



ზვიად ალიევი

აზერბაიჯანელი ხალხის საერთო
ეროვნული ლიდერი

Çap için değil

რამინ მაჰმუდზადე, ისმაილ სადიგოვი, ნაილა ისაევა

ინფორმატიკა 9

სახელმძღვანელო

ზოგადსაგანმანათლებლო სკოლების მე-9 კლასის
მოსწავლეთათვის

დამტკიცებულია აზერბაიჯანის რესპუბლიკის
განათლების სამინისტროს მიერ
(ბრძანება №369, 03.06. 2016)

© აზერბაიჯანის რესპუბლიკის განათლების სამინისტრო – 2017.

საავტორო უფლებები დაცულია. ამ გამოცემის ან მისი რომელიმე ნაწილის
გადაბეჭდვა, კოპირება და ინფორმაციის ელექტრონული საშუალებებით გავრცელება
სპეციალური ნებართვის გარეშე კანონსაწინააღმდეგოა.

გამომცემლობა „Yaz“

სახელმძღვანელოსთან დაკავშირებული მოსაზრებები, შენიშვნები და წინადადებები
გთხოვთ, გამოგვიგზავნოთ შემდეგ ელექტრონულ მისამართებზე:

yazneshr@mail.ru და derslik@edu.gov.az.

წინასწარ გიხდით მადლობას თანამშრომლობისათვის!

ინფორმატიკა

შინაარსი

1. კოდირება

1. რასტრული გრაფიკა.....	8
2. ვექტორული გრაფიკა	12
3. ვექტორულ რედაქტორში მუშაობა	16
4. გრაფიკული ინფორმაციის კოდირება.....	19
5. ხმოვანი ინფორმაციის კოდირება.....	22
შემაჯამებელი კითხვები და დავალებები	26

2. კომპიუტარი

6. კომპიუტერის ძირითადი მახასიათებლები	28
7. მართვის პანელი	32
8. დამხმარე პროგრამები. დეფრაგმენტაცია	35
9. დამხმარე პროგრამები. დისკის განმენდა	38
შემაჯამებელი კითხვები და დავალებები	42

3. გამოყენებითი პროგრამები

10. რთულ დოკუმენტებზე მუშაობა	44
11. მულტიმედიური პრეზენტაციები	48
12. ელექტრონულ ცხრილებზე მუშაობა.....	52
13. დიაგრამები ელექტრონულ ცხრილებში	56
შემაჯამებელი კითხვები და დავალებები	60

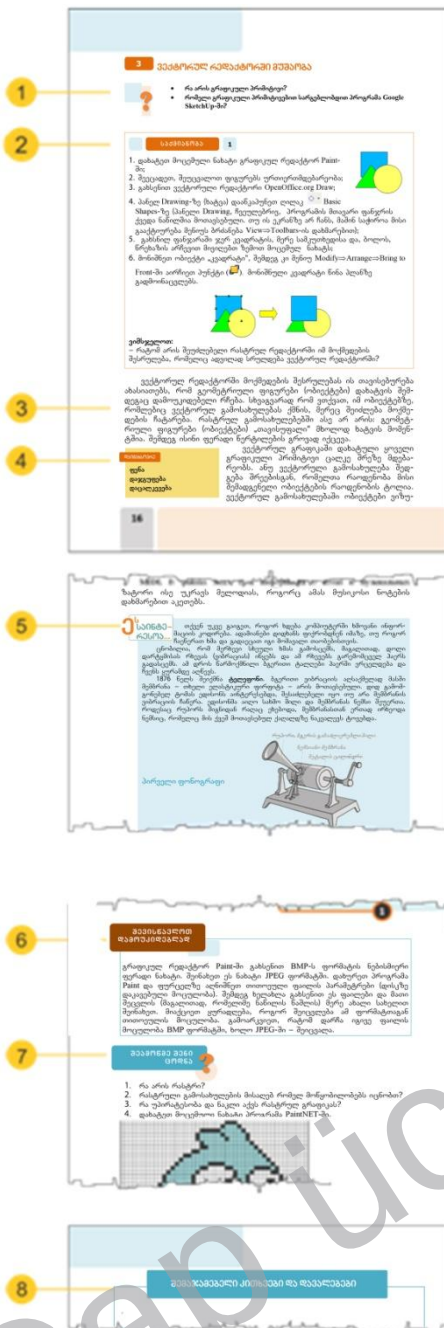
4. დაპროგრამება

14. რიცხვებზე მუშაობა.....	62
15. სტრიქონები.....	66
16. სიები.....	71
17. ფუნქცია	77
18. დაპროგრამების ენების კლასიფიკაცია	81
შემაჯამებელი კითხვები და დავალებები	84

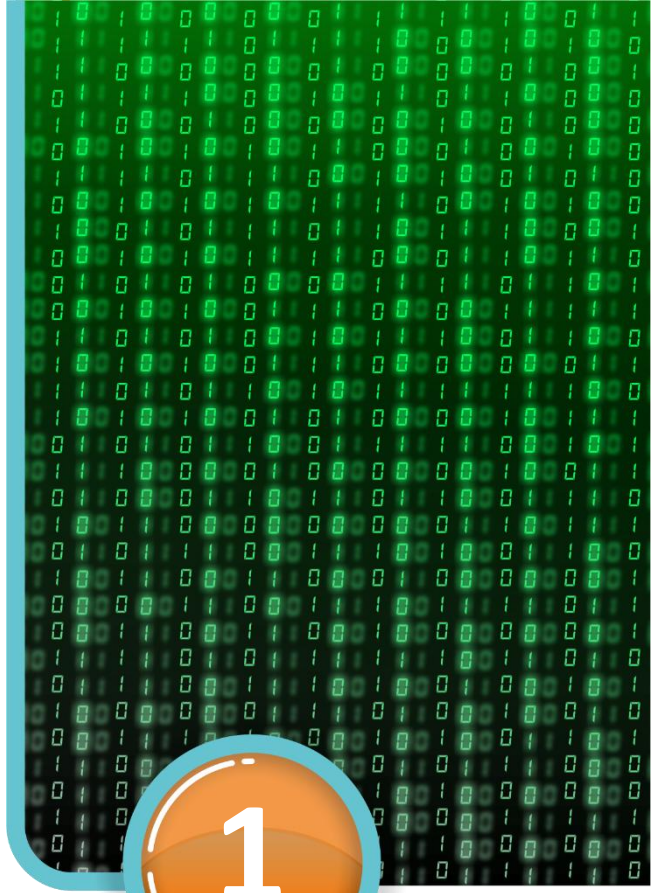
5. ინფორმაციული ტექნოლოგიები

19. ინფორმაციული მოდელი გრაფებში	86
20. გრაფებთან დაკავშირებული ამოცანები	91
21. კომპიუტერული ქსელები.....	96
22. ინფორმაციის გადაცემა.....	100
23. ვებსაიტების შექმნა	104
24. საინფორმაციო საზოგადოებისადმი მიძღვნილი საიტის შედგენა ...	108
25. დამისამართება ინტერნეტში	113
შემაჯამებელი კითხვები და დავალებები	116
ცნობილი ალგორითმები	117
ტერმინების ლექსიკონი	124

გავეცნოთ სახელმკვანელოს



- მოტივაცია.** სხვადასხვა სიტუაციისა და მოვლენის აღწერა, რომელიც შეკითხვების დასმით სრულდება.
- საქმიანობა.** კვლევითი ხასიათის დავალებები, რომელთა მიზანია საინტერესო მოვლენებისა და პროცესების შესწავლა, მათ შორის არსებული მიზეზშედეგობრივი კავშირის გამოვლენა. შესრულებულ სამუშაოზე სამსჯელო და შესაძლო შეცდომების გამოსავლენად დასმული შეკითხვები.
- დამიხსოვრე.** ჩამოთვლილია ყველა ძირითადი ცნება, რომელიც კონკრეტული თემის ფარგლებში ისწავლება.
- განმარტებები.** გაკვეთილის ძირითადი შემადგენელი ნაწილი: ახალი ცნებები, განმარტებები, განსაზღვრებები და ნესები.
- ეს საინტერესოა.** მაგალითების ნიმუშები და საინტერესო ინფორმაცია შეძენილი ცოდნის გასაღრმავებლად.
- შევისწავლოთ დამოუკიდებლად.** დავალებები დამოუკიდებელი მუშაობისა და შეძენილი ცოდნის გამოყენებისათვის.
- შეამოწმე შენი ცოდნა.** თემის ფარგლებში შესწავლილი მასალის განსამტკიცებლად და სწავლის სუსტი მხარეების გამოსავლენად გამიზნული კითხვები და დავალებები.
- შემაჯამებელი კითხვები და დავალებები.** შემაჯამებელი ხასიათის კითხვები და დავალებები ცალკეული თემების მიხედვით.



1

კოდირება

1. რასტრული გრაფიკა
2. ვექტორული გრაფიკა
3. ვექტორულ რედაქტორში მუშაობა
4. გრაფიკული ინფორმაციის კოდირება
5. ხმოვანი ინფორმაციის კოდირება

1

რასტრული გრაფიკა



- რომელ გრაფიკულ რედაქტორებს იცნობთ?
- თქვენი აზრით, რა უპირატესობა და ნაკლი აქვს გრაფიკულ რედაქტორ Paint-ს?

სამუშაო

1

1. გახსენით ნებისმიერი გრაფიკული ფაილი გრაფიკულ რედაქტორ Paint-ში;
2. აირჩიეთ ინსტრუმენტი  Magnifier (მასშტაბი). მაჩვენებელი ამ ინსტრუმენტის სახეს მიიღებს;
3. გადაადგილეთ მაჩვენებელი სამუშაო ველში და დაანკაპუნეთ თავის მარცხენა ლილაკზე. ნახატის ზომები 2-ჯერ გადიდება (ნახატის შესამცირებლად თავის მარჯვენა ლილაკით უნდა ისარგებლოთ);
4. რამდენჯერმე გაიმეორეთ ეს ქმედება.

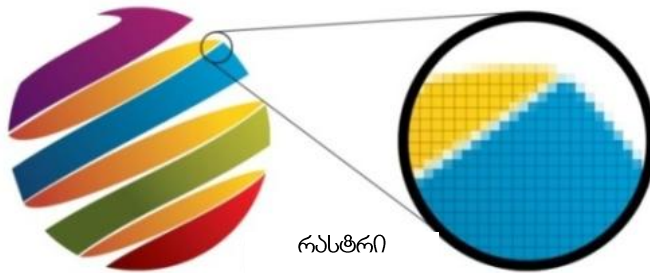


ვიმსჯელოთ:

– რა შეიძლება ითქვას გამოსახულების შესახებ მის გადიდებაზე დაკვირვების შემდეგ?

თქვენ უკვე იცით, რომ კომპიუტერის ეკრანზე გამოსახულება პატარა-პატარა ნერტილების – **პიქსელების** – ერთობლიობიდან შედგება. იმ ნერტილების მართკუთხა ბადეს, რომლებიც გამოსახულებას ქმნის,

რასტრი ეწოდება (ტერმინი „რასტრი“ მომდინარეობს ლათინური „rastrum“-იდან, რომელიც ფოცხს ნიშნავს). გამოთვლითი ტექნიკის დახმარებით შექმნილ და კომპიუტერის მეხსიერებაში ეკრანის პიქსელების შესაბამისი ბიტების ნაკრების სახით შენახულ გამოსახულებას **რასტრულ გრაფიკას** უწოდებენ.



ნერტილების ერთობლიობა ეკრანზე გამოსახულებას ქმნის, მაგრამ თუ რასტრულ გამოსახულებას ძალიან გავადიდებთ, ის პატარ-პატარა უჯრებისგან შედგენილ მოზაიკას (ბადეს) დაემსგავსება. რასტრის ყოველი ნერტილი ორი პარამეტრით ხასიათდება: ეკრანზე მდებარეობითა და ფერით (თუ მონიტორი ფერადია) ან სიკაშკაშის ხარისხით (თუ მონიტორი შავ-თეთრია).

რასტრული გამოსახულებების შესაქმნელად და მათზე სამუშაოდ ბევრი სხვადასხვა პროგრამა არსებობს. ოპერაციული სისტემა Windows-ის პროგრამა **Microsoft Paint**-ს თქვენ ქვედა კლასებიდანვე იცნობთ. ორგანზომილებიანი ნახატების შესაქმნელად უფრო ფართო შესაძლებლობების მქონე პროგრამების, მაგალითად, **PaintNET**-ის, **Corel Painter**-ის, **Adobe PhotoShop**-ის ან **Corel PHOTO-PAINT**-ის გამოყენება შეიძლება. რასტრულ გამოსახულებებზე სამუშაოდ თავისუფლად გავრცელებადი (უფასო) პროგრამებიც არსებობს. მათ შორისაა **PaintNET, GIMP, TuxPaint, StarOffice Image**.



paint.net



Adobe PhotoShop



GIMP



TuxPaint

მიუხედავად იმისა, რომ რასტრული გამოსახულება პიქსელების ნაკრებისგან შედგება, ჩვეულებრივ, ცალკეულ პიქსელებზე კი არა, მთელ ფრაგმენტზე მუშაობენ. ამისათვის გრაფიკულ რედაქტორში სხვადასხვა ინსტრუმენტი – სწორი ხაზი, მართკუთხედი, ოვალი, მრავალკუთხედი, მომრგვალებული მრავალკუთხედი, ფუნჯი, სამ-ლეილი, ფერის ჩამსხმელი და სხვ. – არის გათვალისწინებული. მაგრამ ზოგჯერ მაინც ხდება აუცილებელი გამოსახულების ცალკეულ პიქსელზე მუშაობა. მაგალითად, წარმოვიდგინოთ, რომ შექმნილი გამოსახულების გარკვეულ ნაწილში რაღაც სიმბოლოს (ვთქვათ, (Y)-ს) მოთავსება გვჭირდება.

სამანსოვრო

პიქსელი
რასტრი
რასტრული გრაფიკა
ფორმატი BMP

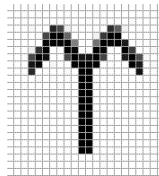
თუ ამ ამოცანის შესრულებას იმ ინსტრუმენტების დახმარებით შევეცდებით, რომლებიც Paint-ში აქამდე ვმუშაობდით, ეს არც ისე იოლი აღმოჩნდება. იმ მეთოდით კი, რომელსაც ახლა შეისწავლით, ამის გაკეთება ძალიან ადვილად შეიძლება.

გამოსახულების ცალკეული პიქსელის რედაქტირების (მისი ფერის შეცვლის) მიზნით ინსტრუმენტ  „ფანქრით“ (Pencil) სარგებლობენ. ამ დროს მოსახერხებელია რეჟიმ „ბადის“ გამოყენება. ამ რეჟიმზე პროგრამა Paint-ის მენიუს ბრძანება View⇒Gridlines-ის (პროგრამა PaintNET-ში – View⇒Pixel Grid -ის) დახმარებით გადადიან.

სამუშაო

2

1. გახსენით გრაფიკული რედაქტორი PaintNET;
2. ინსტრუმენტ  -ის (Magnifier – მასშტაბი) გამოყენებით რამდენჯერმე გაადიდეთ სამუშაო ველი;
3. მენიუში აირჩიეთ ბრძანება View⇒Pixel Grid. სამუშაო ველზე გამოჩნდება „ბადე“;
4. აიღეთ ინსტრუმენტი  (Pencil – ფანქარი) და დახატეთ შემდეგი სიმბოლო:



5. ინსტრუმენტ Magnifier-ის დახმარებით დაუბრუნეთ სამუშაო ველს საწყის ზომები.

ოპერაციულ სისტემა Windows-ში რასტრული გამოსახულებების შენახვის ძირითადი ფორმატია *Windows Bitmap*. ამ ფორმატის ფაილების გაფართოებაა **BMP**. რასტრული გამოსახულებებისათვის ასევე გამოიყენება (განსაკუთრებით, ინტერნეტში) ფორმატები GIF და JPEG. ბოლო ხანებში ფართო გამოყენება პოვა PNG ფორმატმაც. დასახელებულ პროგრამებში ფაილის ზომების შესამჩნევად შემცირების შემთხვევაში გამოსახულების ხარისხი, პრაქტიკულად, არ უარესდება. მაღალხარისხოვანი რასტრული გამოსახულების შესანარჩუნებლად ფორმატ TIFF-ს იყენებენ.

რასტრული გრაფიკა საკმაოდ რთული და ზუსტი ნახატების შექმნის საშუალებას იძლევა. ესაა რასტრული გამოსახულებების ძირითადი **უპირატესობა**. თუმცა რასტრულ გრაფიკას გარკვეული **ნაკლიც** აქვს. ეს ნაკლი მაშინ ვლინდება, როდესაც ნახატს განსაკუთრებით მაღალი სიზუსტე და მკაფიო საზღვრები მოეთხოვება. ჯერ ერთი, მაღალი

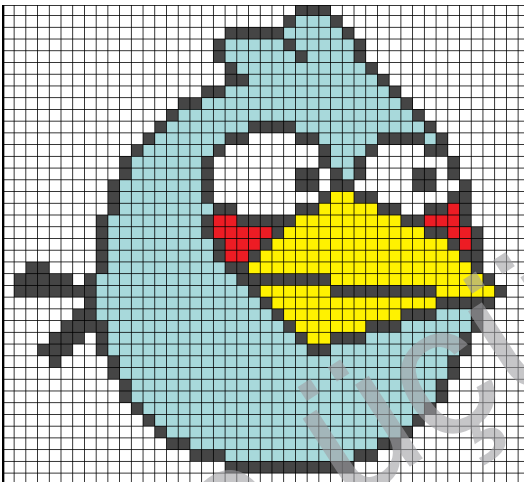
გარჩევადობის რასტრულ გამოსახულებებს ძალიან დიდი მოცულობა აქვს; მეორეც – სირთულეები თავს იჩენს რასტრული გამოსახულებების გადიდების ან შემცირების დროს. მასშტაბის გადიდებისას რასტრის ელემენტები მოჩანს, რის შედეგადაც გამოსახულება მახინჯდება.

შევისწავლოთ დამოუკიდებლად

გრაფიკულ რედაქტორ Paint-ში გახსენით BMP-ს ფორმატის ნებისმიერი ფერადი ნახატი. შეინახეთ ეს ნახატი JPEG ფორმატში. დახურეთ პროგრამა Paint და ფურცელზე აღნიშნეთ თითოეული ფაილის პარამეტრები (დისკზე დაკავებული მოცულობა). შემდეგ ხელახლა გახსენით ეს ფაილები და მათი შეცვლის (მაგალითად, რომელიმე ნაწილის ნაშლის) მერე ახალი სახელით შეინახეთ. მიაქციეთ ყურადღება, როგორ შეიცვლება ამ ფორმატთაგან თითოეულის მოცულობა. გამოარკვიეთ, რატომ დარჩა იგივე ფაილის მოცულობა BMP ფორმატში, ხოლო JPEG-ში – შეიცვალა.

შეამოწმა შენი ცოდნა ?

1. რა არის რასტრი?
2. რასტრული გამოსახულების მისაღებ რომელ მონაცემებს იცნობთ?
3. რა უპირატესობა და ნაკლი აქვს რასტრულ გრაფიკას?
4. დახატეთ მოცემული ნახატი პროგრამა PaintNET-ში.





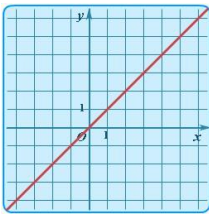
- ფუნქციების წარმოდგენის რომელ ფორმებს იცნობთ მათემატიკაში?

საქმიანობა

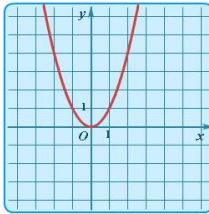
1

განსაზღვრეთ წარმოდგენილი გრაფიკებისა და ფორმულების ურთიერთშესაბამისობა.

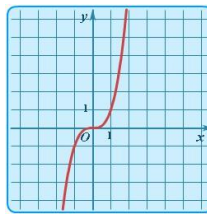
$$y = \sin x$$



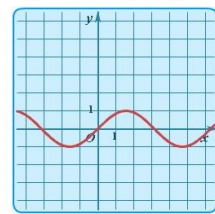
$$y = x^2$$



$$y = kx$$



$$y = x^3$$



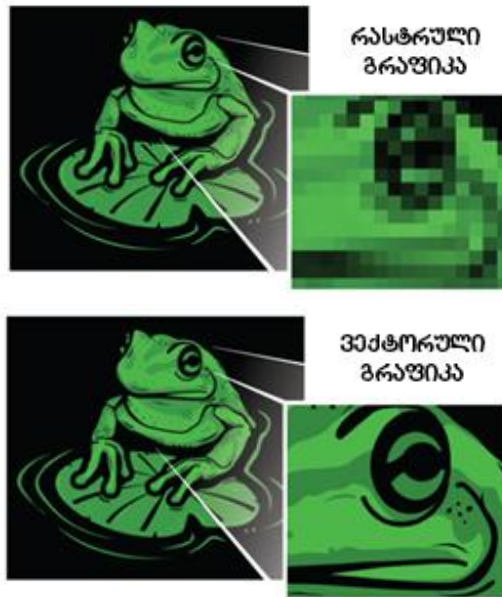
ვიმსჯელოთ:

- რომელი უფრო მეტ ადგილს იკავებს კომპიუტერის მეხსიერებაში: ფორმულა თუ მისი შესაბამისი გრაფიკი?

კომპიუტერში გრაფიკული ინფორმაციის წარმოსადგენად ორი მიდგომით სარგებლობენ. პირველი მათგანის – **რასტრული გრაფიკის** – თავისებურებებს, მის უპირატესობებსა და ნაკლს, წინა გაკვეთილზე გაეცანით. რასტრული გრაფიკის ნაკლთან დაკავშირებული პრობლემების წარმოშობისას მეორე მიდგომას – **ვექტორულ გრაფიკას** მიმართავენ. ვექტორული გრაფიკის უპირატესობები ისეთ ნახატებზე, ნახაზებსა და სქემებზე მუშაობისას ვლინდება, რომლებსაც მკაფიო და გამოკვეთილი კონტურების შენარჩუნება ესაჭიროება.

ვექტორულ გრაფიკაში ობიექტი ცალკეული წერტილების (პიქსელების) კი არა, ხაზების ნაკრების მეშვეობით იქმნება. ხაზის ფორმის არჩევა მათემატიკური კანონზომიერებას ემორჩილება. სხვაგვარად რომ ვთქვათ, ეკრანზე ან ამონაბეჭდში ხაზის ადგილის, სიგრძისა და მიმართულების განსასაზღვრავად მათემატიკური ფორმულებით სარგებლობენ. მაგალითად, როგორც იცით, სწორი ხაზის დასახაზავად ორი პარამეტრია საკმარისი. სწორი ხაზის განტოლება $y = kx + b$ ფორმულით გამოისახება. თუ k და b პარამეტრები ცნობილია, კოორდინატთა სიბრტყეზე სწორი ხაზის დახაზვა შეიძლება, ანუ ამ დროს არ არის საჭირო სწორის ხაზის თითოეული წერტილის კოორდინატებისა და ფერის მითითება. შედეგად, გრაფიკული გამოსახულების ზომა მცირდება. ამასთან, გამოსახულების გადიდებით ან შემცირებით მისი ხარისხი არ უარესდება.

რასტრული და ვექტორული გამოსახულებების შედარება



თუმცა ვექტორულ გამოსახულებაში ადვილი არ არის ფერთა ერთი გამიდან მეორეზე თანდათანობითი გადასვლა, სასურველი კონტურებისა და საზღვრების გავლება. ეს თვისებები ფოტოსურათსა და ნახატს ახასიათებს. ამიტომ ვექტორულ გრაფიკას, როგორც წესი, სქემებისა და ნახაზების შესაქმნელად იყენებენ.

