptr–ს უნდა მივანიჭოთ array[i]-ის მისამართი. მაგალითად,

```
int i = 3;
ptr = &array[i];
```

i- ური სტრუქტურის ველებზე მიმართვა ხდება " -> " ოპერატორის მეშვეობით, მაგალითად,

```
ptr -> array[i].n;
```

ვთქვათ, უნდა ვიპოვოთ array მასივის ის ელემენტი, რომლის c ველის მნიშვნელობა დანარჩენი ელემენტების c ველთან შედარებით უდიდესია. ამოცანა გადავწყვიტოთ პოინტერის დახმარებით:

```
int i;
info *max_ptr;
max_ptr = &array[0];
for(i=1; i<N; i++)
    if(array[i].c > max_ptr -> c) max_ptr = &array[i];
```

ამოხსნის ეს ტექნიკა ამცირებს პროგრამის შესრულების დროს მსგავსი ამოცანების გადაწყვეტის სხვა ხერხებთან შედარებით. დროის შემცირება მით უფრო შესამჩნევი ხდება, რაც უფრო დიდია მასივის განზომილება და სტრუქტურის მოცულობა.

პრაქტიკული მეცადინეობის გეგმა:

1. გვაქვს ინფორმაცია 3 სტუდენტის შესახებ: გვარი, შეფასება ქულებში. დაწერეთ პროგრამა, რომელიც დაბეჭდავს უმაღლესი შეფასების მქონე სტუდენტის გვარს. მონაცემები შეიტანეთ კლავიატურიდან – თითო სტრიქონში თითო სტუდენტის გვარი და შეფასება.

ამოხსნა.

```
#include <stdio.h>
#include <stdlib.h>
typedef struct {
    char gvari[15];
    int shepaseba;
}student;
int main ()
{
    student one, two, three;
    student * max;
    scanf("%s%d", one.gvari, &one.shepaseba);
    scanf("%s%d", two.gvari, &two.shepaseba);
    scanf("%s%d", three.gvari, &three.shepaseba);
    max=&one;
    if(two.shepaseba > max->shepaseba) max=&two;
    if(three.shepaseba > max->shepaseba) max=&three;
    printf("Yvelaze maRali shepaseba %d aqvs students %s\n",
            max->shepaseba, max->gvari);
    system("PAUSE");
    return 0;
}
```

2. ცხრილი 1-ში მოცემულია ინფორმაცია საქართველოს მწვერვალების შესახებ. ტექსტური ფაილის თითოეულ სტრიქონში ცხრილის შესაბამისად ჩაწერეთ მწვერვალის დასახელება და მისი სიმაღლე ზღვის დონიდან. დაწერეთ პროგრამა, რომელიც დაადგენს და დაბეჭდავს ყველაზე მაღალი მწვერვალის დასახელებასა და სიმაღლეს.

ცხრილი 1

Mწვერვალის დასახელება	სიმაღლე ზღვის დონიდან (მ)
შხარა	5068
მყინვარწვერი	5047
დიკლოს მთა	4285
აჭარა	2908
საყორნია	3280
ცივი	1991
უშბა	4695
რუსთაველის პიკი	4622
ფსისი	3790

ამოხსნა.

```
#include <stdio.h>
#include <conio.h>
typedef struct {
    char saxeli[15];
    int simagle;
} mwvervali;
const int N=9;
int main ()
{
    mwvervali jgupi[N];
    mwvervali * max;
    int i;
    freopen("mwvervali.txt", "r", stdin);
    for(i=0; i<N; i++)
    scanf("%s%d", jgupi[i].saxeli, &jgupi[i].simagle);
    max=&jgupi[0];
    for(i=1; i<N; i++)
        if(jgupi[i].simagle>max->simagle) max=&jgupi[i];
    printf("Yvelaze maRali mwvervalea %s simagliT %d\n",
           max->saxeli, max->simagle);
    getch();
    return 0;
}
```

3. ტექსტურ ფაილში მოცემულია დედამიწის ცნობილი კუნძულების ჩამონათვალი – კუნძულის დასახელება და მდებარეობა (ცხრილი 2-ის თანახმად). დაწერეთ პროგრამა, რომელიც დაბეჭდავს იმ კუნძულების დასახელებას, რომლებიც მდებარეობენ:

ა) აზიაში; ბ) ევროპაში; გ) ჩრდილოეთ ამერიკაში.