ვექტორულ გამოსახულებებზე სამუშაოდ ბევრი პროგრამაა შემუშავებული. მათ შორის ყველაზე მეტად გავრცელებლია **CorelDRAW, Adobe Illustrator, Micromedia Freehand**. ინტერნეტში ანიმაციების შესაქმნელად გათვალისწინებული პროგრამა **Micromedia Flash** ასევე ვექტორული გრაფიკის პრინციპებს ეფუძნება. ყველა ჩამოთვლილი პროგრამა კომერციული პროდუქტია, მაგრამ თავისუფლად გავრცელებადი (უფასო) **Micromedia**-ც არსებობს. მათ მიეკუთვნება **Inkscape, OpenOffice.org Draw, Skencil**.



Adobe Freehand



Adobe Flash



Inkscape

ყველა ვექტორულ რედაქტორს ხატვისა და რედაქტირებისათვის დაახლოებით ერთნაირი ინსტრუმენტები აქვს. მათ შორის ძალზე მნიშვნელოვანია **მონიშვნის ინსტრუმენტი**. პროგრამა OpenOffice.org Draw სწორი ხაზების, სხვადასხვა სახის მრუდების, მართკუთხედების, წრეწარებისა და სხვა ფიგურების დახატვის საშუალებას იძლევა.

დაიბასოვრა

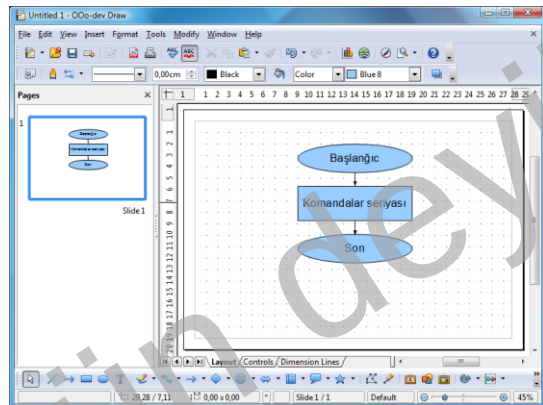
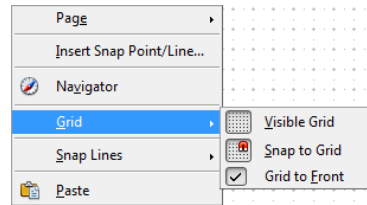
ვექტორული გრაფიკა
ფორმულა
მონიშვნის ინსტრუმენტი
ფორმატი ODF

სამგანზომილებიანი გრაფიკის ინსტრუმენტები პარალელუპიპედის, კონუსის, ცილინდრისა და სხვა გეომეტრიული სხეულების დასახატავადაა გათვალისწინებული. გარდა ამისა, ამ პროგრამის დახმარებით ვებგვერდების მართვის სხვადასხვა ელემენტის (ლილაკების, ტექსტური ველების, რადიოლილაკებისა და ა.შ.) შექმნა შეიძლება. შექმნილ გამოსახულებებს ინახავენ ან თავად პროგრამის ფორმატის – **ODG**-ს სახით, ან სხვა ცნობილ ფორმატებად – **BMP**-ედ, **GIF**-ად, **JPEG**-ედ, **PNG**-ედ, **TIFF**-ად და **WMF**-ად – გარდაქმნის საშუალებით.

საქმიანობა

2

1. ჩართეთ პროგრამა OpenOffice.org Draw. გაიხსნება პროგრამის ძირითადი ფანჯარა;
2. სახატავი ველის ზომებისა და მისი ორიენტაციის განსაზღვრავად გადაადგილეთ მაჩვენებელი სახატავ ველში, შემდეგ კი დააწკაპუნეთ თავის მარჯვენა ლილაკზე. გახსნილ კონტექსტურ მენიუში აირჩიეთ პუნქტი **Page**⇒**PageSetup**. გაიხსნება შესაბამისი დიალოგური ფანჯარა;
3. ფორმატის განყოფილება **Paper**-ში დაადგინეთ ფურცლის ფორმატი, მისი ორიენტაცია (**Landscape**), გვერდის მინდვრები;
4. დასახატავი ფიგურების უფრო ზუსტი გამოსახულების მისაღებად შექმენთ ბადე და გამოაჩინეთ იგი. ამისათვის სახატავ ველში ხელახლა გახსენით მენიუ და პუნქტ **Grid**-ში გაააქტიურეთ ბადის პარამეტრების მაჩვენებლები;
5. დააწკაპუნეთ პროგრამული ფანჯრის ქვედა ნაწილში მდებარე ელიფსის (**Eclipse**) ლილაკზე და დახატეთ შესაბამისი ფიგურა. დააწკაპუნეთ ტექსტის ასაკრეფ ლილაკ **Text**-ზე და ელიფსში ჩანერეთ სიტყვა **დასაწყისი**;
6. იმავე თანამიმდევრობით დახატეთ მართკუთხედისა (**Rectangle**) და კვლავ ელიფსის (**Eclipse**) ფიგურები, ჩანერეთ მათში, შესაბამისად, სიტყვები **ბრძანებათა თანამიმდევრობა** და **დასასრული**;
7. დააწკაპუნეთ ლილაკ **Line Ends with Arrow**-ზე და ისრებით შეაერთეთ დახატული ფიგურები. შედეგად მიიღება ხაზოვანი ალგორითმის ზოგადი ბლოკ-სქემა;
8. ცალკეული ობიექტებისაგან შედგენილი ბლოკ-სქემის გადასაადგილებლად და დასაკოპირებლად აუცილებელია მისი დაჯგუფება. ამისათვის თქვენთვის ცნობილი ერთ-ერთი მეთოდით მონიშნეთ ყველა ობიექტი. შემდეგ თავის მაჩვენებელი მონიშნულ ობიექტებზე დააყენეთ და მის მარჯვენა ლილაკზე დააწკაპუნეთ. გახსნილ კონტექსტურ მენიუში აირჩიეთ



- ბრძანება Group. ყველა მონიშნული ობიექტი ერთ ჯგუფად გაერთიანდება;
9. შექმნილი ვექტორული გამოსახულების შესანახად აირჩიეთ მენიუს ბრძანება File⇒Save As. გახსნილ შესაბამის დიალოგურ ფანჯარაში მიუთითეთ ფაილის ტიპი *ODF Drawing* და მიაწიქეთ ფაილს სახელი **alqoritm.odg**;
 10. აირჩიეთ მენიუს ბრძანება File⇒Export. გახსნილ შესაბამის დიალოგურ ფანჯარაში მიუთითეთ ფორმატი JPEG. ბლოკ-სქემა რასტრული გამოსახულების სახით შეინახება;
 11. დახურეთ პროგრამა OpenOffice.org Draw.

შევისწავლოთ დამოუკიდებლად

გადადით იმ საქალაქში, რომელშიც ბლოკ „საქმიანობაში“ შექმნილი ფაილები, alqoritm.odg და alqoritm.jpg, გაქვთ შენახული. ჯერ შეადარეთ, რა ადგილი უკავია ამ ფაილებს დისკზე. შემდეგ ისინი რიგრიგობით გახსენით OpenOffice.org Draw-სა და Paint-ში. რამდენჯერმე გაადიდეთ გამოსახულება და შეადარეთ მათი ხარისხი (კონტურების სიზუსტე). ემთხვევა თუ არა ხილული შედეგი იმას, რაც ისწავლეთ?

შეამოწმა შენი ცოდნა ?

1. რა არის ვექტორული გრაფიკა?
2. რისგან იქმნება ვექტორული გამოსახულება?
3. რა უპირატესობა და ნაკლი აქვს ვექტორულ გრაფიკას?
4. დახატეთ ალგორითმული კონსტრუქციების, „სრული განშტოებისა“ და „არასრული განშტოების“, ბლოკ-სქემები.
5. დახატეთ კომპიუტერის მოწყობილობის სტრუქტურული სქემა.

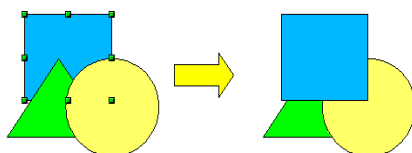
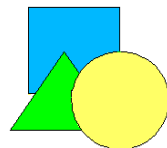


- რა არის გრაფიკული პრიმიტივი?
- რომელი გრაფიკული პრიმიტივებით სარგებლობდით პროგრამა Google SketchUp-ში?

სამიანიშობა

1

1. დახატეთ მოცემული ნახატი გრაფიკულ რედაქტორ Paint-ში;
2. შეეცადეთ, შეუცვალოთ ფიგურებს ურთიერთმდებარეობა;
3. გახსენით ვექტორული რედაქტორი OpenOffice.org Draw;
4. პანელ Drawing-ზე (ხატვა) დაანკაპუნეთ ლილაკ Basic Shapes-ზე (პანელი Drawing, ჩვეულებრივ, პროგრამის მთავარი ფანჯრის ქვედა ნაწილშია მოთავსებული. თუ ის ეკრანზე არ ჩანს, მაშინ საჭიროა მისი გააქტიურება მენიუს ბრძანება View⇒Toolbars-ის დახმარებით);
5. გახსნილ ფანჯარაში ჯერ კვადრატის, მერე სამკუთხედისა და, ბოლოს, წრეხაზის არჩევით მივიღებთ ზემოთ მოცემულ ნახატს;
6. მონიშნეთ ობიექტი „კვადრატი“, შემდეგ კი მენიუ Modify⇒Arrange⇒Bring to Front-ში აირჩიეთ პუნქტი . მონიშნული კვადრატი წინა პლანზე გადმოინაცვლებს.



ვიმსჯელოთ:

- რატომ არის შეუძლებელი რასტრულ რედაქტორში იმ მოქმედების შესრულება, რომელიც ადვილად სრულდება ვექტორულ რედაქტორში?

ვექტორულ რედაქტორში მოქმედების შესრულებას ის თავისებურება ახასიათებს, რომ გეომეტრიული ფიგურები (ობიექტები) დახატვის შემდეგაც დამოუკიდებელი რჩება. სხვაგვარად რომ ვთქვათ, იმ ობიექტებზე, რომლებიც ვექტორულ გამოსახულებას ქმნის, მერეც შეიძლება მოქმედების ჩატარება. რასტრულ გამოსახულებებში ასე არ არის: გეომეტრიული ფიგურები (ობიექტები) „თავისუფალი“ მხოლოდ ხატვის მომენტშია. შემდეგ ისინი ფერადი წერტილების გროვად იქცევა.

ღიზიანება

ფენა
დაჯგუფება
დაცალკევება

ვექტორულ გრაფიკაში დახატული ყოველი გრაფიკული პრიმიტივი ცალკე შრეზე მდებარეობს. ანუ ვექტორული გამოსახულება შედგება შრეებისგან, რომელთა რაოდენობა მისი შემადგენელი ობიექტების რაოდენობის ტოლია. ვექტორულ გამოსახულებაში ობიექტები ვიზუ-

ალურად თითქოს ერთმანეთზეა გადაფარებული, მაგრამ სინამდვილეში ისინი აბსოლუტურად დამოუკიდებელია.

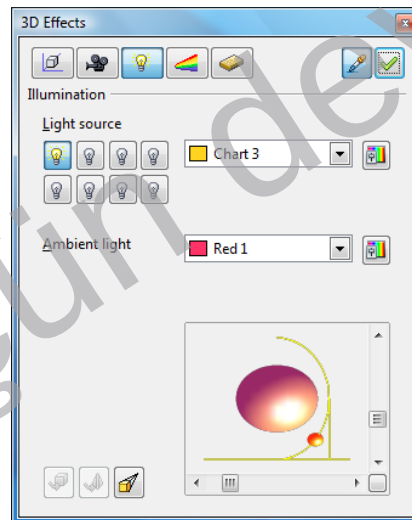
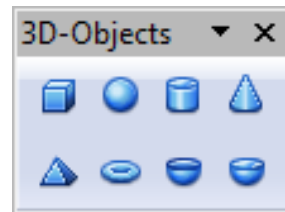
ვექტორულ რედაქტორში დახატული ფიგურების დაჯგუფებით მთელი ობიექტის შექმნა შეიძლება. მოქმედება **დაჯგუფებას** თქვენ წინა გაკვეთილზე გაეცანით. ჯგუფის მიმართ განხორციელებული მოქმედება მის შემადგენელ ყველა ობიექტს ეხება. მაგრამ ზოგჯერ აუცილებელი ხდება გარკვეული ცვლილების შეტანა ჯგუფში შემავალ ერთ რომელიმე ობიექტში. ამისათვის დაჯგუფების საპირისპირო ბრძანება **განცალკევებას** (Ungroup) მიმართავენ.

თქვენ უკვე იმუშავეთ მარტივი სამგანზომილებიანი ფიგურების შექმნაზე პროგრამა Google SketchUp-ში. პროგრამა OpenOffice.org Draw-საც გააჩნია სამგანზომილებიან გრაფიკაზე მუშაობის ფართო შესაძლებლობები.

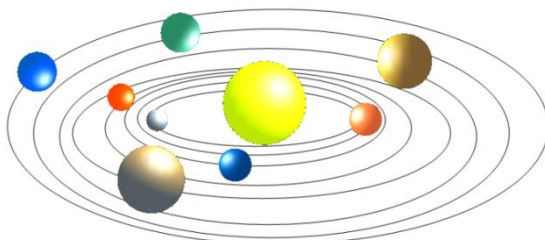
საქმიანობა

2

1. გახსენით პროგრამა OpenOffice.org Draw;
2. აირჩიეთ მენიუს ბრძანება File⇒New⇒Drawing.
გაიხსნება სუფთა სახატავი ველი;
3. გაააქტიურეთ ინსტრუმენტების პანელი 3D Objects. ამისათვის აირჩიეთ მენიუს ბრძანება View⇒Toolbars⇒3D-Objects;
4. თანამიმდევრობით დახატეთ კუბი (Cube), სფერო (Sphere), ცილინდრი (Cylinder), კონუსი (Cone), პირამიდა (Pyramid), რგოლი (Torus);
5. სახატავ ველში მონიშნეთ სფერო და დაანკაპუნეთ მასზე თავის მარჯვენა ლილაკით. გახსნილ კონტექსტურ მენიუმში აირჩიეთ პუნქტი 3D Effects.
გაიხსნება შესაბამისი დიალოგური ფანჯარა;
6. დაანკაპუნეთ ფანჯრის ზედა ნაწილში მდებარე ლილაკ Illumination-ზე (განათება). აირჩიეთ პარამეტრები Light source (სინათლის წყარო) და Ambient light (გარს შემოედენილი სინათლე). თავის ან გვერდის მოძრაობის პანელების დახმარებით შეცვალეთ სინათლის წყაროს მდებარეობა და დააკვირდით, როგორ იცვლება სამგანზომილებიანი ფიგურის განათება. ობიექტისთვის არჩეული თვისებების მისანიჭებლად დაანკაპუნეთ ლილაკ  Assign-ზე (მინიჭება);



7. რედაქტორის ამ შესაძლებლობების გამოყენებით, შექმენით მზის სისტემის სამგანზომილებიანი მოდელი;



8. დაარქვით ფაილს შესაბამისი სახელი და შეინახეთ იგი თქვენს საქალაქო ფაილში.

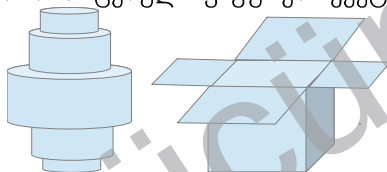
შევისწავლოთ დამოუკიდებლად

1. გარკვეით პროგრამა OpenOffice.org Draw-ის პანელ Drawing-ის ლილაკების დანიშნულება.
2. გამოიყენეთ მენიუ Arrange-ის სხვა პუნქტები და დააკვირდით, როგორ შეიცვლება ამ დროს მონიშნული ობიექტის მდებარეობა სხვა ობიექტების მიმართ.

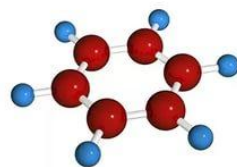


შეამოწმა შენი ცოდნა

1. რასტრული და გრაფიკული გამოსახულებების განმასხვავებელ რომელ თავისებურებებს გაეცანით?
2. რა უპირატესობა აქვს გრაფიკული რედაქტორის შრეებზე მუშაობას?
3. რატომ არ არის გრაფიკულ რედაქტორ Paint-ში დაჯგუფების ბრძანება?
4. შექმენით ქვემოთ მოცემული ფიგურები ვექტორულ რედაქტორში.



5. დახატეთ ვექტორულ რედაქტორში ქიმიური ნივთიერების მოლეკულის მოდელი. შეეცადეთ, გამოიყენოთ ამ ნვთიერების სახელწოდება.



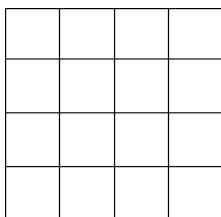


- ინფორმაციის რა სახეებს იცნობთ?
- როგორ ხდება ტექსტური ინფორმაციის კოდირება კომპიუტერში?

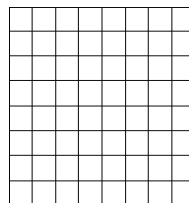
საქმიანობა

1. უჯრებიან ფურცელზე დახაზეთ მართკუთხედი ჰორიზონტალური და ვერტიკალური უჯრების შესაბამისი რაოდენობით;

0110
1001
1001
0110



00111100
01111110
11100111
11000011
11000011
11100111
01111110
00111100



2. მოძებნეთ უჯრები, რომლებიც შეესაბამება „1“-ს და გააფერადეთ ისინი;
3. შეადარეთ მიღებული გამოსახულებები ერთმანეთს.

ვიმსჯელოთ:

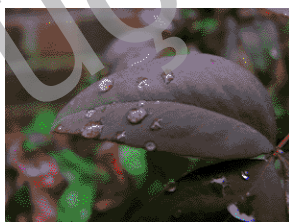
- რომელი ცხრილის გამოსახულებაა უფრო მკაფიო? რა უნდა გავაკეთოთ, რომ მეორე გამოსახულება უფრო ხარისხიანი გამოვიდეს?
- შეგვიძლია, თუ არა, ეს ნულებიანი და ერთიანებიანი ცხრილები მიღებული გამოსახულებების კოდებად ჩავთვალოთ?

ნარმოდგენის ფორმისგან (რასტრულია თუ ვექტორული) დამოუკიდებლად, გრაფიკული ინფორმაცია კომპიუტერის ეკრანზე რასტრის ფორმატით კოდირდება. ამასთან, ყოველი გამოსახულება ფერადი ნერტილების – პიქსელებისგან ფორმირებული მართკუთხა ცხრილისგან შედგება. თითოეული ნერტილის ფერი და სიკაშკაშე რიცხვით გამოისახება, რაც გრაფიკული ინფორმაციის ნარმოსადგენად ორობითი კოდის გამოყენების საშუალებას იძლევა.

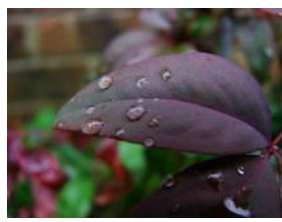
ბიტების რაოდენობას, რომელიც გამოსახულების ერთი პიქსელის ფერის კოდირებისათვის გამოიყენება, **ფერის სიღრმეს** უწოდებენ. როგორც თქვენთვის ცნობილია, 1 ბიტით 2 ფერის კოდირება (2^1) შეიძლება, 2 ბიტით – 4 ფერის (2^2), 4 ბიტით – 16 ფერის (2^4), 8 ბიტით – 256 ფერის (2^8). ციფრულ ვიდეოში ფერის სიღრმისთვის 15 ბიტია საკმარისი. იმისათვის, რომ ფოტოსურათზე ფერი ბუნებრივი იყოს, სულ ცოტა, 24 ბიტია აუცილებელი.



1 ბიტი (2 ფერი)



4 ბიტი (16 ფერი)



24 ბიტი (16777216 ფერი)

გრაფიკული გამოსახულების მოცულობის დასადგენად საჭიროა, ვიცოდეთ პიქსელების რაოდენობა სიგანესა და სიმაღლეზე (შესაბამისად, a და b), ასევე ფერის სიღრმე (n). მაშინ გრაფიკული გამოსახულების მოცულობა (V) შემდეგი ფორმულით გამოითვლება:

$$V = a \cdot b \cdot n$$

მაგალითი

რასტრულ გამოსახულებაში, რომლის ზომა 256×256 პიქსელია, 512 ფერია გამოყენებული. რა ადგილს დაიკავებს ეს გამოსახულება კომპიუტერის მეხსიერებაში?

ამოხსნა: ჯერ ვიპოვოთ გამოსახულების შემადგენელი პიქსელების საერთო რაოდენობა. რიცხვ 2-ის ხარისხების გამოყენებით, შეგვიძლია, გავამარტივოთ გამოთვლები.

$$a \cdot b = 256 \cdot 256 = 2^8 \cdot 2^8 = 2^{16}$$

ვინაიდან პალიტრა 512 ფერს შეიცავს, წარმოვადგინოთ ეს რიცხვი 2-ის ხარისხების სახით და ვიპოვოთ ფერის სიღრმეს:

$$512 = 2^n \Rightarrow 2^9 = 2^n \Rightarrow n = 9$$

მაშასადამე, ფერის კოდირებისათვის გამოყენებულია 9 ბიტი. ამგვარად, რასტრული გამოსახულების მიერ დაკავებული მეხსიერების მოცულობა იქნება

$$V = 2^{16} \cdot 9 \text{ ბიტი} = 2^{13} \cdot 9 \text{ ბაიტი} = 2^3 \cdot 9 \text{ კბაიტი} = 72 \text{ კბაიტი}$$

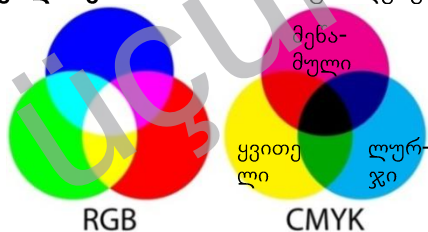
ღამიანსუვრა

ფერის სიღრმე
ფერის შემავსებელი
კოდირება RGB
კოდირება CMYK

მონიტორის ეკრანზე ნებისმიერი ფერი სამი ძირითადი ფერის – წითლის (Red), მწვანის (Green) და ლურჯის (Blue) – შერევით მიიღება. კოდირების ასეთ სისტემას **RGB** (იკითხება „ერ-ჯი-ბი“) ეწოდება. ამ ფერების

გარკვეული თანაფარდობით შერევით ნებისმიერი სხვა ფერის მიღება შეიძლება, რომლის აღქმაც კი შეუძლია ადამიანის თვალს. სამივე ფერის არარსებობა იძლევა შავ ფერს, ხოლო თეთრი სამივე ფერს ას-ასი პროცენტით შეიცავს.

თუ თითოეული ძირითადი ფერის სიკაშკაშის კოდირებისთვის 8 ბიტი (256 -დან) გამოიყენება, მაშინ გამოსახულების ყოველი წერტილის კოდირებას 24 ბიტი დასჭირდება. კოდირების ასეთი სისტემა $16,8$ მილიონი ფერის ცალსახად განსაზღვრის საშუალებას იძლევა. ფერადი გამოსახულებების 24-ბიტისანი ორობითი რიცხვების დახმარებით კოდირებას **სრულ ფერად კოდირებას (True Color)** უწოდებენ.