ცხრილი 2

კუნძულის დასახელ.	მდებარეობა	კუნძულის დასახელ.	მდებარეობა
გრენლანდია	ჩრდილ. ამერიკა	კუბა	ჩრდილ. ამერიკა
ახალი გვინეა	ოკეანია	ისლანდია	ევროპა
სუმატრა	აზია	ირლანდია	ევროპა
დიდი ბრიტანეთი	ევროპა	ჰოკაიდო	აზია
ჰონსიუ	აზია	ჰაიტი	ჩრდილ. ამერიკა
იავა	აზია	სახალინი	აზია
ახალი ზელანდია	ოკეანია	მადაგასკარი	აფრიკა

4. ტექსტურ ფაილში წერია მონაცემები მზის სისტემის პლანეტების შესახებ – პლანეტის დასახელება, დიამეტრი, მანძილი მზემდე და ორბიტული სიჩქარე (იხილეთ ცხრილი 3). დაწერეთ პროგრამა, რომელიც დაბეჭდავს:

ა) პლანეტა ურანის ყველა მონაცემს;

ბ) მზისგან ყველაზე დაშორებული პლანეტის დასახელებას;

გ) ყველაზე მცირე დიამეტრის მქონე პლანეტის მონაცემებს;

დ) ყველაზე დიდი ორბიტული სიჩქარის მქონე პლანეტის დიამეტრს.

ცხრილი 3

პლანეტა	დიამეტრი(კმ)	მანძილი მზემდე (მლნ კმ)	ორბიტული სიჩქარე (კმ/წმ)	პლანეტა	დიამეტრი (კმ)	მანძილი მზემდე (მლნ კმ)	ორბიტული სიჩქარე (კმ/წმ)
მერკური	5000	58	48	სატურნი	120000	1500	7
ვენერა	12400	108	30	ურანი	51000	2900	5
დედამიწა	12700	150	24	ნეპტუნი	50000	4500	5
მარსი	6800	228	13	პლუტონი	18000	5910	4
იუპიტერი	140000	778	10	-	-	-	-

5. ტექსტურ ფაილში მოცემულია ინფორმაცია საქართველოს მდინარეების შესახებ – მდინარის დასახელება და სიგრძე (იხილეთ ცხრილი 4). შეადგინეთ ფუნქცია, რომელიც დაადგენს და დაბეჭდავს:

ა) 200 კმ-ზე მეტი სიგრძის მქონე მდინარეების რაოდენობას;

ბ) ყველაზე მოკლე მდინარის სიგრძეს;

გ) ყველაზე გრძელი მდინარის დასახელებას;

დ) 200 კმ-ზე ნაკლები სიგრძის მქონე მდინარეების შესახებ სრულ ინფორმაციას.

ფუნქცია გამოიყენეთ ძირითად პროგრამაში.

ცხრილი 4

მდინარე	სიგრძე (კმ)	მდინარე	სიგრძე (კმ)
მტკვარი	1515	ცხენისწყალი	184
ალაზანი	407	იორი	351
რიონი	333	სუფსა	117
ენგური	206	არაგვი	112
ხრამი	220	-	-

6. ტექსტურ ფაილში მოცემულია მსოფლიოს ზოგიერთი მდინარის შესახებ მონაცემები – მდინარის დასახელება და მისი სიგრძე (იხილეთ ცხრილი 5). დაწერეთ ფუნქცია, რომელიც დაადგენს და დააბრუნებს:

ა) ყველაზე მოკლე მდინარის სიგრძეს;

ბ) იმ მდინარეთა რაოდენიბას, რომელთა სიგრძე 600 კმ-ზე მეტია;

გ) იმ მდინარეთა რაოდენიბას, რომელთა სიგრძე 600 კმ-ზე ნაკლებია.

გამოიძახეთ ფუნქცია ძირითად პროგრამაში და დაბეჭდეთ შედეგი.

ცხრილი 5

მდინარე	სიგრძე (კმ)	მდინარე	სიგრძე (კმ)
ამაზონი	6500	ნილოსი	6700
კონგო	4600	ობი	5600
იანცზი	5500	ნიგერი	4200
მისისიპი	6200	-	-

7. დედამიწის სხვადასხვა კუნძულზე მოქმედი ვულკანების რაოდენობა მოცემულია ცხრილ 6-ში. ჩაწერეთ ეს ინფორმაცია ტექსტურ ფაილში სტრიქონებად. შეადგინეთ ფუნქცია, რომელიც:

ა) დაითვლის და დააბრუნებს იმ კუნძულების რაოდენობას, რომლებზეც მოქმედი ვულკანების რიცხვი 20-ზე მეტია;

ბ) დაითვლის და დააბრუნებს იმ კუნძულების რაოდენობას, რომლებზეც მოქმედი ვულკანების რიცხვი 18-ზე ნაკლებია;

გ) დაბეჭდავს იმ კუნძულების დასახელებებს, რომლებზეც მოქმედი ვულკანების რიცხვი ეკუთვნის [3, 15] შუალედს.