ბეჭდვისას მიზანშეწონილია არა ამ სამი ძირითადი ფერის, არამედ **შემავსებელი ფერების** გამოყენება. შემავსებელი ფერებია: ციანური (Cyan), მენამული (Magenta) და ყვითელი (Yellow). ვინაიდან საღებავი ნივთი-

ერებები არასრულყოფილია, ამ ფერების შერევით იდეალური შავი ფერი არ მიიღება. ამიტომ ბექდვისას შავი ფერი (Black) ცალკე გამოიყენება. კოდირების ასეთ სისტემას **CMYK** (იკითხება „ცმიკი“) ეწოდება. შავი ფერი, ლურჯისგან განსასხვავებლად, K ასოთი აღინიშნება.

თუ ფერების პალიტრის გადმოცემისას დიდი სიზუსტე არ მოითხოვება, ნაკლები რაოდენობის ბიტებს იყენებენ. მაგალითად, გამოსახულების კოდირებას 16-ბიტისანი ორობითი რიცხვების დახმარებით **High Color**-ს უწოდებენ. ასეთი გამოსახულებების მოცულობა (დისკზე დაკავებული ადგილი) სრულფერებიან (True Color) გამოსახულებასთან შედარებით ცოტათი ნაკლებია (ცხადია, ხარისხიც ნაკლებია).

შავისნაწლათ დამოუკიდებლად

გრაფიკულ რედაქტორ Paint-ში გახსენით BMP ფორმატის ნებისმიერი ფერადი გამოსახულება. შეინახეთ ეს ნახატი კოდირების სხვადასხვა სქემის (Monochrome Bitmap, 16 Color Bitmap, 256 Bitmap, 24-bit Bitmap) გამოყენებით. დახურეთ პროგრამა და გადადით იმ საქალაქში, რომელშიც ამ ფაილებს ინახავთ. შეადარეთ შენახული ფაილების ზომები და შეეცადეთ, ახსნათ მათ შორის არსებული განსხვავების მიზეზი.

24-bit Bitmap (*.bmp;*.dib)
Monochrome Bitmap (*.bmp;*.dib)
16 Color Bitmap (*.bmp;*.dib)
256 Color Bitmap (*.bmp;*.dib)
24-bit Bitmap (*.bmp;*.dib)
JPEG (*.jpg;*.jpeg;*.jpe;*.jfif)
GIF (*.gif)
TIFF (*.tif;*.tiff)
PNG (*.png)

შეამოწმა შენი ცოდნა

- რომელ ფერებს უწოდებენ ძირითადს?
- განსაზღვრეთ, რა გამოსახულება მიიღება, თუ ციფრ 1-ის შესაბამის უჯრებს გავაფერადებთ?
- რასტრული გამოსახულების, რომლის ზომა 32X32 პიქსელია, შესანახავად გამოყოფილია 512 ბიტი ადგილი. ფერების რა მაქსიმალურ რაოდენობას შეიძლება შეიცავდეს გამოსახულების პალიტრა?
- 256-ფერიანი გამოსახულების ზომა 1034X640 პიქსელია. განსაზღვრეთ გრაფიკული ფაილის მოცულობა.
- რა მოცულობას დაიკავებს გამოსახულება, რომლის ზომაა 640X480 პიქსელი, 16-ბიტისანი კოდირების შემთხვევაში?

0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	1	0	0	0	0	1	1	0	0	0	0	0	0	1	1
1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1
1	1	0	1	1	0	1	1	0	0	0	0	0	0	1	1
1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0
0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0
0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0



- რას წარმოადგენს ბგერა ფიზიკის თვალსაზრისით?
- კომპიუტერის რომელი მოწყობილობაა განკუთვნილი ხმოვან ინფორმაციაზე სამუშაოდ?

სამუშაო

1

ტექსტურ რედაქტორში შექმენით შემდეგი ცხრილი. ჩანერეთ შესაბამის ადგილებზე ინფორმაცია მოცემული მოწყობილობების შესახებ.

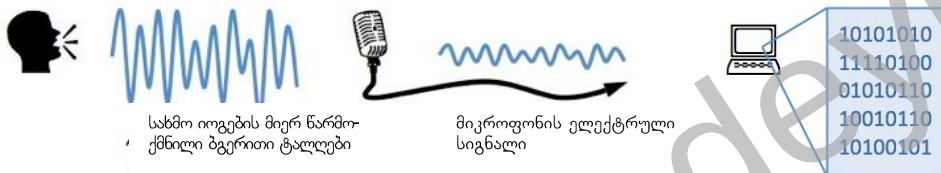
მოწყობილობა	დანიშნულება, მუშაობის პრინციპი
მიკროფონი	
დინამიკები	
ხმის ბარათი	

ვიმსჯელოთ:

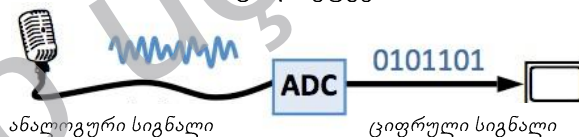
– როგორ უკავშირდება ეს მოწყობილობები ერთმანეთს კომპიუტერში?

კომპიუტერი, ისევე როგორც თანამედროვე ელექტრონულ მოწყობილობათა უმრავლესობა (ფოტო და ვიდეოკამერები, მობილური ტელეფონები და სხვ.), **ციფრული მოწყობილობაა**, რადგანაც ინფორმაციის შენახვისა და დამუშავებისას ის რიცხვებზე (ციფრებზე) მუშაობს.

მუდმივად ცვლადი ფიზიკური სიდიდეების აღსანიშნავად ხშირად იყენებენ ტერმინ „ანალოგურს“. მაგალითად, ადამიანის სახმო იოგების მიერ ლაპარაკისას წარმოქმნილ ბგერით ტალღებს ანალოგური ბუნება აქვს. მიკროფონს შეუძლია ამ ტალღების ცვლად ელექტრულ სიგნალებად გარდაქმნა, რომლებიც ასევე ანალოგური ბუნებისაა.

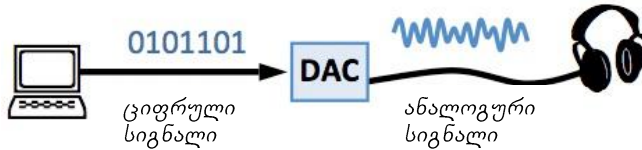


ვინაიდან კომპიუტერი ციფრული მოწყობილობაა, ანალოგურ მოწყობილობებთან სამუშაოდ მას გარდამქმნელი სჭირდება. ასეთი გარდამქმნელი კომპიუტერის ხმის ბარათშია ჩადგმული. მიკროფონიდან მიღებულ ელექტრონულ სიგნალებს (ანალოგურ სიგნალებს) ციფრულ სიგნალებად ხმის ბარათზე განთავსებული **ანალოგურ-ციფრული გარდამქმნელი (analog-to-digital converter, ADC)** გადააქცევს.

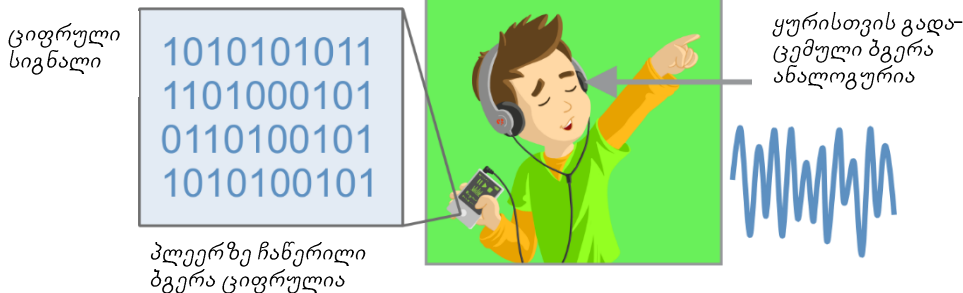


გარდა ამისა, კომპიუტერის ხმის ბარათი ისეთი მოწყობილობითაც არის აღჭურვილი, რომელიც ციფრულ მონაცემებს ანალოგურ სიგნალ-

ლებად (ბგერებად) გარდაქმნის. **ციფრულ-ანალოგური გარდაქმნელი (digital-to-analog converter, DAC)** მიღებულ ციფრულ სიგნალებს კვლავ ანალოგურ სიგნალებად გადააქცევს და მათ ყურსაცემებს ან დინამიკებს გადასცემს. ამ მოწყობილობებს, თავის მხრივ, ანალოგური სიგნალები ხმის სახით გამოაქვს „გარეთ“.



ციფრულ-ანალოგური და ანალოგურ-ციფრული გარდაქმნელები სხვა მოწყობილობებსაც აქვს, მაგალითად, კომპაქტ-დისკების პლეერს (MP3 - პლეერს). მისი ციფრულ-ანალოგური გარდაქმნელი დისკიდან ამოკითხულ ციფრულ მონაცემებს ბგერის სახის ანალოგურ სიგნალებად გადააქცევს.



კოდირების ასეთი სისტემა ბგერას გარკვეულად ამახინჯებს, ვინაიდან ციფრულ-ანალოგური გარდაქმნელიდან მიღებული სიგნალი ორიგინალისაგან ცოტათი განსხვავდება. ეს განსხვავება განსაკუთრებით შესამჩნევია მაღალხარისხიანი საორკესტრო მუსიკის ჩაწერისა და მოსმენის დროს.

სხვა მონაცემების მსგავსად, ხმაც (ბგერა) კომპიუტერში ფაილებში ინახება. ძირითადად სამი სახის ხმოვანი ფაილებს (აუდიოფაილებს) იყენებენ: **wav**-ს, **mp3**-ს და **midi**-ს.

Wav. ამ ფორმატის ფაილი ხმოვანი ფაილია, რომელიც შეიცავს მუსიკის ფრაგმენტებს ციფრული ფორმით. **Wav** (ინგლ. wave-დან – ტალღა) ფაილებში ინახება საკუთრივ Windows-ისა და თქვენ მიერ შექმნილი აუდიოფაილები.

MP3. ეს არის ფორმატი, რომელიც გულისხმობს ხმოვანი ფაილების შეკუმშვას მათ მიერ დისკზე დაკავებული მოცულობის შემცირების მიზნით. MP3-ფაილის ყოველი წუთი მეცხიერების დაახლოებით 1 მბაიტს იკავებს. შესაძლებელია wav ფორმატის ფაილების mp3

დამატებითი

ანალოგური სიგნალი
ციფრული სიგნალი
ანალოგურ-ციფრული გარდაქმნელი
ციფრულ-ანალოგური გარდაქმნელი
ხმის ბარათი
ფონოგრამი

ფორმატის ფაილებად გარდაქმნა.

MIDI. ამ ფორმატის ფაილები შეიცავს ინფორმაციას ნოტებისა და მათი წარმომქმნელი მუსიკალური ინსტრუმენტების შესახებ. სინთეზატორი ისე უკრავს მელოდიას, როგორც ამას მუსიკოსი ნოტების დახმარებით აკეთებს.

ესაინტა-რესო...

თქვენ უკვე გაიგეთ, როგორ ხდება კომპიუტერში ხმოვანი ინფორმაციის კოდირება. ადამიანები დიდხანს ფიქრობდნენ იმაზე, თუ როგორ ჩაენერათ ხმა და გადაეცათ იგი მომავალი თაობებისთვის.

ცნობილია, რომ მერხევი სხეული ხმას გამოსცემს, მაგალითად, დოლი დარტყმისას რხევას (ვიბრაციას) ინყებს და ამ რხევებს გარემომცველ ჰაერს გადასცემს. ამ დროს წარმოქმნილი ბგერითი ტალღები ჰაერში ვრცელდება და ჩვენს ყურამდე აღწევს.

1876 წელს შეიქმნა **ტელეფონი**. ბგერითი ვიბრაციის აღსაქმელად მასში მემბრანა – თხელი ელასტიკური ფირფიტა – არის მოთავსებული. დიდ გამომგონებელ ტომას ედისონს აინტერესებდა, შესაძლებელი იყო თუ არა მემბრანის ვიბრაციის ჩანერგა. ედისონმა აიღო სახმო მილი და მემბრანას ნემსი შეუერთა. როდესაც რუპორს შიგნიდან რაღაც ეხებოდა, მემბრანასთან ერთად ირხეოდა ნემსიც, რომელიც მის ქვეშ მოთავსებულ ქაღალდზე ნაკვალევს ტოვებდა.

პირველი ფონოგრაფი



ამგვარად, ხმის ჩანერგა მოხერხდა, მაგრამ როგორ უნდა გახმოვანებულიყო ეს ჩანაწერი? ედისონს კიდევ ერთი შესანიშნავი იდეა მოუვიდა: ჩვეულებრივი ქაღალდის ფურცლის მაგივრად მან კალის ფოლგა გამოიყენა. როდესაც ნემსით ფოლგაშემოხვეულ ცილინდრს ეხებოდნენ, ნემსი მასზე ღარებს აჩენდა. ამ ღარების სიღრმე ნემსის ვიბრაციის სიხშირის შესაბამისად იცვლებოდა.

ედისონს კვლავ ახალი იდეა ენვია: ხომ შეიძლებოდა ამ პროცესის უკუღმა შებრუნება!

თუ ნემსის წვერი ხახუნისას ღარის ფსკერს შეეხება, ის, ისევე როგორც ხმის ჩანერგისას, მემბრანის რხევას გამოიწვევს! ედისონმა აპარატის ესკიზი დახატა და ლაბორატორიის თანამშრომლებს გადასცა. როგორც კი აპარატი დამზადდა, ედისონმა ის გამოსცადა. მან ცილინდრი დაატრიალა და სახმო მილში (რუპორში) ცნობილი საბავშვო ლექსი „მერს ჰყავდა ბეკეკა“ წაიკითხა. ნემსმა ცილინდრზე ღარები გაკანრა. შემდეგ ედისონმა ნემსის წვერი ღარის დასაწყისში მოათავსა, ცილინდრი დაატრიალა და გახარებულმა სახმო მილში საკუთარი ხმა გაიგონა.

ასე გაჩნდა ქვეყნად პირველი ხმის ჩანაწერი მონყობილობა – **ფონოგრაფი**. მისმა შემდგომმა განვითარებამ **გრამოფონისა** და **პატეფონის** შექმნა განაპირობა.

ხმის ჩანერგა და ფაილების ერთი ფორმატიდან მეორედ გარდაქმნა სპეციალური პროგრამული უზრუნველყოფის დახმარებით ხორციელდება.

ინტერნეტში ამ პროგრამების საძიებლად მიმართეთ საიტ www.google.com-ს სიტყვებით: „**sound conversion software**“.

ვიდეოინფორმაციის კოდირებისას საჭირო ხდება როგორც კადრების (გამოსახულების) თანამიმდევრობის, ისე ხმის კოდირება. ამ დროს ფაილის საერთო მოცულობის შესამცირებლად სხვადასხვა მეთოდით სარგებლობენ, მაგალითად, ორ მეზობელ კადრს შორის მხოლოდ განსხვავებას ინერენ.

შავისწავლთ დამოუკიდებლად

ინტერნეტში მოძებნეთ რაიმე ინფორმაცია, მაგალითად, მუჰამად ჰუსაინ შაჰრიარის პოემა "Heydərbabaya salam"-ის ფაილი და საკუთარ კომპიუტერში სხვადასხვა ფორმატით (ტექსტი, ხმა, ვიდეო) გადმოქაჩეთ. შეადარეთ ერთმანეთს ამ ფაილების ზომები. რომელი ფორმატია უდიდესი და რომელი – უმცირესი მოცულობის? შეეცადეთ, ახსნათ ამის მიზეზი.

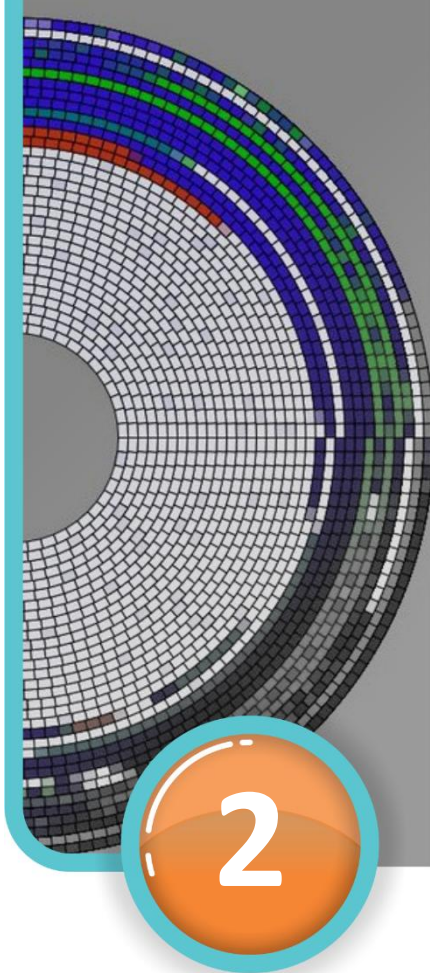
შეამოწმე შენი ცოდნა ?

1. რა ფუნქციას ასრულებს მონოკილობა ADC? DAC?
2. კიდეც რომელი თქვენთვის ცნობილი მონოკილობის სახელწოდება შეიცავს სიტყვა „ფონს“?
3. როგორ კოდირდება ხმოვანი ინფორმაცია კომპიუტერში?
4. რა ტიპის ხმოვანი ფაილები არსებობს?
5. რა არის ფონოგრაფი და როგორ მუშაობს იგი??

შემაჯავებელი კითხვები და დავალებები

1. რა განსხვავებაა რასტრული და ვექტორული გამოსახულებების გადმოცემაში?
2. დაასახელეთ ვექტორული გრაფიკის უპირატესობები და ნაკლოვანებები.
3. რომელი ფორმულით გამოითვლება გრაფიკული გამოსახულების მოცულობა?
4. რას გულისხმობს ცნება „ფერის სიღრმე“ გრაფიკულ გამოსახულებაში?
5. დაადგინეთ შესაბამისობა:

ა) რასტრული გრაფიკული რედაქტორი;	1) .bmp;
ბ) ვექტორული გრაფიკული რედაქტორი;	2) .odg;
გ) რასტრული ფორმატი;	3) PaintNET;
დ) ვექტორული ფორმატი.	4) OpenOffice Draw.
6. ბიტების რა მინიმალური რაოდენობა დაგვჭირდება რასტრული გამოსახულების კოდირებისათვის, თუ მისი შექმნისას 1025 სხვადასხვა ფერია გამოყენებული?
7. შავ-თეთრი გამოსახულების ზომა 640X320 პიქსელს შეადგენს. რას უდრის ამ გამოსახულების საინფორმაციო მოცულობა?
8. რომელი ფაილი დაიკავებს კომპიუტერის მესხიერებაში მეტ ადგილს: 2056 სიმბოლოს შემცველი ტექსტი Unicond-ის კოდირებით თუ 10X10 პიქსელის ზომის 4-ფეროვანი რასტრული გამოსახულება?
9. რა დანიშნულება აქვს კომპიუტერის ხმის ბარათს?
10. რა ფორმატის ხმოვანი ფაილები გამოიყენება უფრო ხშირად თანამედროვე მობილურ ტელეფონებში?



კომპიუტერი

6. კომპიუტერის ძირითადი მახასიათებლები
7. მართვის პანელი
8. დამხმარე პროგრამები. დეფრაგმენტაცია
9. დამხმარე პროგრამები. დისკის განმენდა

6

კომპიუტარის ძირითადი მახასიათებლები



- რა სახის პერსონალურ კომპიუტერებს იცნობთ?
- კომპიუტერის რომელ ნაწილს უწოდებენ მის „ტვინს“?

საქმიანობა

1

შეავსეთ ცხრილი იმ კომპიუტერის მონაცემების მიხედვით, რომელზეც მუშაობთ.

პარამეტრი	პარამეტრის მნიშვნელობა
ოპერაციული სისტემა	
პროცესორის სახელწოდება	
პროცესორის სიჩქარე	
ოპერაციული მეხსიერების მოცულობა	
მონიტორის ეკრანის გარჩევადობა	

ვიმსჯელოთ:

- როგორ ფიქრობთ, ამ მახასიათებელთაგან რომელი განსაზღვრავს ერთი კომპიუტერის უპირატესობას მეორის წინაშე?

თქვენთვის, ალბათ, ცნობილია, რომ კომპიუტერების სიმძლავრე და შესაძლებლობები (გამოთვლების სიჩქარე) წლიდან წლამდე განუხრელად იზრდება. ამ ზრდას, პირველ რიგში, ის მოწყობილობები განაპირობებს, რომელთაგანაც კომპიუტერის სისტემა შედგება. რაც უფრო მაღალია ამ მოწყობილობათა შესაძლებლობები, მით უფრო მაღალი იქნება კომპიუტერის საერთო შესაძლებლობები.

კომპიუტერის ძირითადი მოწყობილობაა **ცენტრალური პროცესორი**, ან, უბრალოდ, **პროცესორი**. პროცესორის მუშაობის სისწრაფე არის ის მთავარი ფაქტორი, რომელიც კომპიუტერის სიმძლავრეზე მოქმედებს. პროცესორის სისწრაფედ იმთავითვე ჩაითვალია მის მიერ ერთ წამში შესრულებულ ოპერაციათა რაოდენობა. ეს იმ მოცემულობას ეფუძნება, რომ კომპიუტერი ერთი და იმავე ტიპის ამოცანებს წყვეტდა, სხვადასხვა ოპერაციის შესრულებას კი დაახლოებით ერთსა და იმავე დროს ანდომებდა. თანამედროვე კომპიუტერებში ასე არ არის. პროცესორის ბრძანებათა ნაკრებში შემავალი ბრძანებები, შესასრულებელი მოქმედებების სირთულის თვალსაზრისით, არერთგვაროვანია. ამიტომ პროცესორის მიერ სხვადასხვა ბრძანების შესრულებაზე დახარჯული დროც შესაძლოა, მნიშვნელოვნად განსხვავებული იყოს. ამასთან, პროგრამების მუშაობის სისწრაფე მხოლოდ პროცესორით არ განისაზღვრება. ის კომპიუტერის სისტემის სხვა კომპონენტებზეც (მაგალითად, ვიდეოსისტემის სიმძლავრეზე) არის დამოკიდებული. გარდა ამისა, თანამედროვე

კომპიუტერები რამდენიმე ოპერაციას (მათ შორის, ცალკეულ ბრძანებებსაც) პარალელურად, ანუ ერთდროულად ასრულებს.

ესაინტერესოებს...

კომპიუტერის პროცესორი თანამედროვე სიჩქარით კი არა, ადამიანისთვის ჩვეული რიტმით რომ მუშაობდეს, მაშინ:

- სიგნალებს კლავიატურიდან იგი ათ წელიწადში ერთხელ მიიღებდა;
- თავისი მაჩვენებლის ეკრანის ერთი კუთხიდან მეორეში გადაადგილებას ათასწლეულები დასჭირდებოდა.

პროცესორის მიერ ნებისმიერი ბრძანების შესრულება ელემენტარული ოპერაციების შესრულების თანამიმდევრობად შეგვიძლია, წარმოვადგინოთ. ნებისმიერ ელემენტარულ ოპერაციას, რომელიც ცვლის პროცესორის მდგომარეობას, განსაზღვრული დრო – ერთი ტაქტი სჭირდება. ბრძანებათა უმრავლესობა რამდენიმე ტაქტში სრულდება. პროცესორის ტაქტების სიხშირეს (**ტაქტურ სიხშირეს**) სპეციალური გენერატორი აღძრავს.

თუ ორი პროცესორი ერთმანეთისგან მხოლოდ ტაქტური სიხშირით განსხვავდება, მაშინ მათ მიერ მონაცემთა დამუშავების სიჩქარე ტაქტური სიხშირის პირდაპირპროპორციული იქნება. ამიტომ ტაქტური სიხშირე კომპიუტერის სიმძლავრის ძირითად მაჩვენებლად განიხილება.

სიხშირის გასაზომად მიღებულია ფიზიკური სიდიდე **1 ჰერცი (ჰც)**. ვინაიდან პირველ პერსონალურ კომპიუტერებში ტაქტების რაოდენობა წამში რამდენიმე მილიონს შეადგენდა, ტაქტური სიხშირისთვის დანესდა სტანდარტული საზომი ერთეული – **1 მეგაჰერცი (1 მჰც)**. თანამედროვე პერსონალური კომპიუტერების ტაქტური სიხშირე **გიგაჰერცებით (გჰც)** იზომება.

პროცესორის ტაქტური სიხშირე კომპიუტერული სისტემის სიმძლავრის ერთადერთი მაჩვენებელი როდია. ერთნაირი ტაქტური სიხშირის მქონე კომპიუტერებიც შეიძლება სხვადასხვა სიმძლავრით ხასიათდებოდეს. გარდა ამისა, კომპიუტერის შესაძლებლობები მარტო პროცესორზეც არ არის დამოკიდებული. თუ კომპიუტერის სხვა კომპონენტები პროცესორს მონაცემებს მისი შესაბამისი სიჩქარით არ მიაწვდის, პროცესორი მოცდება, რაც, საბოლოო ჯამში, სისტემის საერთო სიმძლავრეს დააქვეითებს.

ცენტრალური პროცესორის ნაწილს, რომელიც გამოთვლებზეა პასუხისმგებელი, **ბირთვი** ეწოდება. პირველ კომპიუტერებს მხოლოდ ერთი ბირთვის ჰქონდა. თანამედროვე კომპიუტერებში ორი ან მეტი ბირთვია. რამდენიმე ბირთვის არსებობა რამდენიმე ამოცანის (პროგრამის) პარალელურად შესრულების საშუალებას იძლევა, რაც მნიშვნელოვნად ზრდის კომპიუტერის სიჩქარეს. სხვაგვარად რომ ვთქვათ, რაც მეტი ბირთვი აქვს ცენტრალურ პროცესორს, მით მეტია მისი შესაძლებლობები.

კომპიუტერის მუშაობის დროს მის პროცესორსა და ოპერაციულ მეხსიერებას შორის მონაცემთა განუწყვეტელი გაცვლა მიმდინარეობს. ამ გაცვლის სიჩქარე კომპიუტერული სისტემის მწარმოებლურობის ძირითადი მაჩვენებელია. ის, ერთი მხრივ, მეხსიერების მიკროსქემის პარამეტრებზე, ხოლო მეორე მხრივ, კომპიუტერის მთელი სისტემის თავისებურებებზეა დამოკიდებული.

დაინახოსურა

ცენტრალური პროცესორი
ტაქტური სიხშირე
ბირთვი
ოპერაციული მეხსიერება
ქეშ-მეხსიერება
მყარი დისკი
ვიდეოდაპტერი

ჩვეულებრივ, სალტის (მონაცემთა გადაცემისათვის გათვალისწინებული ელექტრული გამტარის, რომელიც პროცესორსა და მეხსიერებას ერთმანეთთან აერთებს) სიხშირე პროცესორის სამუშაო სიხშირეზე რამდენჯერმე ნაკლებია. ეს კი მუშაობაში შეფერხებას იწვევს.



ამ შეფერხების აღმოსაფხვრელად პროცესორი სპეციალური **ქეშ-მეხსიერებითაა** აღჭურვილი. პროცესორის მიკროსქემაზე მოთავსებული მცირე ზომის ქეშ-მეხსიერება მისი სიხშირით მუშაობს. კითხვისა და ჩანერის დროს მონაცემთა ასლი ოპერაციული მეხსიერებიდან ქეშ-მეხსიერებაში ჩაინერება. როდესაც პროცესორი ხელმეორედ უბრუნდება ამ მონაცემებს, მათ უშუალოდ ქეშ-მეხსიერებიდან იღებს, რომელიც მოცდენა-შეფერხებების თავიდან აცილებას უზრუნველყოფს.

კომპიუტერის პროცესორისა და ოპერაციული მეხსიერების სიჩქარის მაჩვენებლები არსებით გავლენას ახდენს გამოთვლების შესრულების სისწრაფეზე. მაგრამ, როგორც იცით, თანამედროვე კომპიუტერები მარტო გამოთვლებისთვის არ გამოიყენება. ამიტომ კომპიუტერის მწარმოებელურობაზე მისი სხვა მახასიათებლებიც მოქმედებს. ეს მახასიათებლებია:

- ოპერატიული მეხსიერების მოცულობა;
- გარე მეხსიერების მოცულობა;
- ვიდეოსისტემის ხარისხი.

კომპიუტერის მუშაობის სიჩქარის განმსაზღვრელი ერთ-ერთი მნიშვნელოვანი პარამეტრია **ოპერატიული მეხსიერების მოცულობა (RAM)**. თანამედროვე კომპიუტერები მრავალამოცანიან რეჟიმში მუშაობს. ოპერაციული სისტემა ყოველი მომუშავე ამოცანისთვის (პროგრამისთვის) ოპერატიულ მეხსიერებაში ცალკე ადგილს გამოყოფს. თუ ყველა ამოცანისთვის საჭირო ოპერატიული მეხსიერების საერთო მოცულობა არსებული ოპერატიული მეხსიერების მოცულობას გადაჭარბებს, ოპერაციული სისტემა მყარ დისკზე დამატებით ადგილს შექმნის. მაგრამ ვინაიდან მყარი დისკისთვის მიმართვის სიჩქარე ბევრად ნაკლებია ოპერატიული მეხსიერებისთვის მიმართვასთან შედარებით, ეს კომპიუტერის მუშაობას ანელებს. კომპიუტერზე ოპერატიული მეხსიერების დამატებითი მოდულების დაყენებით ასეთი მოქმედების აუცილებლობა მცირდება და მთელი სისტემის მუშაობის სიჩქარე იზრდება.

კომპიუტერის ასევე მნიშვნელოვანი მახასიათებელია მყარი დისკის მოცულობა. თუ პროგრამებისა და მონაცემების ხანგრძლივი შენახვისთვის განკუთვნილი დისკის მოცულობა არასაკმარისია, აუცილებელი ხდება დიდი მოცულობის დისკების გამოყენება.

კომპიუტერზე მუშაობისას მომხმარებელს განუწყვეტილად უხდება მონიტორში ყურება, ამიტომ ძალზე მნიშვნელოვანია **მონიტორ-ვიდეო-ადაპტერის** წყვილის სწორად შერჩევა. ამ დროს ძირითადი მაჩვენებელია კადრების რაოდენობა წამში. თუ ეს სიდიდე წამში 70 კადრზე ნაკლებია, ადამიანის თვალი (და მისი ტვინი) ეკრანზე გამოსახულების ციმციმს აღიქვამს, რაც ადვილად დაღლას იწვევს.

კომპიუტერის პარამეტრების მისათითებლად ზოგიერთ არასტანდარტულ შემოკლებასა და აღნიშვნას იყენებენ. კომპიუტერის ყოველ მყიდველს უნახავს ჩანაწერები, რომლებიც ერთმანეთისგან მძიმით ან "/" სიმბოლოთი გამოყოფი-

ლი ლათინური ანბანის ასოებითა და ციფრებითაა შედგენილი. მაგალითად, ნახატზე მითითებულია ნოუტბუქის შემდეგი პარამეტრები: პროცესორის მოდელი (Intel Core i3-3217U), მისი ტაქტური სიხშირე (1,7GHz – 1,7 გჰც). შემდეგ ჩამოთვლილია ოპერატიული მეხსიერების მოცულობა, მისი მოდელი (4GB DDR3L) და მყარი დისკის მოცულობა (500 GB HDD).



HP 250 G3

Intel Core i3-3217U 1,7GHz, 4GB DDR3L, 500GB HDD, DVD+/-RW, Intel HD 4400, 802.11n WLAN, BT, 15.6 HD w/CAM display, Çanta, Dr.Web Antivirus, Mouse, Win 7 Pro

HP 3500 MT PC

Intel Pentium G2030, 500GB HDD 7200 SATA, DVD+/-RW, 2GB PC3-10600+ w/ HP W2072a 20-inch Widescreen LCD Monitor Dr.Web Antivirus, Win 7 Pro

კომპიუტერების აპარატული უზრუნველყოფა სწრაფი ტემპით მიდის წინ, მათი საშუალო მაჩვენებლები ყოველდღიურად იცვლება. 2015 წელს პერსონალური კომპიუტერების ტაქტური სიხშირე 4 გჰც-ს შეადგენდა. ზოგადი დანიშნულების კომპიუტერების ოპერატიული მეხსიერების მოცულობამ უკვე რამდენიმე გიგაბაიტამდე მიაღწია, ხოლო იმ პერსონალურ კომპიუტერებში, რომლებიც უფრო რთული ამოცანების შესასრულებლადაა განკუთვნილი, ეს მაჩვენებელი 16 გიგაბაიტამდე აღწევს. მყარი დისკის მოცულობა 500 გიგაბაიტს და მეტს შეადგენს.

შევისწავლოთ დამოუკიდებლად

1981 წელს გამოშვებული პირველი IBM კომპიუტერ 8088/86-ის პროცესორის ტაქტური სიხშირე 4,77 მჰც იყო. 1993 წელს Pentium-ის პროცესორს ეს მაჩვენებელი 66 მჰც-ის ტოლი ჰქონდა, ხოლო 1997 წლის ბოლო გამოშვების Pentium-ის პროცესორში მან უკვე 233 მჰც-ს მიაღწია.

შეადარეთ თქვენი კომპიუტერის პროცესორის ტაქტური სიხშირე Pentium-ის ყველაზე მძლავრი პროცესორის ანალოგიურ პარამეტრს. მასზე რამდენჯერ უფრო მძლავრია თქვენი კომპიუტერი?

შეამოწმა შენი ცოდნა ?

1. კომპიუტერის რომელი ძირითადი მახასიათებელი განსაზღვრავს მის სიმძლავრეს?
2. რა არის ტაქტური სიხშირე და რა ერთეულებით იზომება იგი?
3. სად არის მოთავსებული ქეშ-მეხსიერება და რა დანიშნულება აქვს მას?
4. რა მოცულობის ოპერატიული მეხსიერება და მყარი დისკი აქვს ქვემოთ მოყვანილი მახასიათებლების მქონე კომპიუტერს?

Intel core i7 2600k 3.4/Asus P4PE-2x/16 GB DDR3/1,5 TB HDD/Video Asus Radeon AX600 Pro 128 MB TV-out/DVD+RW NEC/ATX



- რა შემთხვევაში ჩნდება სამუშაო მაგიდაზე არსებული პიქტოგრამების მონესრიგების აუცილებლობა?
- როგორ დავახარისხოთ პიქტოგრამები სამუშაო მაგიდაზე პროგრამის ტიპის შესაბამისად?

როგორც იცით, კომპიუტერის ოპერაციული სისტემა მართავს მის აპარატულ და პროგრამულ უზრუნველყოფას და მოხმარებელსა და კომპიუტერს შორის დამაკავშირებლის როლს თამაშობს. მომხმარებელთა მოთხოვნების მრავალფეროვნების გამო ხშირად აუცილებელი ხდება სისტემის მართვის პარამეტრების შეცვლა.

გასულ წელს თქვენ ისწავლეთ, როგორ უნდა მოანესრიგოთ პიქტოგრამები კომპიუტერის სამუშაო მაგიდაზე. ამისათვის გამოიყენება კონტექსტური მენიუ, რომელიც სამუშაო მაგიდის თავისუფალ ადგილას თავის მარჯვენა ღილაკზე დაწკაპუნების შედეგად იხსნება.

მომხმარებლისთვის მოსახერხებელი რომ იყოს, ოპერაციული სისტემა Windows-ისა და მისი ელემენტების მართვის ყველა ინსტრუმენტი **Control Panel-ზე (მართვის პანელზე)** არის მოთავსებული.

სამუშაო

1

1. გახსენით მენიუ Start;
2. აირჩიეთ პუნქტი Control Panel და გახსენით შესაბამისი ფანჯარა;
3. მართვის პანელის ინსტრუმენტები კატეგორიებად (Category) არის დაჯგუფებული. გაეცანით თითოეული კატეგორიის ინსტრუმენტებს;
4. ყველა ინსტრუმენტის ერთ სიაში გამოსახვისთვის შედით განყოფილება View by-ში (დათვალიერება) და აირჩიეთ სიაში პუნქტი Large Icons (მსხვილი პიქტოგრამები) ან Small Icons (ნვრილი პიქტოგრამები);
5. დახურეთ ფანჯარა Control Panel.



ვიმსჯელოთ:

- რამდენ კატეგორიადაა დაყოფილი მართვის პანელის ინსტრუმენტები?
- რომელ კატეგორიაში უნდა იყოს სამუშაო მაგიდაზე პიქტოგრამების მოსანესრიგებელი ინსტრუმენტები?

სათანადო ინსტრუმენტების არჩევით, შესაძლებელია, მაგალითად, ოპერაციული სისტემის ენების, ფანჯრების გარეგნული ხედის მართვა, მენიუ Start-ის გამოჩენა, კომპიუტერის ახალი მომხმარებლის შექმნა. ზოგიერთი პროგრამა მართვის პანელს თავის ინსტრუმენტებს უმატებს. ამიტომ სხვადასხვა კომპიუტერში მისი შემადგენლობა სხვადასხვანაირია.

მართვის პანელის ბევრი პუნქტი, პრაქტიკულად, არ გამოიყენება, ვინაიდან გულისხმობის პრინციპით დაყენებული რეგულირება მომხმარებელთა უმრავლესობას აკმაყოფილებს. ასეთ პუნქტებს მიეკუთვნება Mouse (თაგვი), Sound (ხმა), Ease of Access Center (სპეციალური შესაძლებლობების ცენტრი). მაგრამ ისეთი პუნქტებიც არსებობს, რომლებითაც ოპერაციული სისტემის მართვისას ხშირად სარგებლობენ. ასეთია, მაგალითად, Personalization (პერსონალიზაცია). მართვის პანელის ბევრი ინსტრუმენტი, უშუალოდ პანელის გარდა, სხვა ადგილებზეც მოიპოვება. ასე მაგალითად, პუნქტ Personalization-ზე გადასასვლელად საკმარისია, თავის მარჯვენა ღელაკით საშუალო მაგიდის თავისუფალ ადგილას დავანკაპუნოთ და კონტექსტურ მენიუში იმავე სახელწოდების პუნქტი ავირჩიოთ.

კლავიატურის წყობა განსაზღვრავს იმ სიმბოლოებს, რომლებიც კლავიშებზე დაჭერისას გამოისახება ეკრანზე და ენის ყველა სიმბოლოს შეტანის საშუალებას იძლევა. კლავიატურის სასურველი წყობის დაყენების შემდეგ ტექსტის აკრეფისას, ენების პანელის დახმარებით, მარტივად შეიძლება ერთი ენიდან (მაგალითად, ინგლისურიდან) მეორეზე (მაგალითად, ქართულზე) გადასვლა.

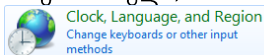
დაიმახსოვრე

**მართვის პანელი
კლავიატურის წყობა**

საქმიანობა

2

1. გახსენით მართვის პანელი;



2. კატეგორია -ში (საათი, ენა და რეგიონი) დააწკაპუნეთ მითითება Change keyboards or other input methods (კლავიატურის წყობის ან ინფორმაციის შეტანის სხვა საშუალებების წყობის შეცვლა);

3. გახსნილ დიალოგურ ფანჯარაში გადადით ჩანართ Keyboards and Languages-ზე (კლავიატურა და ენები) და დააწკაპუნეთ ღილაკი **Change keyboards...** -ზე (შეცვალეთ კლავიატურა);

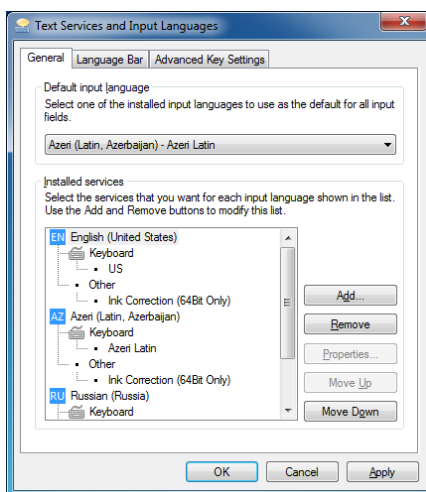
4. გახსნილი დიალოგური ფანჯრის, Text Services and Input Languages-ის (ტექსტის შეტანის ენები და მომსახურება), განყოფილება Default Input Language-ში (ტექსტის შეტანის ენა გულისხმობის პრინციპით) აირჩიეთ ენა, რომლის გამოყენებაც გსურთ გულისხმობის პრინციპით.

თუ სიაში ეს ენა არ არის, მაშინ შემდეგი ნაბიჯები გადადგით:

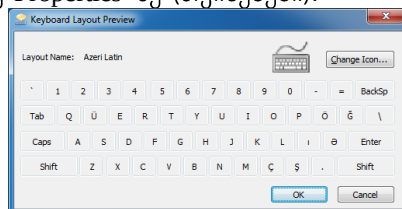
5. განყოფილება Installed services-ში (დაყენებული მომსახურების სახეები) დააწკაპუნეთ ღილაკი Add-ზე (დამატება);

6. ჩამოშლილ სიაში აირჩიეთ ჯერ ის ენა, რომლის დაყენებაც გსურთ, მაგალითად, Arabic (Saudi Arabia), შემდეგ კი აირჩიეთ პუნქტი Keyboard (კლავიატურა);

7. აირჩიეთ კლავიატურის სასურველი წყობა, მაგალითად, Arabic (101) და დაანკაპუნეთ ღილაკ OK-ზე. ენა სიას დაემატება.



8. იმისათვის, რომ ნახოთ კლავიატურის წყობა ნებისმიერი ენისათვის, დაანკაპუნეთ ღილაკ Properties-ზე (თვისებები).



შევისნავლოთ დამოუკიდებლად

გადიყვანეთ მართვის პანელი Large icons-ის (მსხვილი პიქტოგრამების) დათვალიერების რეჟიმზე. გამოიყენეთ პიქტოგრამა Mouse (თაგვი), გახსენით დიალოგური ფანჯრა Mouse Properties (თაგვის თვისებები). დაათვალიერეთ ფანჯრის ჩანართები და შეცვალეთ თაგვის პარამეტრები (მაგალითად, მაჩვენებლის გადაადგილების სიჩქარე, მისი გარეგნული ხედი).

შეამოწმა შენი ცოდნა

1. რა არის მართვის პანელი და რისგან შედგება იგი?
2. რას ნიშნავს კლავიატურის წყობა და რა წყობაა გააქტიურებული თქვენს კომპიუტერში ამ მომენტში?
3. რომელი კლავიშების კომბინაციით სარგებლობთ თქვენს კომპიუტერში ერთი ენობრივი წყობიდან მეორეზე გადასასვლელად?



- რა არის პროგრამული უზრუნველყოფა და რა სახეობად იყოფა იგი?
- რომელ პროგრამებს მიაკუთვნებენ სისტემურ პროგრამულ უზრუნველყოფას?

სისტემური პროგრამული უზრუნველყოფის უმეტეს ნაწილს **დამხმარე პროგრამები**, ანუ **უტილიტები**, შეადგენს. ისინი კომპიუტერის მომსახურების, შემონებისა და მომართვისთვის, ასევე ტექნიკური ოპერაციების შესრულებისთვისაა განკუთვნილი. ჩვეულებრივ, ამ პროგრამებს მომხმარებლები მართავენ.

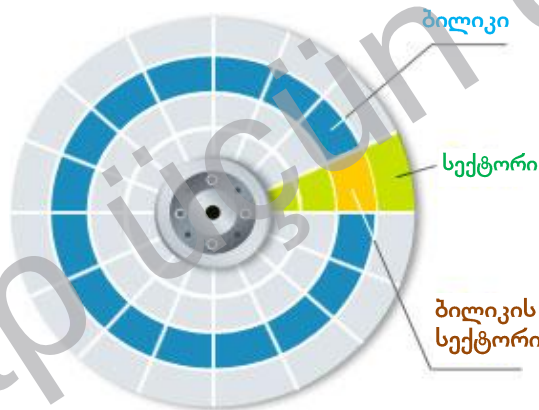
დამხმარე პროგრამების ფუნქციები მჭიდროდაა დაკავშირებული ოპერაციული სისტემის შესაძლებლობებთან, რის გამოც მათი უმრავლესობა სისტემის შემადგენლობაში შედის. მაგრამ დამოუკიდებელი დამხმარე პროგრამებიც არსებობს.

კომპიუტერში არსებული ყველა პროგრამა, მათ შორის პროგრამების მუშაობის შედეგად მიღებული მონაცემებიც, ოპერაციულ სისტემასთან ერთად, მყარ დისკზე ინახება. ამიტომ მყარი დისკი განსაკუთრებულ ყურადღებას იმსახურებს. მყარი დისკის მომსახურება დადებით გავლენას ახდენს კომპიუტერის საერთო მწარმოებლურობაზე, მისი არარსებობა კი კომპიუტერის მუშაობის სისწრაფეს აქვეითებს.

დისკის იმ ნაწილს, რომელშიც მონაცემები ინახება, **სექტორი** ჰქვია. დისკს აქვს ზედა და ქვედა მხარეები. თითოეულ მათგანზე არის წრიული **ბილიკები** და სექტორები. დაფორმატების დროს თითოეული ბილიკი გარკვეული ზომის (ჩვეულებრივ, 512 ბაიტის ტოლ) სექტორებად იყოფა. დისკის გამოყენებისას მისი ჩამწერ/წამკითხავი თავაკები ამ ბილიკებს გაივლის და ინფორმაციას სექტორში ან სექტორთა ჯგუფში ჩაწერს (ან ამოიკითხავს). ამგვარად, **სექტორი** დისკის **მეხსიერების უმცირესი უბანია, რომელსაც განსაზღვრული ზომა გააჩნია**.

დაიმახსოვრე

დამხმარე პროგრამები
სექტორი
ბილიკი
ფრაგმენტაცია
დეფრაგმენტაცია



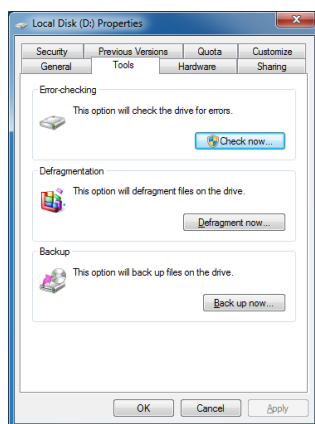
მექანიკური ზემოქმედების ან რაიმე სხვა მიზეზის გამო ზოგჯერ დისკზე დაზიანებული სექტორები ჩნდება. მათ **ცუდ სექტორებს** (bad sectors) უწოდებენ. ოპერაციულ სისტემას არ შეუძლია ინფორმაცია შესანახავად მათი გამოყენება. თუ დისკზე შენახული ფაილის ნაწილი ცუდ სექტორში აღმოჩნდება, ოპერაციული სისტემა მას ვერ წაიკითხავს და გამოიტანს შეტყობინებას შეცდომის შესახებ (File system error). Windows-ის ოპერაციული სისტემის შემადგენლობაში არსებული პროგრამა **Scan Disk**-ის დახმარებით შესაძლებელია მყარი დისკის სტრუქტურაში შეცდომის აღმოჩენა და გასწორება.