გამოიყენეთ ფუნქცია ძირითად პროგრამაში.

ცხრილი 6

კუნძული	მოქმედი ვულკანების რაოდენობა	კუნძული	მოქმედი ვულკანების რაოდენობა
კამჩატკა	22	ახალი გვინეა	10
კურილიის კუნძულები	38	ალუეტის კუნძულები	17
იაპონია	58	ჰავაის კუნძულები	4
ფილიპინები	12	აზორის კუნძულები	9
ახალი ზელანდია	5	ისლანდია	26

8. საქართველოს ტბების შესახებ ინფორმაცია მოცემულია ცხრილი 7-ში. ჩაწერეთ იგი ტექსტურ ფაილში. შეადგინეთ ფუნქცია, რომელიც:

ა) დაბეჭდავს მაქსიმალური სიღრმის მქონე ტბის დასახელებას;

ბ) დააბრუნებს უმცირესი ზედაპირის ფართობის მქონე ტბის სიღრმეს;

გ) დაბეჭდავს უდიდესი ზედაპირის ფართობის მქონე ტბის შესახებ სრულ ინფორმაციას;

დ) დააბრუნებს მინიმალური სიღრმის მქონე ტბის ზედაპირის ფართობს;

ე) დაბეჭდავს იმ ტბის დასახელებას, რომლის სიღრმე ეკუთვნის [5, 20] შუალედს.

გამოიყენეთ ფუნქცია ძირითად პროგრამაში.

ცხრილი 7

ტბა	ზედაპირის ფართ., კმ^2	სიმაღლე ზღვის დონ., მ	მაქსიმ. სიღრმე, მ	ტბა	ზედაპირის ფართ., კმ^2	სიმაღლე ზღვის დონ., მ	მაქსიმ. სიღრმე, მ
ფარავანი	37.5	2073	3.3	ჯანდარი	10.6	291	7.2
პალიასტომი	18.2	-0.3	3.2	რიწა	1.49	884	101
ტაბაწყური	14.2	1997	40.2	ბაზალეთი	1.22	879	7

9. მე-8 ცხრილის საფუძველზე შექმენით ტექსტური ფაილი "ზგვები.ტხტ", რომლის თითოეულ სტრიქონში მოათავსეთ ატლანტის ოკეანის თითოეული ზღვის მონაცემები – დასახელება, ფართობი და უდიდესი სიღრმე. დაწერეთ ფუნქცია, რომელიც:

ა) დაბეჭდავს ყველა იმ ზღვის შესახებ სრულ ინფორმაციას, რომლის ფართობი ეკუთვნის [1000, 1500] შუალედს, ხოლო უდიდესი სიღრმე ეკუთვნის [3000, 5000] შუალედს;

ბ) დაბეჭდავს ყველა იმ ზღვის შესახებ სრულ ინფორმაციას, რომლის უდიდესი სიღრმე არ აღემატება 500-ს;

გ) დააბრუნებს ყველა იმ ზღვის რაოდენობას, რომლის ფართობი 544-ზე არანაკლებია;

დ) დააბრუნებს ყველა იმ ზღვის რაოდენობას, რომლის ფართობი ეკუთვნის [200, 600] შუალედს, ხოლო უდიდესი სიღრმე ეკუთვნის [250, 450] შუალედს.

ცხრილი 8. ატლანტის ოკეანის ზღვები

დასახელება	ფართობი, 1000 კვ. კმ	Uუდიდესი სიღრმე
ხმელთაშუა ზღვა	2505	5421
მარმარილოს ზღვა	11	1355
ნორვეგიის ზღვა	1383	3860
კარიბის ზღვა	2754	768
შავი ზღვა	423	2938
გრენლანდიის ზღვა	1205	4846
ჩრდილოეთის ზღვა	544	433
ბალტიის ზღვა	386	459
ირლანდიის ზღვა	210	272