საქმიანობა

1

მყარი დისკის შემოწმება არსებული შეცდომების აღმოჩენის მიზნით

1. ორჯერ დაანკაპუნეთ სამუშაო მაგიდაზე არსებულ პიქტოგრამა Computer-ზე;
2. მარჯვენა ღილაკით დაანკაპუნეთ C დისკის (ან სხვის) პიქტოგრამაზე;
3. კონტექსტურ მენიუში აირჩიეთ ბრძანება Properties;
4. გადადით ჩანართ Tools-ში და დაანკაპუნეთ ღილაკი Check now-ზე;
5. გახსნილ დიალოგურ ფანჯარაში დაანკაპუნეთ ღილაკი Start-ზე. ჰორიზონტალური ინდიკატორი გიჩვენებთ, შემოწმების რომელი ნაწილი დამთავრდა. სამუშაოს დასასრულს კომპიუტერი გამოიტანს ინფორმაციას განსაზღვრული სტატისტიკური მონაცემების შესახებ.



ვიმსჯელოთ:

- აღმოჩნდა თუ არა დისკზე ცუდი სექტორი?
- რამდენი თავისუფალი ადგილია დისკზე?

კომპიუტერის მყარ დისკზე მილიარდობით სექტორია. ამიტომ ოპერაციული სისტემა ცალკეულ სექტორზე კი არა, სექტორთა ჯგუფზე მუშაობს. სექტორთა ჯგუფს **კლასტერს** უწოდებენ. კლასტერის ზომა მყარი დისკის ზომით (ჩვეულებრივ, 4 კბ) განისაზღვრება. კლასტერების რაოდენობა, რომლებსაც თითოეული ფაილი იკავებს, მთელი რიცხვია. მყარ დისკზე ფაილის შენახვის პროცესი ასე მიმდინარეობს: ოპერაციული სისტემა ფაილის ნაწილს ერთ თავისუფალ კლასტერში ინახავს, მეორე ნაწილს – მეორე თავისუფალ კლასტერში და ა.შ. თუ ჩანერა ნაწილობრივ შევსებულ დისკზე მიმდინარეობს, ერთი ფაილის კუთვნილი კლასტერები მიმდევრობით იწერება. თუ დისკი გადავსებულია, მასზე შესაძლოა, ვერ მოიძებნოს ფაილის დამტევი ერთიანი არე (ანუ მეზობელი კლასტერები). მიუხედავად ამისა, ფაილი მაინც ჩაიწერება, თუკი დისკზე მოიძებნება ბევრი პატარა-პატარა თავისუფალი უბანი, რომელთა საერთო ზომა ფაილის ჩასაწერად საკმარისი აღმოჩნდება. ამ შემთხვევაში ფაილი რამდენიმე ფრაგმენტად ჩაიწერება. თუ კომპიუტერი დიდი



ხნის განმავლობაში იხმარება, ბევრი ფაილი დისკზე ფრაგმენტებად გაფანტული აღმოჩნდება. ფაილების ასეთ მდგომარეობას **ფრაგმენტაცია** ეწოდება. ასეთ დანაწილებულ ფაილებზე მუშაობისას კომპიუტერის სიჩქარე მცირდება, ვინაიდან იმ კლასტერების ძებნას, რომლებზედაც ფაილები ინახება, მეტი დრო სჭირდება. ფრაგმენტაცია ოპერაციულ მეხსიერებაშიც ხდება ხოლმე, კერძოდ, მეხსიერების დაშლისა და განმენდის დროს. ფრაგმენტაციის მოსაცილებლად, ანუ დისკის **დეფრაგმენტაციისათვის**, სპეციალური უტილიტები არსებობს. დეფრაგმენტაციის დროს დისკზე გაფანტული ფაილები ახალ ადგილებზე ოპტიმალური ფორმით ჩაინერება. Windows-ის სისტემაში დისკის დეფრაგმენტაციისთვის პროგრამა **Disk Defragmenter**-ით სარგებლობენ.

საშუალო

2

მყარი დისკის დეფრაგმენტაცია

1. გახსენით მენიუ Start;
2. პუნქტ All Programs-ის არჩევის შემდეგ გადადით საქალაქე Accessories-ში;
3. გახსენით საქალაქე System Tools და აირჩიეთ პუნქტი Disk Defragmenter. გაიხსნება შესაბამისი დიალოგური ფანჯარა;
4. აირჩიეთ საჭირო დისკი და დაანკაპუნეთ ლილაკ Analyze disk-ზე. დაიწყება დისკის ანალიზი. ამ პროცესს გარკვეული დრო დასჭირდება. დაელოდეთ მის დამთავრებას;
5. თუ პროგრამა დეფრაგმენტაციის შესრულებას შემოგთავაზებთ, დაანკაპუნეთ ლილაკ Defragment disk-ზე.

დისკის დეფრაგმენტაციის ჩატარების სიხშირე დამოკიდებულია იმაზე, თუ რამდენად ინტენსიურად იყენებთ კომპიუტერს. თუ ყოველდღე მუშაობთ, მიზანშეწონილია ამის კვირაში ერთხელ გაკეთება. თუ კომპიუტერთან იშვიათად ჯდებით, მაშინ დეფრაგმენტაცია შეგიძლიათ, ორ-სამ თვეში ერთხელ ჩატაროთ.

შევისწავლეთ დამოუკიდებლად

კონტექსტური მენიუს პუნქტ Properties-ის დახმარებით, შეადარეთ ერთმანეთს რამდენიმე ფაილის ჭეშმარიტი ზომა (Size) თქვენს კომპიუტერში მათ მიერ დისკზე დაკავებულ ადგილს (Size on disk). ახსენით თითოეულ შემთხვევაში ფაილის ჭეშმარიტ ზომასა და მის მიერ დისკზე დაკავებულ ადგილს შორის არსებული განსხვავების მიზეზი.

შეამოწმეთ ცოდნა

1. რა არის სექტორი და კლასტერი?
2. როგორ ინახება ფაილები კლასტერებში დისკზე?
3. რატომ ხდება დისკზე ფრაგმენტაცია?
4. განსაზღვრეთ, რამდენ კლასტერს დაიკავებს დისკზე ფაილი, თუ მისი მოცულობა 302 კბ-ია?



- აღმოგიჩენიათ თუ არა თქვენს პერსონალურ კომპიუტერში თქვენთვის უსარგებლო პროგრამები და ფაილები? როგორ იქცევით ასეთ შემთხვევაში?
- რას ნიშნავს პროგრამის ან კომპიუტერის „დაკიდება“? ამ შემთხვევაში რას აკეთებთ ხოლმე?

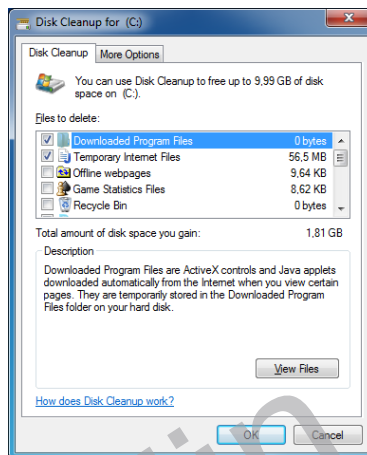
კომპიუტერით სარგებლობის პროცესში მასში აუცილებლად გროვდება სხვადასხვა სახის უსარგებლო ფაილები. ასე, მაგალითად, ვებსაიტების დათვალიერებისას შესაძლებელია დროებითი ფაილების წარმოშობა. დისკზე ზედმეტი ადგილის დამკავებელი ფაილების აღმოჩენისას ნებისმიერ მომხმარებელს უჩნდება სურვილი, წაშალოს ისინი. მათგან დისკის გასაწმენდად გათვალისწინებულია დამხმარე პროგრამა **Disk Cleanup**.

საქმიანობა

1

უსარგებლო ფაილებისგან მყარი დისკის გაწმენდა

1. გახსენით მენიუ Start;
2. აირჩიეთ ბრძანება All Programs⇒Accessories⇒System Tools⇒Disk Cleanup.
გაიხსნება შესაბამისი დიალოგური ფანჯარა;



3. თუ სისტემაში ერთზე მეტი დისკია, აირჩიეთ გასაწმენდი დისკი; სხვა შემთხვევაში გადადით შემდეგ პუნქტზე. სიაში საჭირო დისკის ასარჩევად ისარგებლეთ ისრისანი ლილაკით;
4. დაანკაპუნეთ ლილაკ OK-ზე. პროგრამა დაითვლის და მიგიითითებთ, რა ადგილის გათავისუფლებაა შესაძლებელი დისკზე;
5. მონიშნეთ ის ფაილები, რომელთა წაშლაც გსურთ, და დაანკაპუნეთ ლილაკ OK-ზე.

ვიმსჯელოთ:

- რა ადგილი უკავია ინტერნეტის დროებით ფაილებს (Temporary Internet Files) დისკზე, რომლის გაწმენდაც გინდათ?
- რა ადგილი გათავისუფლდა დისკზე?

მიზანშეწონილად ითვლება დისკების გამწმენდი პროგრამის თვეში ერთხელ გაშვება. დიალოგურ ფანჯარა Disk Cleanup-ში არის გვერდი More Options, სადაც შეგიძლიათ, ნახოთ სისტემის მიერ გამოუყენებელი კომპონენტების მოსაძებნი რამდენიმე პარამეტრი.

თქვენს კომპიუტერში ოდესღაც ჩანერილი პროგრამები გარკვეული ხნის შემდეგ შეიძლება აღარ დაგჭირდეთ და მათი ნაშლა მოისურვოთ.

ღიათაქსოვრა

- Disk Cleanup
- Task Manager

ყურადღება!

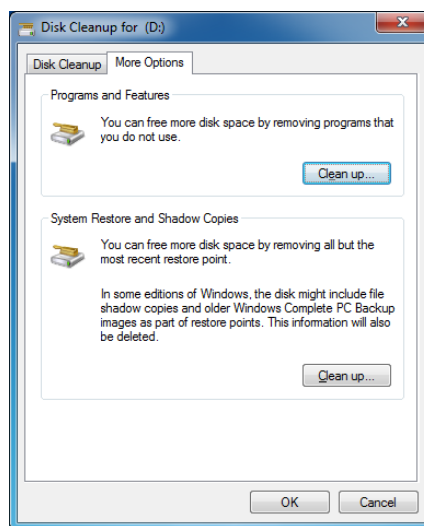
არასოდეს ნაშალოთ პროგრამები, რომელთა დანიშნულება ზუსტად არ იცით!

საქმიანობა

2

დაყენებული პროგრამების ნაშლა

1. გახსენით პროგრამა Disk Cleanup;
2. გადადით გვერდ More Options-ზე;



3. განყოფილება Programs and Features-ში დაანკაპუნეთ ღილაკ Clean up-ზე. გაიხსნება თქვენს კომპიუტერში დაყენებული პროგრამების სია;
4. დაანკაპუნეთ იმ პროგრამის სახელწოდებაზე, რომლის ნაშლაც გსურთ. პროგრამის სახელი მონიშნება, მის ქვეშ კი მისი აღწერილობა გამოჩნდება;
5. საჭირო პროგრამის მონიშვნის შემდეგ დაანკაპუნეთ ღილაკ Uninstal/Change-ზე. ეკრანზე გამოჩნდება შეტყობინება ნაშლის ოპერაციის დადასტურების შესახებ;
6. თუ ნამდვილად გსურთ მონიშნული პროგრამის სისტემიდან ამოშლა, დაანკაპუნეთ ღილაკ OK-ზე, თუ არა და – ღილაკ Cancel-ზე.

ზოგჯერ ისე ხდება, რომ პროგრამა, რომელზედაც მუშაობთ, თქვენს ბრძანებებზე აღარ პასუხობს. კლავიატურის კლავიშებზე დაჭერას არავი-

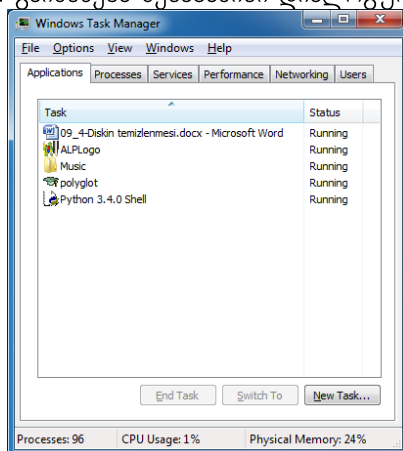
თარი შედეგი აღარ მოაქვს. ასეთ შემთხვევებში ამბობენ, რომ პროგრამა „დაეკიდა“. ოპერაციული სისტემა ცდილობს, თვითონ გადაწყვიტოს ეს პრობლემა. მაგრამ თუ ლოდინის დრო არ გაქვთ, დამხმარე პროგრამა Task Manager-ის დახმარებით, შეგიძლიათ, დაასრულოთ დაკიდებული პროგრამის მუშაობა. თუ ამ მეთოდს დროის დასაზოგავად მიმართავთ, გაითვალისწინეთ, რომ ყველა დაუმასხვრებელი მონაცემი დაგეკარგებათ. მაგრამ თუ ის დოკუმენტი, რომელზეც მუშაობდით, თქვენთვის მნიშვნელოვანია, დაიცადეთ რამდენიმე ნუთი, ვიდრე ოპერაციული სისტემა თავად გადაწყვეტდეს ამ პრობლემას.

საქმიანობა

3

დაკიდებული პროგრამის მუშაობის დასრულება

1. გახსენით ამოცანათა პანელის კონტექსტური მენიუ. ამისათვის თავგის მარჯვენა ღილაკით დაანკაპუნეთ პანელის თავისუფალ ადგილას. მენიუში არჩიეთ პუნქტი Start Task Manager (ან დააჭირეთ კლავიშების კომბინაცია <Ctrl+Shift+Esc>-ს). გაიხსნება შესაბამისი დიალოგური ფანჯარა;



2. გადადით ჩანართ Applications-ში. აქ გამოისახება იმ პროგრამების სია, რომლებიც მოცემულ მომენტში სრულდება. ყურადღება მიაქციეთ პროგრამების მდგომარეობას (სტატუსს). მომუშავე პროგრამებს, როგორც წესი, აქვს სტატუსი Running („მუშაობს“), ხოლო დაკიდებულ პროგრამებს – Not responding („არ პასუხობს“);
 3. გახსენით პროგრამა Paint და დააკვირდით ჩანართ Applications-ში მიმდინარე ცვლილებებს;
 4. მოძებნეთ სიაში სტრიქონი პროგრამა Paint-ის სახელწოდებით და დაანკაპუნეთ ღილაკი End Task-ზე. პროგრამის შესრულება შეწყდება და სიაში მისი სახელწოდება წაიშლება.
- გაითვალისწინეთ, რომ ეს ოპერაცია, როგორც წესი, დაკიდებულ პროგრამებთან მიმართებით სრულდება.

შავისწავლოთ დამოუკიდებლად

პროგრამა Task Manager-ის გვერდ Processes-ის (პროცესები) დახმარებით დაადგინეთ, რამდენი პროცესი სრულდება მოცემულ მომენტში თქვენი კომპიუტერის ოპერაციულ მესხიერებაში, მესხიერების რა ნაწილს იკავებს ისინი ერთად, რომელ პროცესს (პროგრამას) უჭირავს მეტი ადგილი. ჩართეთ რომელიმე ვებბრაუზერი და დააკვირდით, როგორ იცვლება პროცესების რაოდენობა.

Name	Status	2% CPU	19% Memory	0% Disk	0% Network
Apps (2)					
Google Chrome		0%	54.6 MB	0 MB/s	0 Mbps
Task Manager		0.1%	12.9 MB	0 MB/s	0 Mbps
Background processes (57)					
ABBY network license server (3...		0%	11.3 MB	0 MB/s	0 Mbps
ACMON (32 bit)		0%	0.1 MB	0 MB/s	0 Mbps
ActivateDesktop		0%	0.6 MB	0 MB/s	0 Mbps
Adobe Acrobat Update Service (...)		0%	0.5 MB	0 MB/s	0 Mbps

შეამოწმა შენი ცოდნა



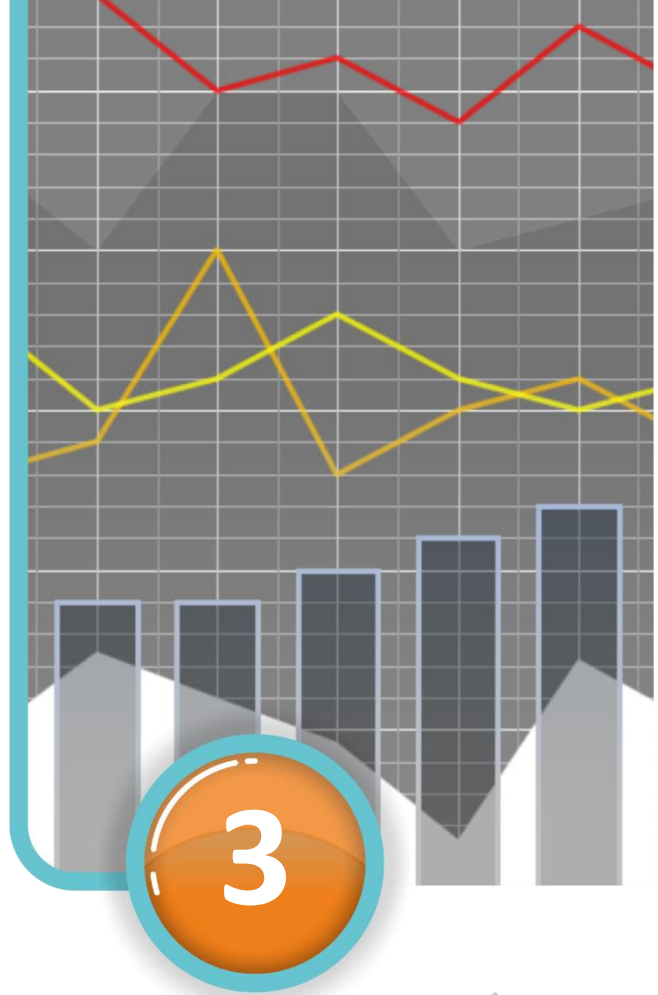
1. რის შედეგად ჩნდება კომპიუტერში უსარგებლო ფაილები?
2. ახსენით პროგრამა Disk Cleanup-ის დანიშნულება.
3. როგორ დავრწმუნდეთ, რომ მომუშავე პროგრამა დაეკიდა?
4. რა გზით შეიძლება კომპიუტერის ოპერაციული მესხიერებიდან დაკიდებული პროგრამის ამოშლა?

შემაჯავებელი კითხვები და დავალებები

1. დაასახელეთ კომპიუტერის ძირითადი მახასიათებლები. როგორ შეიძლება კომპიუტერის პარამეტრების დადგენა?
2. რა როლს ასრულებს მართვის პანელი Windows-ის ოპერაციულ სისტემაში? შესაძლებელია თუ არა, რომ კომპიუტერს მართვის პანელი არ გააჩნდეს? ახსენით თქვენი პასუხი.
3. რა ჰქვია დისკზე მეხსიერების გარკვეული მოცულობის უმცირეს ბლოკს?
4. მყარ დისკზე ჩანერილია 305 კბ, 604 კბ და 2340 კბ-ის მოცულობის სამი ფაილი. რომელი ფაილის ზომა იქნება დისკზე დაკავებული ადგილის შესაბამისი?
5. რისთვისაა განკუთვნილი პროგრამა Task Manager?
6. პროგრამული უზრუნველყოფის რომელ სახეს მიეკუთვნება პროგრამები Task Manager და Disk Cleanup?
7. თუ სახლში კომპიუტერი გაქვთ, ჩაატარეთ მისი ერთ-ერთი დისკის ფრაგმენტაციის ანალიზი. განსაზღვრეთ დისკის სრული მოცულობა, მეხსიერების თავისუფალი ნაწილის მოცულობა, თავისუფალი ადგილის პროცენტული რაოდენობა, კლასტერის ზომა.
8. რომელი გამონათქვამია მცდარი?
 - ა) პროცესორის ტაქტების სიხშირე ჰერცებით იზომება;
 - ბ) ქეშ-მეხსიერება კომპიუტერის მყარ დისკზეა მოთავსებული;
 - გ) ოპერაციული მეხსიერების მოცულობა მყარი დისკის მოცულობაზე მეტია;
 - დ) დამხმარე პროგრამები კომპიუტერის სისტემურ პროგრამებს მიეკუთვნება;
 - ე) პროგრამა Task Manager-ის დახმარებით პროგრამის მუშაობის შეჩერება შეიძლება;
 - ვ) პროგრამა Scan Disk კომპიუტერს უსარგებლო ფაილებისგან ათავისუფლებს.
9. შეადარეთ ორი კომპიუტერის პარამეტრები. რომლის პარამეტრებს აქვს უპირატესობა? ახსენით თქვენი პასუხი:

Notebook X1/1920×1080/Intel® Core™ i7/2.5 GHz /RAM 16Gb/ HDD 256 Gb/Veb-kamera 1.3 Mpiksel/Windows 8.1 (64-bit)/25.4×38.1×3.3 sm/2.7 kg

Notebook X2/1366×768/Intel® Celeron®/2.16 GHz/ RAM 4Gb/HDD 1500Gb/Veb-kamera 0.3 Mpiksel/Windows 8.1 (64-bit)/38.1×25.9×2.54 sm/2.6 kg



გამოყენებითი პროგრამები

10. რთულ დოკუმენტებზე მუშაობა
11. მულტიმედიური პრეზენტაციები
12. ელექტრონულ ცხრილებზე მუშაობა
13. დიაგრამები ელექტრონულ ცხრილში



- რას ეწოდება ტექსტრი დოკუმენტი?
- რა სახის ობიექტები შეიძლება იყოს ტექსტურ დოკუმენტში?

საქმიანობა

1

სიტბოს იმ რაოდენობას, რომელიც ლღობის ტემპერატურამდე მიყვანილ კრისტალურ ნივთიერებას მყარი კრისტალური მდგომარეობიდან თხევად მდგომარეობაში გადასასვლელად უნდა მიეწოდოს, დნობის სიტბო ეწოდება. დნობის სიტბო დამოკიდებულია კრისტალის მასასა და სახეობაზე:

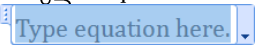
$$Q = \lambda m.$$

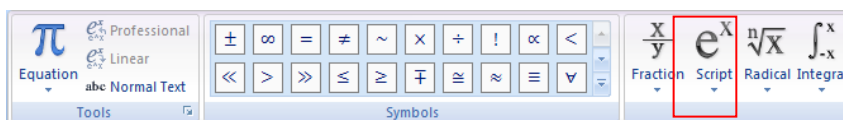
სადაც Q – დნობის სიტბოა, m – სხეულის მასა, ხოლო λ – **დნობის კუთრი სიტბო**. ნივთიერების დნობის კუთრი სიტბო არის ფიზიკური სიდიდე, რომელიც გამოიყენება ლღობის ტემპერატურამდე მიყვანილი 1 კგ მასის მქონე სხვადასხვა მყარი სხეულის სრულ გალღობაზე დახარჯული სიტბოს რაოდენობებს შორის სხვაობის განსაზღვრავად.

ფორმულა	საზომი ერთეული
$\lambda = \frac{Q}{m}$	$[\lambda] = 1 \frac{\text{ჯ}}{\text{კგ}}$
განსაზღვრება	განსაზღვრება
ნივთიერების დნობის კუთრი სიტბო – სიტბოს იმ რაოდენობის ტოლი ფიზიკური სიდიდე, რომელიც საჭიროა ლღობის ტემპერატურამდე მიყვანილი 1 კგ მასის კრისტალური ნივთიერების გასაღნობად.	1 ჯოული 1 კილოგრამზე – ეს არის დნობის ისეთი კუთრი სიტბო, როდესაც 1 კგ კრისტალური ნივთიერების გასაღნობად 1 ჯოული სიტბო იხარჯება.