ამოხსნა.

```
#include <stdio.h>
#include <conio.h>
typedef struct {
    char saxeli[15];
    int partobi;
    int sigrme;
} zgva;
int get_number(zgva a[], int raod);
int main ()
{
   const int K=9;
   zgva array[K];
   int i;
   int raodenoba;
   freopen("zgvebi.txt", "r", stdin);
   for(i=0; i<K; i++)
      scanf("%s%d%d",
            array[i].saxeli, &array[i].partobi, &array[i].sigrme);
   raodenoba=get_number(array, K);
   printf("AseTi zgvebis raodenoba=%d\n", raodenoba);
   getch();
   return 0;
}
int get_number(zgva a[], int raod){
   int i;
   int r=0;
   for(i=0; i<raod; i++)
      if(a[i].partobi>=200 && a[i].partobi<=600)
          if(a[i].sigrme>=250 && a[i].sigrme<=450) r++;
   return r;
}
```

10. დედამიწის ვულკანების მონაცემები გაერთიანებულია ცხრილი 9-ში. შეიტანეთ ეს ინფორმაცია ტექსტურ ფაილში. დაწერეთ ფუნქცია, რომელიც დაბეჭდავს:

ა) 3500 მ-ზე ნაკლები სიმაღლის ყველა ვულკანის დასახელებასა და მდებარეობას;

ბ) იტალიის ვულკანებიდან ყველაზე მაღალი ვულკანის მდებარეობას;

გ) იტალიის ვულკანების დასახელებებსა და სიმაღლეებს.

გამოიყენეთ ფუნქცია ძირითად პროგრამაში.

ცხრილი 9

დასახელება	სიმაღლე (მ)	ადგილმდებარეობა
ეტნა	3340	იტალია
ჰეკლა	1491	ისლანდია
ვეზუვი	1277	იტალია
კლუჩევსკაია სოპკა	4750	კამჩატკა
ფუძიამა	3776	იაპონია
კრაკატაუ	813	აზია
კამერუნი	4070	კამერუნი
სტრომბოლი	926	იტალია
პოპოკატეპეტლი	5452	მექსიკა

11. ტექსტურ ფაილში წერია მსოფლიოს ოკეანეების შესახებ მონაცემები – ოკეანეს დასახელება და მისი ფართობი (იხილეთ ცხრილი). დაწერეთ ფუნქცია, რომელიც:

ა) დაადგენს და დააბრუნებს ოკეანეების საშუალო ფართობს;

ბ) დაბეჭდავს იმ ოკეანეს დასახელებას, რომლის ფართობი ეკუთვნის [20, 80] შუალედს;

გ) დაადგენს და დააბრუნებს ატლანტის ოკეანეს ფართობს.

გამოიძახეთ ფუნქცია ძირითად პროგრამაში და დაბეჭდეთ შედეგი. სათანადო მონაცემთა ცხრილი იხილეთ ლაბორატორიული მეცადინეობა ¹ 12 –ის მე-2 ამოცანაში.

დანართი 3.

ლაბორატორიული მეცადინეობა №4

ლაბორატორიული სამუშაოს აღწერა: კლავიატურიდან (ინფორმაციის შეტანის სტანდარტული ნაკადიდან _ stdin) რიცხვების, სტრიქონების და სიმბოლოების წაკითხვა.

მასალის ათვისების წინაპირობები:

1. წინა დავალების შემოწმება და გარჩევა.
2. ფუნქცია scanf.

მეცადინეობის გეგმა:

1. **ამოცანა**: შემდეგი პროგრამის:

```
/*********************************************************
 * ავტორი:  თამარ ანთიძე                                *
 * პროგრამა: კლავიატურიდან მთელი რიცხვის წაკითხვა  *
 *             და შემდეგ მისი გამობეჭდვა                *
 *********************************************************/
#include <stdio.h>
#include <stdlib.h>
int main()
{  int k;
    printf("Enter number: ");
    scanf("%d", &k);
    printf("The number is: %d\n",k);
    system("pause");
    return 0;
}
```

გაშვების შემდეგ, შესატანი რიცხვი აკრიფეთ რამდენიმენაირად, წინ ჰარებით, ან ‘\n’ სიმბოლოებით, ან ორივეთი ერთად და გააკეთეთ დასკვნები. მაგალითად, შეტანის ეკრანს შეიძლება ჰქონდეს სახე:

```
Enter number:
                 ან
    543
```

```
Enter number:
     543
                 ან
```

```
Enter number: 543
```

2. **ამოცანა**: შემდეგი პროგრამის:

```
/************************************************
 * ავტორი:  თამარ ანთიძე                       *
 * პროგრამა: კლავიატურიდან სტრიქონის წაკითხვა scanf-ით *
 *             და შემდეგ მისი გამობეჭდვა       *
 ************************************************/
#include <stdio.h>
#include <stdlib.h>
int main()
{  char str[50];
    printf("Enter string: ");
    scanf("%s", str );
    printf("string \"%s\" is entered\n",str);
    system("pause");
    return 0;
```

```
}
```