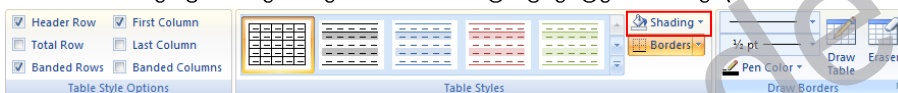
შექმენით ასეთი დოკუმენტი ტექსტურ რედაქტორში (Microsoft Word-ში ან OpenOffice.org Writer-ში). ამისათვის შეასრულეთ შემდეგი ალგორითმი:

1. გახსენით პროგრამა Microsoft Word;
2. შეიტანეთ და დააფორმატეთ შემდეგი ტექსტი: **სიტბოს იმ რაოდენობას, რომელიც უნდა მიეწოდოს ლღობის ტემპერატურამდე მიყვანილ კრისტალურ ნივთიერებას მყარი კრისტალური მდგომარეობიდან თხევად მდგომარეობაში გადასასვლელად, დნობის სიტბო ეწოდება.** დნობის სიტბო კრისტალის მასასა და სახეობაზეა დამოკიდებული.

3. გადადით ახალ ხაზზე და შეიტანეთ ფორმულა $Q = \lambda m$. ბერძნული ალფაბეტის ასოს λ -ს შესატანად შრიფტების სიაში აირჩიეთ Symbol და დააჭირეთ კლავიატურის ასო <L>-ს (კლავიატურის ინგლისური წყობისას).
ლილაკ -ის დახმარებით მოათავსეთ ფორმულა სტრიქონის ცენტრში;
4. შეიტანეთ მოცემული დოკუმენტის მომდევნო აბზაცი;
5. ცხრილის შესატანად ისარგებლეთ მენიუ Insert-ის ლილაკ -ით. ააგეთ ცხრილი, რომელიც ორი სვეტისა და ოთხი სტრიქონისგან შედგება;
6. შეავსეთ ცხრილის უჯრები ისე, როგორც ნახატზეა ნაჩვენები;
7. ფორმულის შესატანად აირჩიეთ მენიუ Insert და დაანკაპუნეთ ლილაკ -ზე. გაიხსნება ინსტრუმენტების პანელი Equation tools და დოკუმენტში გაჩნდება ფორმულის ჩასანერი  კელი;

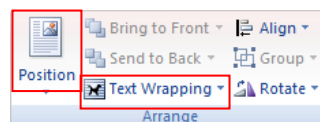


8. წილადების შესატანად დაანკაპუნეთ პანელ Equation tools-ის ლილაკ Fraction-ზე. სიაში აირჩიეთ ჩვეულებრივი წილადის ვარიანტი. კურსორის შესაბამისად მრიცხველსა და მნიშვნელში მოთავსებით, შეიტანეთ ფორმულის სათანადო სიმბოლოები;
9. ცხრილის უჯრებში ჩანაწერები ნიმუშის მიხედვით განლაგეთ: პირველი სამი სტრიქონის ჩანაწერები გაასწორეთ ცენტრის მიმართ, ბოლო სტრიქონისა კი – მარცხენა კიდის მიმართ. დააფორმატეთ შესაბამისი ჩანაწერები მუქი შრიფტით გამოყოფით;
10. შეცვალეთ ცხრილის პირველი და მესამე სტრიქონების უჯრების ფერი. ამისათვის დაანკაპუნეთ ფანჯრის სათაურის ზოლში განთავსებულ ლილაკ Table Tools-ზე. გაიხსნება შესაბამისი ინსტრუმენტების პანელი;



11. სია Shading-ში აირჩიეთ შესაბამისი ფერი. არჩეული ფერი შეეღებავს ცხრილის გააქტიურებულ (რომელშიც კურსორია ჩასმული) უჯრას;
12. დაარქვით ფაილს სათანადო სახელი (მაგალითად, **დნობის სითბო**) და შეინახეთ დოკუმენტი.

ზოგჯერ შეიძლება არ მოგეწონოთ ნახატის მდებარეობა ტექსტში. მაგალითად, თქვენ გსურთ, რომ ნახატი ორ აბზაცს შორის კი არა, ტექსტის შუაში იყოს ჩასმული, ან გვერდის მარჯვენა ზედა კუთხეში მდებარეობდეს. ალბათ, გინახავთ ასეთი განლაგება სახელმძღვანელოებში ან სხვა ბეჭდურ გამოცემებში. გამოსახულებების დოკუმენტის სათანადო ადგილას განსათავსებლად ტექსტურ რედაქტორ Word-ში სახატავი ინსტრუმენტების პანელზე



(Picture Tools) ორი ინსტრუმენტი: Position და Text Wrapping არის გათვალისწინებული.



სახატავი ინსტრუმენტების პანელის გამოსაჩენად აუცილებელია გრაფიკულ გამოსახულებაზე ორჯერ დაწკაპუნება.

ტესტურ რედაქტორში არსებობს დოკუმენტისთვის განსაკუთრებული ეფექტების მქონე დეკორატიული ტექსტების დამატების შესაძლებლობა.

ამისათვის გათვალისწინებულია ინს-



ტრუმენტი WordArt. მისი დახმარებით შესაძლებელია ტექსტის შემოტრიალება, გადახრა, მისთვის ჩრდილის მინიჭება და სხვა ეფექტების გამოყენება. მენიუ Insert-ში ამ ლილაკზე დაწკაპუნებით გახსნილ ფანჯარაში ნებისმიერი თქვენთვის სასურველი შაბლონის არჩევა შეგიძლიათ.

საქმიანობა

2

მოცემული ნიმუშის მოსამზადებლად შეასრულეთ მითითებული ნაბიჯები.

ხმელეთის ძუძუმწოვრები

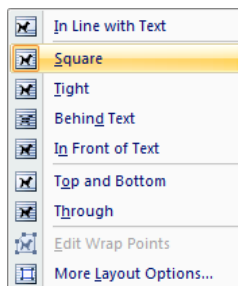
ხმელეთის ძუძუმწოვრები ძირითადად ტყეებსა და გაშლილ სივრცეებში ბინადრობენ. მათ პროპორციული აგებულების ძლიერი სხეული, კარგად განვითარებული გრძელი კიდურები და კუნთოვანი კისერი აქვთ. ისინი სიარულით, სირბილითა და ხტუნვით გადაადგილდებიან.

ხმელეთის ცხოველთა უმრავლესობას ბალახისმჭამელები შეადგენენ, მაგალითად, წყვილჩლიქოსანთა წარმომადგენლები – ირმები, ანტილოპები, ჟირაფები, თხები, ცხვრები და სხვ. მათვე მიეკუთვნებიან აგრეთვე კენტჩლიქოსანთა წარმომადგენლები (ცხენი, ზებრა, ვირი) და კურდღლისებრნი.



1. შექმენით ახალი ფაილი ტექსტურ რედაქტორში;
2. მენიუ Insert-ში დააწკაპუნეთ ლილაკ WordArt-ზე და გახსნილ ფანჯარაში ნებისმიერი შაბლონი აირჩიეთ;
3. გახსნილ ფანჯარაში, ველ Text-ში ჩანერეთ ტექსტის სათაური (ხმელეთის ძუძუმწოვრები) და დააწკაპუნეთ ლილაკ OK-ზე;
4. ახალი ხაზიდან აკრიფეთ შემდეგი ტექსტი:
 ხმელეთის ძუძუმწოვრები ძირითადად ტყეებსა და გაშლილ სივრცეებში ბინადრობენ. მათ პროპორციული აგებულების ძლიერი სხეული, კარგად განვითარებული გრძელი კიდურები, კუნთოვანი კისერი აქვთ. ისინი სიარულით, სირბილითა და ხტუნვით გადაადგილდებიან.
 ხმელეთის ცხოველთა უმრავლესობას ბალახისმჭამელები შეადგენენ, მაგალითად, წყვილჩლიქოსანთა წარმომადგენლები – ირმები, ანტილოპები, ჟირაფები, თხები, ცხვრები და სხვ. მათვე მიეკუთვნებიან აგრეთვე კენტჩლიქოსანთა წარმომადგენლები (ცხენი, ზებრა, ვირი) და კურდღლისებრნი.
5. მოძებნეთ (ვთქვათ, ინტერნეტში) ამ ტექსტის შესაბამისი სურათი და ჩასვით იგი დოკუმენტში;

6. გადაიტანეთ თავისი მაჩვენებელი ნახატზე და დაანკაპუნეთ მარჯვენა ლილაკზე;
7. გახსნილ კონტექსტურ მენიუში აირჩიეთ სტრიქონი Text Wrapping, ხოლო ქვემენიუში – პუნქტი Square;
8. გადაადგილეთ ნახატი დოკუმენტის სათანადო ადგილას;
9. შეინახეთ დოკუმენტი შესაფერისი სახელით.



შევისწავლოთ დამოუკიდებლად

გაეცანით სახელმძღვანელოში თემა „ინფორმაციის გადაცემის“ საქმიანობის ბლოკში მოცემულ დავალებას (გვ.100). ამ თემაზე შექმენით ტექსტურ რედაქტორში ერთგვერდიანი დოკუმენტი. ჩასვით მასში მითითებული ნახატები. ამ ნახატებით სარგებლობისათვის გადმოქაჩეთ ინტერნეტიდან (საიტ derslik.edu.az-იდან ან informatik.az-იდან) ამ სახელმძღვანელოს ელექტრონული ვერსია ან მე-6 კლასის ინფორმატიკის სახელმძღვანელო. მოძებნეთ ის გვერდები, რომლებზეც ეს ნახატებია, შეინახეთ ისინი ნებისმიერი საშუალების გამოყენებით და ყოველი სურათი ტექსტის შესაბამის ადგილას ჩასვით.

შეამოწმა შენი ცოდნა ?

1. როგორ განვათავსოთ ნახატები ტექსტის სხვადასხვა ადგილას?
2. რა შემთხვევებში იყენებენ ინსტრუმენტ WordArt-ს?
3. ტექსტურ რედაქტორ WordArt-ის შესაძლებლობების გამოყენებით, მოამზადეთ ერთ-ერთი ამ დოკუმენტთაგანი: სიგელი, დიპლომი ან სერთიფიკატი.
4. მოამზადეთ ტექსტურ რედაქტორში ქვემოთ მოცემული დოკუმენტი და შეასრულეთ მათემატიკის სახელმძღვანელოდან აღებული დავალება.

შეავსეთ ცხრილი ნამრავლის ზარისხის სახით ჩანერჩით

$(-3,2x)^2 \cdot (-3,2x)^4 =$	$3^7 \cdot 3^5 \cdot 3^2 =$	$(-0,6)^4 \cdot (-0,6) =$
$(a-b)^5 \cdot (a-b)^8 =$	$7^2 \cdot 7^4 \cdot 7 \cdot 7^8 =$	$16^2 \cdot 16 \cdot 16 \cdot 16^5 =$
$(x+2y)^9 \cdot (x+2y)^{10} =$	$12^4 \cdot 12 \cdot 12^3 =$	$2,3 \cdot 2,3^8 \cdot 2,3^6 \cdot 2,3 =$
$\left(\frac{3}{4}x\right)^{11} \cdot \left(\frac{3}{4}x\right)^8 \cdot \left(\frac{3}{4}x\right)^8 =$	$x^{11} \cdot x^8 \cdot x^2 \cdot x^3 =$	$(1,(5))^3 \cdot \left(\frac{14}{9}\right)^8 =$



- რა არის მულტიმედია?
- როგორ ამატებენ ხმასა და ვიდეოს OpenOffice Impress-ში შექმნილ პრეზენტაციას?

უმცროს კლასებში თქვენ შეისწავლეთ ელექტრონული პრეზენტაციების შექმნა, სლაიდებისთვის ტექსტების, გრაფიკის, ნახატების, ხმოვანი და ვიდეოფაილების, ცხრილების, დიაგრამებისა და სხვა ობიექტების დამატება, სლაიდების ობიექტებისთვის ანიმაციის ეფექტების მინიჭება. ამ სამუშაოს თქვენ ძირითადად პროგრამა **OpenOffice Impress**-ში ატარებდით. ახლა კი ცოდნასა და უნარებს Microsoft Offices-ს პაკეტში შემავალ სხვა ფართოდ გავრცელებულ პროგრამაში, **Power Point**-ში, მუშაობით გაიმყარებთ.

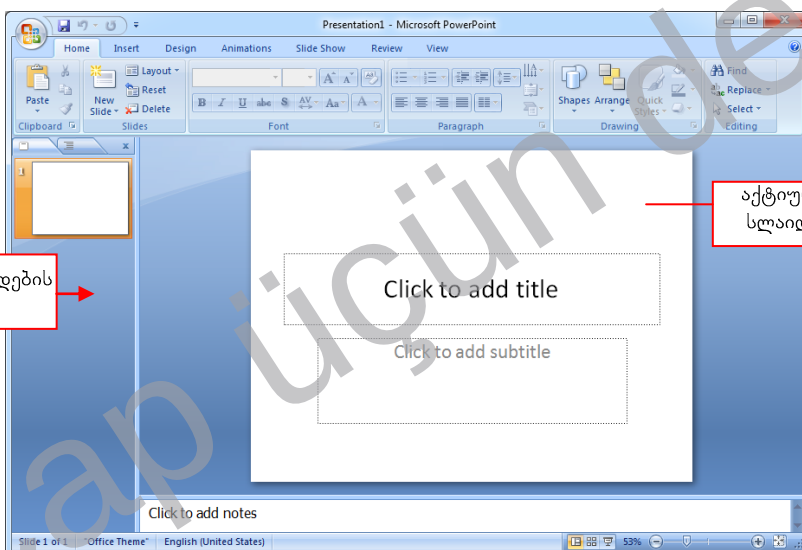
საქმიანობა

1

მულტიმედია პრეზენტაციის შექმნა პროგრამა Microsoft Power Point-ში

პრეზენტაციის შექმნაზე მუშაობის დაწყებამდე შეასრულეთ შემდეგი: შექმენით კომპიუტერში ახალი საქალაქე „აზერბაიჯანელი პოეტები“; ინტერნეტში მოძებნეთ და გადმოქაჩეთ გამოსახულებები ხუთი ცნობილი აზერბაიჯანელი პოეტის პორტრეტებით, პატარ-პატარა ნაწყვეტები მათი ლექსებიდან და ეს ყველაფერი შესაბამის საქალაქეში შეინახეთ. ასევე ინტერნეტში ან ვიდეოფაილებიდან დისკზე მოძებნეთ ვიდეორგოლი ამ პოეტებიდან რომელიმეს შესახებ და მასაც იმავე საქალაქეში მიუჩინეთ ბინა.

1. გაუშვით პროგრამა Microsoft Power Point. გაიხსნება პროგრამის მთავარი ფანჯრა და მასთან ერთად, ახალი ფაილი, რომელიც ერთი სლაიდისგან შედგება. როგორც წესი, ფანჯრის შუაში **აქტიური სლაიდი** თავსდება. ყველა სლაიდს (ამასაც და იმ ახლებსაც, რომლებსაც შემდეგ შექმნით) დაინახავთ ფანჯრის მარცხენა ნაწილში მდებარე სლაიდების ველში.



2. კლავიშ <Enter>-ის ან ბლოკ **Slides**-ის ინსტრუმენტების პანელის ღილაკ

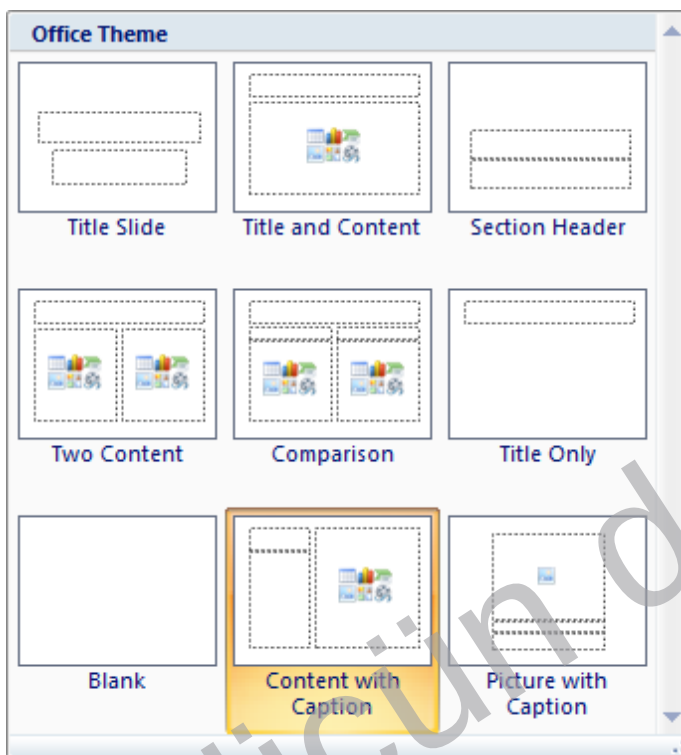


-ის გამოყენებით შექმენით რამდენიმე (5-6) ახალი სლაიდი;

3. გააქტიურეთ პირველი სლაიდი. ჩვეულებრივ, ეს სლაიდი სატიტულო სლაიდად გვევლინება. მასზე იწერება პრეზენტაციის სახელწოდება და ინფორმაცია ავტორის შესახებ. ველ Click to add title-ში მიუთითეთ პრეზენტაციის სათაური (**აზერბაიჯანელი პოეტები**). თუ გსურთ, რომ სათაური უფრო მიმზიდველი იყოს, ისარგებლეთ WordArt-ის ინსტრუმენტებით. მაგალითისთვის, სათაური შეიძლება ასეთი გამოგვიდეთ:

აზერბაიჯანელი პოეტები

4. ველ Click to add subtitle-ში შეიტანეთ ინფორმაცია თქვენ შესახებ;
5. გააქტიურეთ მეორე სლაიდი და მისთვის სხვა მაკეტი აირჩიეთ. ამისათვის ბლოკ **Slides**-ის ინსტრუმენტების პანელზე დააწკაპუნეთ ღილაკ -ზე. გამოისახება სლაიდების სხვადასხვა მაკეტი.



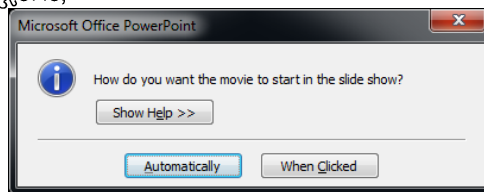
6. აირჩიეთ მაკეტი **Content with Caption**. ჩვეულებრივ, ამ მაკეტს ერთ სლაიდზე ტექსტისა და ნახატის განსათავსებლად იყენებენ;
7. ველ Click to add title-ში მიუთითეთ სლაიდის სახელწოდება – ერთ-ერთი პოეტის სახელი. ჩაწერეთ ველ Click to add text-ში ამ პოეტის ლექსის ერთი სტროფი.
8. სლაიდისთვის იმ ნახატის დასამატებლად, რომელზედაც პოეტის

პორტრეტია გამოსახული, ისარგებლეთ ღილაკთა ჯგუფ -ით. აირჩიეთ

მათში ღილაკი , გადადით გრაფიკული გამოსახულებების (აზერბაიჯანელი პოეტების) შემცველ საქალაქში და აირჩიეთ შესაბამისი ნახატი;

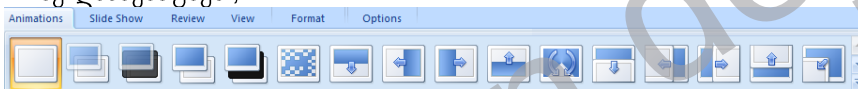
9. გააქტიურეთ მესამე სლაიდი. აირჩიეთ მისთვის მეორე სლაიდის ანალოგიური მაკეტი. ეს სლაიდი იმ პოეტს მიუძღვნით, რომლის ვიდეორგოლიც გაქვთ. სლაიდის სათაურში ამ პოეტის სახელი შეიტანეთ, ხოლო ლექსისთვის განკუთვნილ ველში შესაბამისი ვიდეო მოათავსეთ. ამისათვის მენიუ Insert-ში

დააწკაპუნეთ ღილაკ  movie-ზე. მოძებნეთ შესაბამისი საქალაქში საჭირო ვიდეოფაილი და დაამატეთ იგი სლაიდს. ეკრანზე გამოჩნდება დიალოგური ფანჯარა;



თუ გსურთ, რომ სლაიდის გახსნისას ვიდეო ავტომატურად გავიდეს, აირჩიეთ ვარიანტი Automatically. თუ ვარიანტ When Clicked-ს აირჩევთ, მაშინ ვიდეოს გასაშვებდ თავის მარცხენა ღილაკით უნდა ისარგებლოთ. ჩვეულებრივ, პრეზენტაციის დემონსტრირების დროს უპირატესობას მეორე ვარიანტს ანიჭებენ. სლაიდზე ნახატის სახით გამოტანილი იქნება ვიდეოფაილის კადრი. ნახატის ზომას და მის ადგილს სლაიდზე თქვენი სურვილის მიხედვით დაარეგულირებთ;

10. ანალოგიურად შეავსეთ დანარჩენი სლაიდები;
11. ერთი სლაიდიდან მეორეზე გადასვლის ეფექტის გამოსაყენებლად აირჩიეთ მენიუ Animations. გადატარეთ თავისი მაჩვენებელი შემოთავაზებული ეფექტების ჩამონათვალზე და მოსალოდნელი ეფექტის წინასწარ სანახავად ცოტა დაიცადეთ. აირჩიეთ გადასვლის სასურველი ეფექტი და თუ მის გამოყენებას ყველა სლაიდისთვის გადანყევთ, ამავე პანელზე ღილაკ Apply to All-ზე დააწკაპუნეთ;



12. მენიუ Insert-ში დააწკაპუნეთ ღილაკ -ზე (Header and Footer). გახსნილ დიალოგურ ფანჯარაში აღნიშნეთ ჩანაწერი Slide Number და დააწკაპუნეთ ღილაკ Apply to All-ზე. პრეზენტაციაში სლაიდები თანამიმდევრობით გადაინომრება, დანეჭბული პირველი სლაიდიდან;

13. გადადით მენიუ Slide Show-ზე და დააწკაპუნეთ ღილაკ -ზე. დაიწყება მომზადებული პრეზენტაციის სლაიდების თანამიმდევრული დემონსტრირება;
14. სლაიდების დემონსტრირების დასასრულებლად და ჩვეულებრივ რეჟიმზე გადასასვლელად დააჭირეთ ღილაკ <Esc>-ს;
15. დაარქვით ფაილს შესაბამისი სახელი და შეინახეთ იგი საქალაქში „აზერბაიჯანელ პოეტებში“.

პრეზენტაციის შექმნის შემდეგ ან მასზე მუშაობის პროცესში ზოგჯერ საჭირო ხდება სლაიდების თანამიმდევრობის შეცვლა. სლაიდებზე მუშაობა, როგორც წესი, ნორმალური დათვალიერების რეჟიმში (Normal View) მიმდინარეობს. მაგრამ პრეზენტაციაზე სრული წარმოდგენის შესაქმნელად ეს რეჟიმი საკმარისი არ არის. ამისათვის შესაძლებელია სლაიდების დახარისხების რეჟიმით (Slide Sorter View) სარგებლობა. ამ რეჟიმზე გადასასვლელად, პროგრამის ფანჯრის ქვედა ნაწილში არსებული სტატუსის  პანელის ლილაკებიდან უნდა ავირჩიოთ ლილაკი

დაიმახსოვრე

**სლაიდების ველი
აქტიური სლაიდი**

Slide Sorter View ან მენიუ View-ში დავანკაპუნოთ ლილაკ  -ზე (Slide Sorter). ამ რეჟიმში შეგვიძლია, მარტივად შევუცვალოთ ადგილები სლაიდებს, დავამატოთ ახალი სლაიდები და წავშალოთ ზოგიერთი მათგანი.