გაშვების შემდეგ, შესატანი რიცხვი აკრიფეთ რამდენიმენაირად, წინ ჰარებით, ან ‘\n’ სიმბოლოებით, ან ორივეთი ერთად და გააკეთეთ დასკვნები. მაგალითად, შეტანის ეკრანს შეიძლება ჰქონდეს სახე:

```
Enter string:
       NewTopic
```

ან

```
Enter string:
    NewTopic
```

```
Enter string: NewTopic
```

3. შემდეგი პროგრამა

```
/*******************************************************
 * ავტორი:  თამარ ანთიძე                              *
* პროგრამა: დიალოგურ რეჟიმში კლავიატურიდან           *
 *           მიმდევრობით ნამდვილი რიცხვისა და სიმბოლოს      *
 *           შეყვანა და შემდეგ მათი დაბეჭდვა                 *
*******************************************************/
 #include <stdio.h>
 #include <stdlib.h>
 int main()
 {
     char c;
     int i;
     printf("Enter int: ");
     scanf("%d", &i );
     printf("Enter char: ");
     scanf("%c", &c );
     printf("Entered int \"%d\" and char \"%c\"\n", i, c);
     system("pause");
     return 0;
 }
```

რატომღაც არ ბეჭდავს შეყვანილ მონაცემებს. როგორ გამოვასწოროთ მდგომარეობა?

მითითება: მთელი რიცხვის შესაყვანად ვაჭერთ Enter ღილაკს, რომელსაც შეესაბამება '\n' (ახალი ხაზის)სიმბოლო. ეს სიმბოლო რჩება ბუფერში და ავტომატურად ებმება მომდევნო scanf ფუნქციას. ერთ-ერთი გამოსავალი ისაა, რომ scanf("%d", &i); შეტყობინების მერე ჩავამატოთ getchar შეტყობინება, რომელიც ბუფერიდან ამოიღებს “დაუპატიჟებელ” სიმბოლოს.

4. რა დაიბეჭდება პროგრამული კოდის შემდეგი ფრაგმენტის შესრულების შედეგად:

```
{
   int i;
   char c;
   printf("Enter int and char: ");
   scanf("%d%c", &i,&c );
   printf("Entered int \"%d\" and char \"%c\"\n", i, c);
   system("pause");
   return 0;
}
```

თუ კლავიატურიდან შევიყვანთ შემდეგ სტრიქონს:

ა) 543a ბ) 253 a გ) g53 დ) 23 943

სტილისტური მითითება: გაურკვევლობის თავიდან ასაცილებლად, ზოგჯერ უმჯობესია ცხადად მივუთითოთ თუ როგორი ტიპის მონაცემების შეტანას ველოდებით. მაგალითად, შეგვიძლია კოდში ჩავსვათ უფრო ვრცელი შეტყობინება:

```
printf("Enter int and char (for example 345f): ");
```

დავალება:

1. რა დაიბეჭდება კოდის შემდეგი ფრაგმენტის შესრულების შედეგად?

ა)
```
puts("swavla");
puts("swavla");
```

ბ)
```
printf("swavla");
printf("swavla");
```

გ)
```
printf("swavla\n");
printf("swavla");
```

2. შეადგინეთ პროგრამა, რომელიც კლავიატურიდან მიიღებს სიმბოლოს და დაბეჭდავს მის ASCII (American Standard Code for Information Interchange) კოდს (მითითება: ისარგებლეთ წინა ლაბორატორიულის მასალით).

3. შეადგინეთ პროგრამა, რომელიც კლავიატურიდან მიიღებს ორ სიმბოლოს და დაბეჭდავს მათ შორის უდიდესი ASCII კოდის მქონეს (მითითება: ისარგებლეთ წინა ლაბორატორიულის მასალით).

4. შეადგინეთ პროგრამა, რომელიც კლავიატურიდან მიიღებს ორ ნამდვილ რიცხვს და დაბეჭდავს მათ ჯამს.

5. შეადგინეთ პროგრამა, რომელიც კლავიატურიდან მიიღებს სამ ნამდვილ რიცხვს და დაბეჭდავს მათ შორის უდიდესს.

6. შეადგინეთ პროგრამა, რომელიც კლავიატურიდან მიიღებს სამ მთელ რიცხვს და დაბეჭდავს მათ საშუალო არითმეტიკულს.

7. შეადგინეთ პროგრამა, რომელიც კლავიატურიდან მიიღებს ორ სტრიქონს და დაბეჭდავს მათი სიგრძეების ჯამს.

სტატია შემოსულია: 2009-02-12