- **სლაიდების თანამიმდევრობის შეცვლა.** პრეზენტაციაში სლაიდისთვის ადგილის შესაცვლელად, დააყენეთ თავისი მაჩვენებელი ამ სლაიდზე და, მარცხენა ლილაკიდან თითის აულებლად, გადაადგილეთ მაჩვენებელი ახალ ადგილას. შემდეგ აიღეთ თითი ლილაკიდან.
- **სლაიდის წაშლა.** სლაიდის წასაშლელად, მონიშნეთ იგი და დააჭირეთ კლავიმ <Delete>-ს ან <Backspace>-ს. ამავე მიზნით შესაძლებელია ასევე ინსტრუმენტების პანელის ლილაკ  -ის გამოყენება.
- **ახალი სლაიდის დამატება.** პრეზენტაციისთვის ახალი სლაიდის დასამატებლად, ჯერ მონიშნეთ ის სლაიდი, რომლის შემდეგაც უნდა ჩასვთ ახალი, და დაანკაპუნეთ ლილაკ New Slide-ზე. ახალი სლაიდის გასარედაქტირებლად გადადით ჩვეულებრივ რეჟიმზე.

შევისწავლოთ დამოუკიდებლად

პრეზენტაცია რომ უფრო მიმზიდველი გახადონ და ამავე დროს მსმენელს ინფორმაცია ნაწილ-ნაწილ და გარკვეული თანამიმდევრობით მიანოდონ, სლაიდის ზოგიერთ ელემენტს ანიმაციის ეფექტს ანიჭებენ. გასულ წელს თქვენ ამის პროგრამა OpenOffice Impress-ში გაკეთება ისწავლეთ. ბუნებრივია, ანალოგიური სამუშაოს შესრულება პროგრამა Microsoft Power Point-შიც შეიძლება. მენიუ Animations-ში დაანკაპუნეთ ლილაკ  -ზე და შეისწავლეთ იმავე სახელწოდების პანელი, რომელიც პროგრამის ფანჯრის მარჯვენა ნაწილში გაჩნდება. გამოიყენეთ სხვადასხვა ანიმაციური ეფექტი თქვენ მიერ შექმნილი პრეზენტაციის, აზერბაიჯანელი პოეტების, სლაიდების ობიექტების მიმართ.

შეამოწმა შენი ცოდნა

1. რა იგულისხმება პრეზენტაციის მულტიმედია ობიექტებში?
2. როგორ დავამატოთ პრეზენტაციას ხმა და ვიდეოფაილი?
3. შექმენით პრეზენტაცია ნებისმიერ თემაზე. გამოიყენეთ პრეზენტაციის ობიექტების მიმართ Motion Paths-ის ეფექტები.



- რა მიზნით გამოიყენება ელექტრონული ცხრილები?
- როგორ ჩაინერება ფორმულა ელექტრონულ ცხრილში?

ელექტრონული ცხრილების მართვისათვის განკუთვნილ პროგრამებს (ცხრილურ პროცესორებს) სხვადასხვა ტიპის მონაცემებზე – სიმბოლოებით წარმოდგენილზე (ტექსტებზე), რიცხობრივზე, ლოგიკურზე, დროისა და თარიღის აღმნიშვნელზე და ა.შ. – მუშაობის შესაძლებლობა გააჩნია. უჯრაში ჩანერილი მონაცემის ტიპზეა დამოკიდებული, კერძოდ რა ოპერაციის ჩატარება შეიძლება მასზე. ასე მაგალითად, თუ უჯრაში შევიტანთ რიცხვ 43248-ს და უჯრის ფორმატად ტექსტს მივუთითებთ, უჯრის შინაარსი აღიქმება, როგორც სიმბოლოების (4,3,2,4,8) თანამიმდევრობა. თუ უჯრის რიცხობრივი ფორმატი იქნება მითითებული, მაშინ ელექტრონული ცხრილი ამ ჩანანერს რიცხვად აღიქვამს. უჯრის მიმართ თარიღის ფორმატის გამოყენების შემთხვევაში უჯრაში ჩანერილი რიცხვი აღიქმება, როგორც 2018 წლის 28 მაისი (დღეების რაოდენობა 01.01.1900-დან 28.05.2018-მდე 43248-ს შეადგენს).

საქმიანობა

1

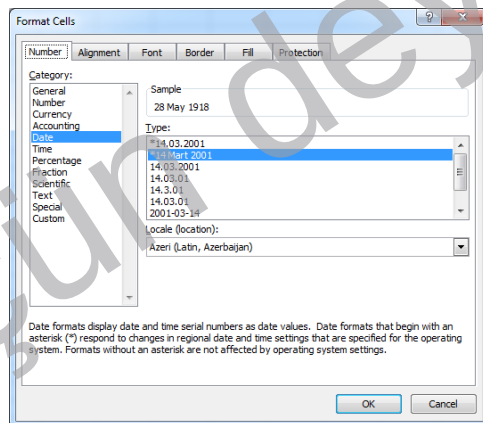
1. გახსენით ელექტრონული ცხრილი Microsoft Excel;
2. შეიტანეთ ნებისმიერ უჯრაში რომელიმე რიცხვი, მაგალითად, 6723.
დააჭირეთ კლავიშ <Enter>-ს;
3. კვლავ შევსებულ უჯრას დაუბრუნდით. ინსტრუმენტების პანელზე სია
General -ის განყოფილება Number-იდან აირჩიეთ პუნქტი Date და
დააკვირდით, რა ცვლილება მოხდა უჯრაში;
4. დაანკაჟუნეთ ინსტრუმენტების

პანელის ლილავ -ზე (Format) და მენიუში აირჩიეთ პუნქტი Format Cells. გაიხსნება შესაბამისი დიალოგური ფანჯარა;

5. გადადით ამ ფანჯრის ჩანართ Number-ში. ყურადღება მიაქციეთ განყოფილება Category-ში მითითებულ მონაცემებს;
6. სიის ელემენტების არჩევით, გაეცანით მონაცემების წარმოდგენის ნიმუშებს.

ვიმსჯელოთ:

- რამდენი ტიპი გააჩნია ფორმატ Date-ს (თარიღის) მონაცემებს?
- რა რიცხვი უნდა შევიტანოთ უჯრაში, რომ პროცენტული (Percentage) ფორმატის მითითებით ფორმატის შეცვლის შემთხვევაში უჯრის შიგთავსი 45%-ს აჩვენებდეს?



ელექტრონული ცხრილი, პირველ რიგში, მონაცემებზე გამოთვლების ჩატარების ავტომატიზაციისთვის არის განკუთვნილი. ამისათვის ცხრილის უჯრებში **ფორმულები** შეაქვთ. ფორმულაში შეიძლება შედიოდეს რიცხობრივი მონაცემები, ოპერაციების ნიშნები, სხვადასხვა ფუნქცია, ასევე ცხრილის ობიექტების მისამართები. უჯრების მისამართების შემცველი ფორმულები შეიძლება შევადაროთ მათემატიკურ განტოლებებს: განტოლებებში უჯრების მისამართების ადგილას ცვლადები იწერება.

ფორმულის ჩანაწერში უჯრის მისამართის ან უჯრების დიაპაზონის მისათითებლად **ბმულებს** იყენებენ. ბმულების მეშვეობით ელექტრონული ცხრილის ნებისმიერი უჯრების ერთმანეთთან დაკავშირება და ცხრილის მონაცემების სათანადო დამუშავება შეიძლება.

B1			f_x	=A1*A1+23
	A	B		C
1	12	=A1*A1+23		
2	4			
3	43			

ხშირად ერთი და იმავე ტიპის ამოცანების ამოსახსნელად ერთნაირ ფორმულებს იყენებენ. ერთი და იმავე ფორმულის სხვადასხვა უჯრაში ცალცალკე შეყვანა რომ არ დასჭირდეთ, მის ასლს ქმნიან. არსებობს უჯრის შიგთავსის ერთი და იმავე სტრიქონის ან სვეტის უჯრებში კოპირების მოსახერხებელი მეთოდი. ამისათვის, პირველ რიგში, უნდა გავაქტიუროთ ის უჯრა, რომლის შიგთავსის კოპირებაც გვჭირდება (უჯრის ირგვლის მუქი შავი ჩარჩო გაჩნდება). შემდეგ თავისი მაჩვენებელი ჩარჩოს მარჯვენა ქვედა კუთხეში არსებულ პატარა კვადრატთან (მაჩვენებელი მიიღებს + სახეს) უნდა დავაყოვნოთ და, თავის მარცხენა ღილაკიდან თითის აულებლად, მაჩვენებელი სტრიქონის ან სვეტის გასწვრივ გადავადგილოთ, მერე კი მოვაშოთ თითი თავის ღილაკს. შედეგად აქტიური უჯრის ფორმულა ყველა მონიშნულ უჯრაში დაკოპირდება. ანალოგიური მეთოდით ნებისმიერი უჯრის შიგთავსი სწრაფად და მარტივად დაკოპირდება სხვა უჯრებში.

ელექტრონულ ცხრილში ფორმულის ერთი უჯრიდან მეორეში კოპირებისას ფორმულებში გამოყენებული უჯრების მისამართები (ბმულები) ავტომატურად იცვლება. მათ **შეფარდებით ბმულებს** უწოდებენ. შეფარდებითი ბმული ჩვეულებრივი ფორმით იწერება, მაგ.: F3,E7.

ზოგჯერ ფორმულის ერთი ადგილიდან მეორეში კოპირებისას არ არის აუცილებელი ბმულის შეცვლა. ასეთ შემთხვევებში ელექტრონულ ცხრილებში **აბსოლუტურ ბმულებს** იყენებენ. აბსოლუტურ ბმულებში სვეტის აღმნიშვნელ ასოსა და სტრიქონის ნომრის წინ სიმბოლო \$ იწერება, მაგალითად: \$F3, \$E7.

თუ ფორმულის კოპირებისას ბმულის მხოლოდ ერთი რომელიმე ნაწილი – სვეტის აღმნიშვნელ ასო ან სტრიქონის ნომერი – იცვლება, მაშინ შერეულ ბმულს იყენებენ. ამ დროს სიმბოლო \$ ბმულის იმ ნაწილის წინ იწერება, რომელიც უცვლელი უნდა დარჩეს, მაგალითად, \$F3, E\$7.

დამატებითი

ფორმულა
ბმული
შეფარდებითი ბმული
აბსოლუტური ბმული
შერეული ბმული

შეფარდებითი და აბსოლუტური ბმულების გამოყენება ელექტრონულ ცხრილში

1. გახსენით ელექტრონული ცხრილი. გაიხსნება ცარიელი დოკუმენტი (სამუშაო წიგნი);
2. A1 უჯრაში შეიტანეთ ტექსტი: **1 აშშ დოლარი = ... მანათს;**
3. A2 უჯრაში ჩანერეთ აშშ დოლარის მიმდინარე კურსი (მაგალითად, 1,5);
4. დანარჩენი უჯრები ქვემოთ მოცემული ნიმუშის მხედვით შეავსეთ;

	A	B	C	D	E	F	G
1	1 აშშ დოლარი = ... მანათს						
2		1,5					
3		№	საქონლის დასახელება	ფასი (USD)	რაოდენობა	ლირებული (USD)	ლირებული (AZN)
4		1	სისტემური ბლოკი	200	2		
5		2	მონიტორი	250	2		
6		3	კლავიატურა	30	2		
7		4	თაგვი	15	2		
8		5	პრინტერი	90	1		
9		6	სკანერი	32	1		
10			ჯამი				

5. F4 უჯრაში შეიტანეთ ფორმულა $=D4*E4$ და დააჭირეთ კლავიშ **<Enter>**-ს. უჯრაში გაჩნდება ფორმულის მიხედვით შესრულებული გამოთვლის შედეგი – რიცხვი 400, კურსორი კი მეზობელ უჯრაში გადაინაცვლებს;
6. ისევ გადადით F4 უჯრაში. ის მეუქი ჩარჩოთი (**400**). შემოისაზღვრება. პროგრამა Excel-ში ასე აქტიური უჯრები მონიშნება. დააყოვნეთ თავგის მაჩვენებელი ჩარჩოს მარჯვენა ქვედა კუთხეში არსებულ პატარა კვადრატთან. მაჩვენებელი + სახეს მიიღებს. დააჭირეთ თავგის მარცხენა ღილაკს და თითის აულებლად გადაატარეთ თავგი მე-9 სტრიქონამდე;
7. F4 უჯრის ფორმულა დაკოპირდება F5:F9 უჯრების დიაპაზონში. მონიშნული დიაპაზონის ყველა უჯრაში აისახება ამ ფორმულის მიხედვით შესრულებული გამოთვლის შედეგი. ამ უჯრებში გადასვლისას დააკვირდით, როგორ იცვლება ბმულები ფორმულებში;
8. ჩანერეთ G4 უჯრაში ფორმულა $=F4*\$A\2 . უჯრაში გაჩნდება სისტემური ბლოკის ღირებულება მანათებში. გამოიყენეთ მე-6 ნაბიჯის ინსტრუქცია და დააკოპირეთ ეს ფორმულა G5:G9 უჯრების დიაპაზონში;
9. იმისათვის, რომ აშშ დოლარებში გამოითვალოს მოწყობილობათა ღირებულება, მონიშნეთ უჯრების დიაპაზონი F4:F10. დაანკაპუნეთ ინსტრუმენტების პანელის ღილაკ **Σ AutoSum** -ზე ან მენიუ **Formulas**-ის შესაბამის ღილაკზე. გამოითვლება ყველა საქონლის საბოლოო ღირებულება, რომელიც F10 უჯრაში აისახება;

	A	B	C	D	E	F	G
1	1 აშშ დოლარი = ... მანათს						
2		1,5					
3		№	საქონლის დასახელება	ფასი (USD)	რაოდენობა	ლირებული (USD)	ლირებული (AZN)
4		1	სისტემური ბლოკი	200	2	400	600
5		2	მონიტორი	250	2	500	750
6		3	კლავიატურა	30	2	60	90
7		4	თაგვი	15	2	30	45
8		5	პრინტერი	90	1	90	135
9		6	სკანერი	32	1	32	48
10			ჯამი			1112	1668

10. ანალოგიურად გამოითვლება საქონლის ღირებულება მანათებში;
11. შეცვალეთ A2 უჯრაში აშშ დოლარის კურსი მანათთან მიმართებით და დააკვირდით ცვლილებებს უჯრებში;
12. დაარეკით ფაილს სათანადო სახელი და შეინახეთ იგი.

ზოგჯერ გამოთვლების წარმოებისას სხვადასხვა სტანდარტულ ფუნქციას (მათემატიკურს, ლოგიკურს, ფინანსურს, სტატისტიკურს და სხვ.) იყენებენ. ამ ფუნქციებს გაეცნობით დიალოგურ ფანჯარაში, თუ ფორმულების პანელის  ლილაკზე დააწკაპუნებთ.

შევისწავლოთ დამოუკიდებლად

გაეცანით მათემატიკური ფუნქციების $|x|$ (ABS), $\sqrt{x}$ (SQRT), $\sin x$ (SIN), $\cos x$ (COS) ჩანაწერებს პროგრამა Excel-ში. გამოყავით ელექტრონული ცხრილის A სვეტი (x) არგუმენტისთვის, სვეტები B-E კი – შესაბამისი ფუნქციებისთვის. გამოიანგარიშეთ ფუნქციების მნიშვნელობა არგუმენტის 10 სხვადასხვა მნიშვნელობისთვის.

შეამოწმე შენი ცოდნა

1. მონაცემთა რა ტიპები გამოიყენება ელექტრონულ ცხრილებში?
2. რას ნიშნავს აბსოლუტური ბმული ფორმულაში და როგორ აღინიშნება იგი?
3. შეადგინეთ ცხრილი მოცემული ნიმუშის მიხედვით. ამისათვის:
 - ა) უჯრა A1-ში მიუთითეთ მიმდინარე წელი;
 - ბ) ჩაწერეთ სათანადო უჯრებში თქვენი მეგობრების სახელები, გვარები, დაბადების თარიღები;
 - გ) პირველი მათგანის ასაკის გამოსათვლელად უჯრა D4-ში ჩაწერეთ შესაბამისი ფორმულა. დააკოპირეთ ეს ფორმულა D სვეტის დანარჩენ უჯრებში.

A	B	C	D
2017			
გვარი	სახელი	დაბადების წელი	ასაკი
აბასლი	ელქანი	2002	15
ალიზადე	გონჩაი	2001	
იაგუბლუ	რუსტამი	2003	

4. უჯრა A2-ში მიუთითეთ ავტომობილის სიჩქარე, უჯრა B2-ში – დრო, რომელსაც ის გზის გავლას ანდომებს. გამოთვალეთ უჯრა C2-ში ავტომობილის მიერ გავლილი გზა ($s=vt$). ცვალეთ სიჩქარის (უჯრა A2-ის შიგთავსის) და დროის (უჯრა B2-ის შიგთავსის) მნიშვნელობა და დააკვირდით, როგორ იცვლება გავლილი გზა (უჯრა C2-ის შიგთავსი);
5. მოცემულია ელექტრონული ცხრილის ერთი ფრაგმენტი. როგორი იქნება C2 და C3 უჯრების მნიშვნელობა C1 უჯრიდან მათში ფორმულის დაკოპირების შემდეგ?

	A	B	C
1	45	110	=A\$1+B1
2	50	23	
3	12	70	



- როგორ აგებენ დიაგრამას ტექსტურ რედაქტორში?
- დიაგრამების რა სახეებს იცნობთ?

საქმიანობა

1

1. გახსენით ელექტრონული ცხრილი Microsoft Excel;
2. ააგეთ აზერბაიჯანის რესპუბლიკის მოსახლეობის რიცხოვნობის ამსახველი ცხრილი. ცხრილის სათაური ჩანერეთ A1 უჯრაში, ხოლო თავად ცხრილი განათავსეთ A3: J5 უჯრების დიაპაზონში.

მოსახლეობის რაოდენობის განაწილება ასაკის მიხედვით (2015 წლის დასაწყისისათვის)

	0-9	10-19	20-29	30-39	40-49	50-59	60-69	70 და ზევით	სულ
მოსახლეობის რაოდენობა ათასი ადამიანი	1489,1	1427,5	1866,5	1465,1	1305,9	1170,7	473,1	395,1	9593,0
მოსახლეობის რაოდენობა პროცენტებში	15,6	14,9	19,4	15,3	13,6	12,2	4,9	4,1	100

3. დაარქვით ფაილს სათანადო სახელი (მაგალითად, აზერბაიჯანის მოსახლეობა) და შეინახეთ იგი.

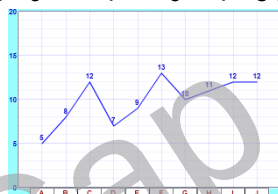
ვიმსჯელოთ:

- რამდენია 50 წლამდე ასაკის მოსახლეობის რაოდენობა?
- რომელი დიაგრამის – სვეტოვანის თუ წრიულის – მეშვეობით არის უფრო მოსახერხებელი ამ ცხრილის წარმოდგენა?

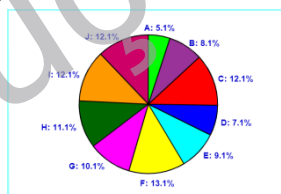
ელექტრონული ცხრილები ძირითადად დიდი მოცულობის ერთგვაროვანი რიცხოვრივი მონაცემების სწრაფად და ხარისხიანად დამუშავებისათვის არის განკუთვნილი, მაგრამ მონაცემთა გრაფიკული ფორმით წარმოდგენა უფრო თვალსაჩინო და ადვილად გასაგებია. ცხრილის მონაცემების გრაფიკული ფორმით წარმოდგენის ძირითადი საშუალებაა დიაგრამა.

დიაგრამა გათვალისწინებულია გრაფიკული ფორმით მონაცემთა თვალსაჩინოდ წარმოსადგენად. იგი საშუალებას გვაძლევს, უკეთ გავაანალიზოთ მონაცემები, შევადაროთ ისინი ერთმანეთს და გამოვავლინოთ ის კანონზომიერებები, რომლებიც ცხრილში არ ჩანს.

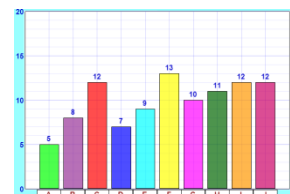
ელექტრონულ ცხრილებში შეიძლება აიგოს ხაზოვანი, წრიული, სვეტოვანი და სხვა დიაგრამები.



ხაზოვანი დიაგრამა



წრიული დიაგრამა



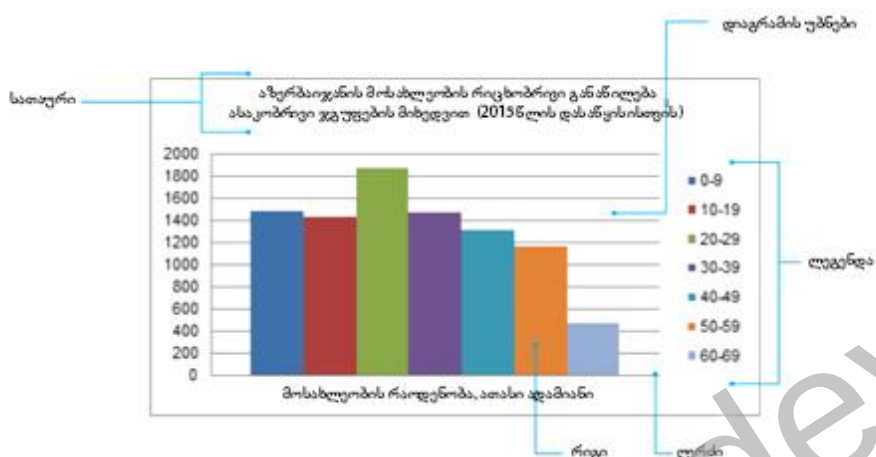
პისტოგრამა

ხაზოვანი დიაგრამა, ანუ **გრაფიკი** – დიაგრამის ტიპი, რომელშიც სიდიდეთა მნიშვნელობები ერთმანეთთან ხაზებითაა შეერთებული, რაც საშუალებას გვაძლევს, დავაკვირდეთ სიდიდეთა ცვლილებას ერთი წერტილიდან მეორეში გადასვლისას.

წრიული დიაგრამა – დიაგრამის ტიპი, რომელიც სექტორებად დაყოფილ წრეს წარმოადგენს. თითოეული სიდიდის შეფარდებითი მნიშვნელობა გამოისახება სექტორით, რომლის ფართობი მნიშვნელობათა საერთო ჯამში ამ მნიშვნელობის ხვედრით წილს შეესაბამება.

ჰისტოგრამა, ანუ **სვეტოვანი დიაგრამა** – დიაგრამის ტიპი, რომელიც სხვადასხვა სიმაღლის (სიგრძის) ვერტიკალური ან ჰორიზონტალური სვეტებისგან შედგება. ამ სვეტების სიმაღლე სიდიდეთა გარკვეულ მნიშვნელობებს შეესაბამება, რაც მათი შედარების საშუალებას იძლევა.

ელექტრონული ცხრილის ერთ სვეტში ან სტრიქონში განლაგებულ მონაცემებს მონაცემთა **რიგს (მწკრივს)** უწოდებენ. დიაგრამის აგებისას აუცილებელია ჯერ მონაცემთა რიგის გამოყოფა, შემდეგ კი დიაგრამის ტიპის არჩევა. დიაგრამა არის რთული ობიექტი, რომლის შემადგენელი ელემენტებია: **რიგი, ღერძი, სათაური, ლეგენდა, დიაგრამის აგების უბნები**.

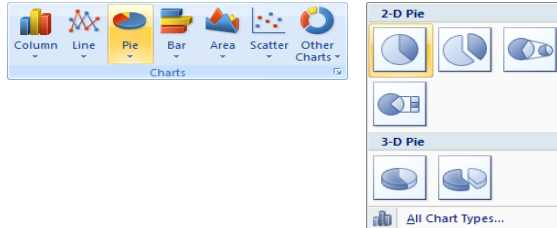


დიაგრამა შეიძლება აიგოს ელექტრონული ცხრილის როგორც ერთი, ისე რამდენიმე რიგის მიხედვით. უჯრების მონიშნული დიაპაზონისთვის დიაგრამის აგება მონაცემთა რამდენიმე რიგის მიხედვით წარმოებს. ამ შემთხვევაში თითოეულ რიგად მონიშნული დიაპაზონის შესაბამის სტრიქონს ან სვეტს იღებენ. სათაურად მომხმარებლის მიერ მონიშნული ტექსტი ითვლება, რომელსაც, როგორც წესი, დიაგრამის ზედა ნაწილში ათავსებენ. დიაგრამის გამოტანისას შეიძლება დაემატოს ლეგენდა – რიგების დასახელების სია (ცვლადების აღნიშვნები). აგების ველები, რომლებიც ღერძებითაა შემოსაზღვრული, მონაცემთა რიგების განსათავსებლადაა განკუთვნილი. შედეგების ანალიზის გასაადვილებლად აგების უბანზე ბადის დატანა შეიძლება.

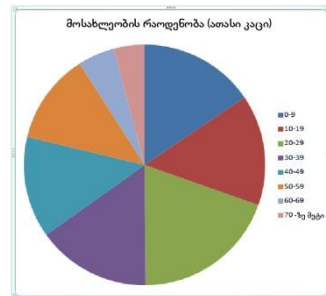
დაიმახსოვრე

დიაგრამა
ხაზოვანი დიაგრამა
წრიული დიაგრამა
ჰისტოგრამა
ლეგენდა

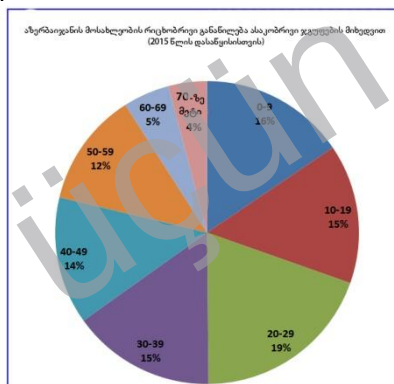
1. გახსენით ელექტრონული ცხრილის ფაილი „აზერბაიჯანის მოსახლეობა“;
2. ცხრილის საფუძველზე დიაგრამის ასაგებად მონიშნეთ უჯრების დიაპაზონი A3:14.
3. მენიუ Insert-ის ბლოკ Chart-ში დაანკაპუნეთ ლილაკ Pie-ზე. შედეგად პროგრამა შემოგთავაზებთ წრიული დიაგრამების სხვადასხვა ტიპს;



4. აირჩიეთ პირველი ვარიანტი. სამუშაო ველში პატარა ჩარჩოში აიგება წრიული დიაგრამა. თუ დიაგრამის მარჯვნივ მოთავსებული ლეგენდა სრულად არ ჩანს, შეგიძლიათ გაჭიმოთ ჩარჩო მანამდე, სანამ ლეგენდა მთლიანად არ გამოისახება;
5. პარალელურად ინსტრუმენტების პანელზე დიაგრამის მაკეტისა და სტილის შესაძლო ვარიანტები გამოჩნდება; აირჩიეთ პირველი მაკეტი (Layout 1) და მეათე სტილი (Style 10);



6. დიაგრამის ყოველ სექტორში მითითებული იქნება ასაკობრივი ჯგუფი და მისი ხვედრითი წილი (პროცენტებში) მოსახლეობის საერთო რიცხვში. დაანკაპუნეთ სათაურზე: „მოსახლეობის რიცხოვნობა (ათასი ადამიანი)“ და ამ ველში შეიტანეთ ტექსტი: „აზერბაიჯანის მოსახლეობის რიცხოვრივი განაწილება ასაკობრივი ჯგუფების მიხედვით (2015 წლის დასაწყისისთვის)“.



7. დაარქვით ფაილს შესაბამისი სახელი და შეინახეთ იგი.

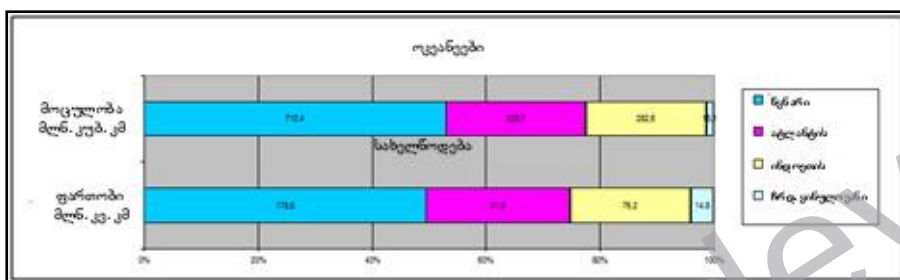
შევისწავლოთ დამოუკიდებლად

შეადარეთ ბოლო დიაგრამის პროცენტული მნიშვნელობები იმ მნიშვნელობებს, რომლებიც გაკვეთილის დასაწყისში მოცემული ცხრილის პირველ სტრიქონში (მოსახლეობის რაოდენობა, ათასი ადამიანი) არის წარმოდგენილი.

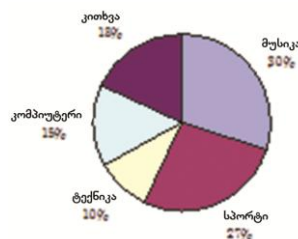
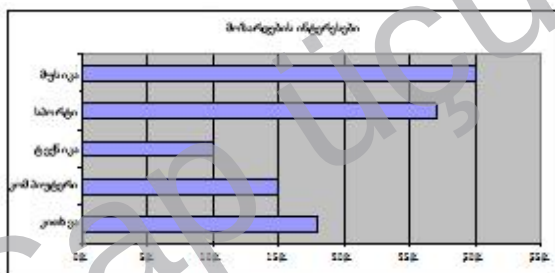
შეამოწმა შენი ცოდნა

- რა ობიექტებისგან შედგება დიაგრამა?
- დიაგრამების რომელი ძირითადი ტიპები არსებობს ელექტრონულ ცხრილებში?
- შეადგინეთ შემდეგი ცხრილი და მისი შესაბამისი დიაგრამა;

	A	B	C
1	ოკეანეები		
2	საბუნალოდება	ფართობი მლნ. კვ. კმ	მოცულობა მლნ. კუბ. კმ
3	წყარი	178,6	710,4
4	ატლანტის	91,6	329,7
5	ინდოეთის	76,2	282,6
6	ჩრდ. ყინულოვანი	14,8	18,1



- გადახედეთ დიაგრამებს და უპასუხეთ შეკითხვებს:
 - რა ინფორმაციაა ასახული დიაგრამებზე? როგორ გაიგეთ ეს?
 - როგორ შეიძლება ამ ინფორმაციის ცხრილის სახით წარმოდგენა?
 - ყველაზე მეტად რა აინტერესებთ მოზარდებს?



შემაჯავებელი კითხვები და დავალებები

1. როგორ გესმით ცნება „ტექსტური დოკუმენტის ობიექტები“?
2. როგორ განვითავსოთ ნახატი ტექსტის მიმართ სხვადასხვა ადგილას?
3. ახსენით მოცემული ლილაკების დანიშნულება.



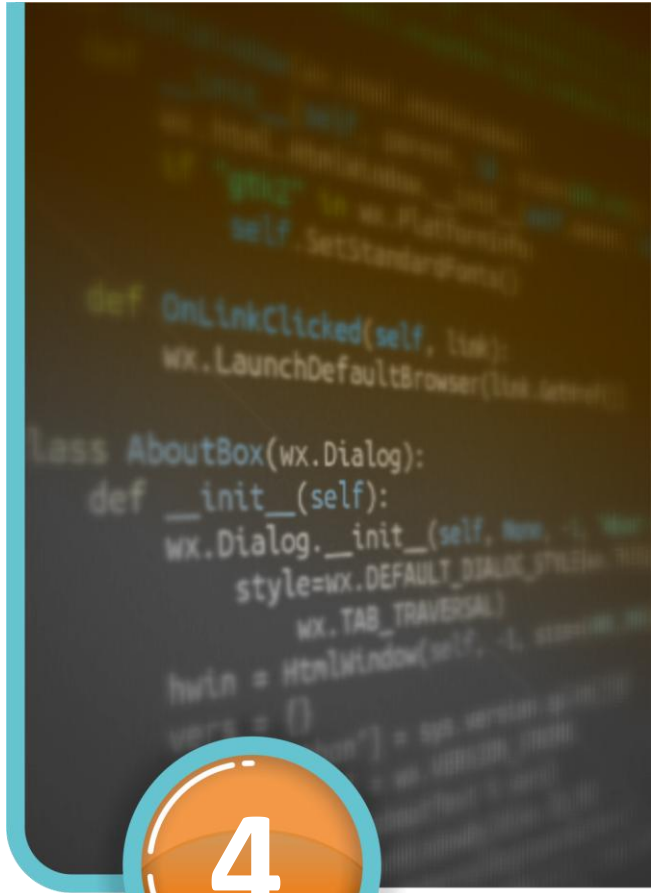
4. როგორ მივუთითოთ მთელ მულტიმედიურ პრეზენტაციაში სლაიდიდან სლაიდზე გადასვლის ერთნაირი ეფექტი?
5. რა განსხვავებაა შეფარდებით, აბსოლუტურ და შერეულ მისამართებს შორის?
6. რა უნდა ჩაწეროთ ელექტრონული ცხრილის E3 უჯრაში ოთახის მოცულობის გამოსათვლელად?

	A	B	C	D	E
1					ოთახის მოცულობა
2	სიგრძე	7	მეტრი		V
3	სიგანე	3,5	მეტრი		
4	სიმაღლე	3,5	მეტრი		

7. მოცემულია ელექტრონული ცხრილის ფრაგმენტი. უჯრა C1-ში ჩანერილია ფორმულა =A1-\$B\$1. განსაზღვრეთ მონაცემთა მნიშვნელობა C2 და C3 უჯრებში მათში C1-უჯრიდან ფორმულის დაკოპირების შემდეგ.

	A	B	C
1	117	29	
2	98	-12	
3	45	23	

8. დიაგრამების რომელი ტიპის გამოყენება მიგაჩნიათ მიზანშეწონილად წლის თითოეულ თვეში მზიანი დღეების რაოდენობის საჩვენებლად?



4

დაპროგრამება

14. რიცხვებზე მუშაობა
15. სტრიქონები
16. სიები
17. ფუნქციები
18. დაპროგრამების ენების კლასიფიკაცია

14

რიცხვებზე მუშაობა



- რა არის პროგრამა?
- დაპროგრამების ენა Python-ის რომელ ოპერატორებს იცნობთ?

საქმიანობა

1

გაეცანით პროგრამის მოცემულ ფრაგმენტს Python-ის ენაზე და განსაზღვრეთ, რას აკეთებს იგი. კომპიუტერში შესრულების გარეშე დაადგინეთ, რა მნიშვნელობას მიიღებს `i` ცვლადი პროგრამის დასასრულს.

```
i = 1
while i <= 10:
    print(i*i)
    i = i + 1
```

ვიმსჯელოთ:

- რატომ წერია ოპერატორი `print` და ოპერატორი `while` სხვადასხვა დონეზე?
- როგორი იქნებოდა `i` ცვლადის საბოლოო მნიშვნელობა, სტრიქონი `i=i + 1` ოპერატორ `while`-ის დონეზე რომ ეწეროს?

Python-ის დაპროგრამების ენაზე პროგრამული კოდი ჩვეულებრივი ტექსტური ფაილია. ეს ფაილი ასოების, ციფრების, შუალედების (ინტერვალების), სასვენი ნიშნებისა და სხვა სიმბოლოებისგან შედგება. ტექსტი სტრიქონებად იყოფა. Python-ის ყველა ოპერატორი რამდენიმე სტრიქონზე შეიძლება ჩაინეროს. ოპერატორის რამდენიმე სტრიქონზე დასაწერად სიმბოლო `"\n"`-ს იყენებენ.

რამდენიმე ხნის წინ დაწერილი პროგრამის ტექსტს მოგვიანებით რომ დავუბრუნდეთ, გაგვიჭირდება იმის გახსენება, თუ რას აკეთებს ამ პროგრამის ესა თუ ის ნაწილი. იმისათვის, რომ პროგრამის დეტალები გვახსოვდეს და იგი სხვებისთვისაც გასაგები იყოს, მიზანშეწონილია მის განსაზღვრულ ადგილებში **კომენტარების** განთავსება. კომენტარი პროგრამის წამკითხველისთვის განკუთვნილი განმარტებაა. კომენტარებში შეიძლება აღინიშნოს, რა მიზნით შეიქმნა პროგრამა, ვინ არის მისი შემქმნელი, როდის შეიტანეს მასში ცვლილებები, რა დანიშნულება აქვს პროგრამაში არსებულ ფუნქციებს.

Python-ის ენაზე კომენტარები სიმბოლო `"#"`-ის შემდეგ იწერება. პროგრამის მანქანურ კოდზე გადაყვანისას კომენტარებს არ ითვალის-

ნინებენ. ამიტომ, როდესაც პროგრამის რომელიმე ნაწილის „გაყინვის“ აუცილებლობა ჩნდება, მას კომენტარის სახით აღნიშნავენ.

ღამასმოვრა

კომენტარი
აბზაცი

არითმეტიკული ოპერაცია
შედეგების ნიშნები

Python-ის ენაზე დანერგილი პროგრამის ლოგიკურად შესრულებისათვის დიდი მნიშვნელობა აქვს მის სტრუქტურასა და

აბზაცებს სტრიქონის დასაწყისში. აბზაცი ოპერატორების ჯგუფებად გაერთიანებისთვის გამოიყენება. ოპერატორების ჯგუფი ზოგჯერ სხვა ჯგუფის შემადგენლობაში შედის და რაც უფრო ღრმად შედის მასში, მით უფრო დიდია აბზაცი. გაითვალისწინეთ, რომ ოპერატორების ერთი ჯგუფის ფარგლებში აბზაცი ერთნაირი უნდა იყოს.

თქვენ უკვე იცნობთ რიცხვით სიდიდეებსა და არითმეტიკულ ოპერაციებს Python-ის ენაზე. კიდევ ერთხელ შეგახსენებთ მათ.

ოპერაცია	აღნიშვნა
$x + y$	შეკრება (x და y რიცხვების ჯამი)
$x - y$	გამოკლება (x და y რიცხვების სხვაობა)
$x * y$	გამრავლება (x და y რიცხვების ნამრავლი)
x / y	გაყოფა (x და y რიცხვების განაყოფი, ნილადი)
$x // y$	ნაშთიანი გაყოფა (განაყოფი – მთელი რიცხვი)
$x \% y$	ნაშთი, რომელიც x და y რიცხვების გაყოფის შედეგად მიიღება
$x ** y$	ახარისხება (x , ხარისხად y)
$-x$	რიცხვის ნიშნის შეცვლა

პირობის ჩასაწერად პროგრამაში **შედეგების ნიშნებს** იყენებენ. ცხრილში ნაჩვენებია ამ ნიშნების ჩანაწერი ALPLogo-სა და Python-ის ენებზე. ყურადღება მიაქციეთ ისეთი ნიშნების ჩანაწერს, როგორებიცაა „უდრის“ და „არ უდრის“, ვინაიდან მათი არასწორად გამოყენება პროგრამულ შეცდომებს იწვევს.

ALPLogo	Python	აღნიშვნა
=	==	უდრის
<>	!=	არ უდრის
>	>	მეტია
<	<	ნაკლებია
>=	>=	მეტია, ანდა ტოლი
<=	<=	ნაკლებია, ანდა ტოლი

რიცხვის ციფრებად დაშლა. როგორც იცით, ნებისმიერი რიცხვის 10-ზე გაყოფისას მიღებული განაყოფი განსაზღვრავს ამ რიცხვის ერთეულის თანრიგის ციფრს, მაგალითად, თუ n რიცხვის ერთეულის თანრიგის ციფრი a -ს ტოლია, მაშინ $a = n \% 10$

გასაგებია, რომ თუ ბოლო ციფრს მოვაშორებთ, მაშინ მიღებული ახალი რიცხვის ბოლო ციფრი სანყისი რიცხვის ათეულების თანრიგის ციფრი იქნება.

ამ ოპერაციის მეშვეობით, ახალი რიცხვის ბოლო ციფრის მოშორებით საწყისი რიცხვის ათეულის ციფრს ვიღებთ. ამ გზით ნებისმიერი რიცხვის ციფრების განსაზღვრა შეიძლება. რიცხვის ბოლო ციფრის (ბოლო თანრიგის) მოსაშორებლად, ის უნდა გავყოთ 10-ზე და ავიღოთ მისი მთელი ნაწილი, ანუ მთელი რიცხვების გაყოფით (/) ვისარ-გებლოთ, მაგალითად, n რიცხვის ბოლო ციფრის მოშორებით მიიღება m რიცხვი, ანუ $m = n // 10$

საქმიანობა

2

აკრიფეთ პროგრამა Python-ის გარემოში და შეასრულეთ იგი.

```
n = input('შეიყვანეთ ორნიშნა მთელი რიცხვი:')
n = int(n)
if n > 9 and n < 100:
    a = n % 10          # ერთეულების თანრიგის ციფრი
    b = n // 10         # ათეულების თანრიგის ციფრი
    s = a + b
    p = a * b
print('ციფრების ჯამი =', s)
print('ციფრების ნამრავლი =', p)
else:
    print('რიცხვი არასწორადაა შეყვანილი')
```

ვიმსჯელოთ:

- რას აკეთებს პროგრამა?
- სხვადასხვა რიცხვების შეყვანით შეამოწმეთ, სწორად მუშაობს კომპიუტერი თუ არა.
- ჩაწერეთ `print`-ის ოპერატორები იმავე დონეზე, სადაც ოპერატორი `if` წერია და ყურადღება მიაქციეთ, როგორ შეიცვლება პროგრამის შესრულების შედეგი.

მაგალითი

რიცხვის ციფრების ჯამისა და ნამრავლის პოვნა. მოცემულია მთელი დადებითი რიცხვი. უნდა ვიპოვოთ მისი ციფრების ჯამი და ნამრავლი.

ამოხსნა. აღვნიშნოთ პირობითად მოცემული რიცხვი n ცვლადით, მისი ციფრების ჯამი – s ცვლადით, ციფრთა ნამრავლი – p ცვლადით. მაშინ ამოცანის ალგორითმი შეგვიძლია ჩავწეროთ ასე:

ალგორითმი:

1. მივანიჭოთ s ცვლადს საწყისი მნიშვნელობა 0;
2. მივანიჭოთ p ცვლადს საწყისი მნიშვნელობა 1;
3. სანამ n ნულზე მეტია,

ვიპოვოთ ნაშთი, რომელიც მიიღება n -ის 10-ზე გაყოფით (რიცხვის ბოლო ციფრი), მივუმატოთ ის ჯამს და გავადიდოთ ნამრავლი.

n რიცხვის ბოლო ციფრის (თანრიგის) მოსაშორებლად გავყოთ ის 10-ზე ნაშთით.

```
n = input('შეიყვანეთ რიცხვი:')
n = int(n)
s = 0
p = 1
while n > 0:
    q = n % 10
    s = s + q
    p = p * q
    n = n // 10
print('ციფრების ჯამი =', s)
print('ციფრების ნამრავლი =', p)
```

შევისწავლოთ დამოუკიდებლად

გაეცანით პროგრამას და გაარკვიეთ, რისთვის არის ის განკუთვნილი. მეორე სტრიქონში მოცემული ფუნქცია `abs()` რიცხვის მოდულის პოვნას ემსახურება. როგორც მათემატიკიდან იცით, რიცხვის მოდული მისი მნიშვნელობაა, ნიშნის გარეშე, მაგალითად:

```
abs(4) = 4, abs(-4) = 4.
m = int(input(' შეიყვანეთ სამნიშნა მთელი რიცხვი: '))
n = abs(m)
a = (n // 100) % 10
b = (n // 10) % 10
c = n % 10
k = 100 * c + 10 * b + a

if m < 0:
    print(-k)
else:
    print(k)
```

ნებისმიერი დადებითი და უარყოფითი სამნიშნა რიცხვის შეყვანით შეამოწმეთ პროგრამის მუშაობის სისწორე.

შეამოწმა შენი ცოდნა ?

1. როგორ კეთდება კომენტარები Python-ის დაპროგრამების ენაზე?
2. მოცემულია ორნიშნა მთელი რიცხვი. გამოიტანეთ რიცხვის პირველი ციფრი მარცხნიდან (ათეული), ორი ინტერვალის შემდეგ კი – ციფრი მარჯვნიდან (ერთეული).
3. გამოიტანეთ მოცემული ნატურალური რიცხვის მარჯვნიდან მესამე ციფრი.
4. შეადგინეთ პროგრამა, რომელიც განსაზღვრავს მოცემული რიცხვის 3-ზე გაყოფადობას.
5. ჩაწერეთ ქვემოთ მოცემული ალგორითმი პროგრამის სახით Python-ის ენაზე.
 - შეიყვანეთ ნებისმიერი ორნიშნა რიცხვი;
 - გაამრავლეთ იგი თავის თავზე;
 - მიუმატეთ მიღებული რიცხვი შეყვანილი რიცხვის 4-ზე ნამრავლს;
 - მიღებულ შედეგს მიუმატეთ რიცხვი 3;
 - გაყავით მიღებული შედეგი რიცხვზე, რომელიც შეყვანილ რიცხვზე 3-ით მეტია;
 - გამოაკელით თქვენ მიერ შერჩეული რიცხვი მიღებულ განაყოფს;
 - გამოიტანეთ შედეგი ეკრანზე.



- რას ნიშნავს მონაცემთა სტრიქონული ტიპი პროგრამირებაში?
- როგორ განვასხვაოთ ციფრებისგან შედგენილი სტრიქონი რიცხვებისგან Python-ის ენაზე?

საქმიანობა

აკრიფეთ და შეასრულეთ პროგრამა Python-ის გარემოში.

```
s1 = 'ა'
s2 = 'ლ'
s3 = 'ჰ'
print (s1 + s2 + s3)
print ("ა", "ლ", "ჰ")
```

ვიმსჯელოთ:

- რა ტიპის სიდიდეებია გამოყენებული პროგრამაში?
- ერთნაირია თუ არა შედეგი ორივე ოპერატორ `print` -ში? თუ არა, რა უნდა გავაკეთოთ იმისათვის, რომ ერთნაირი შედეგი მივიღოთ?

ღაიმახსოვრე

სტრიქონი
ცარიელი სტრიქონი
ინდექსი
სტრიქონის სიგრძე
მეთოდი

Python-ის ენაში მონაცემთა ერთ-ერთი ყველაზე ხშირად გამოყენებული ტიპია სტრიქონი. **სტრიქონი** ნებისმიერი სიმბოლოების თანამიმდევრობაა. ის, ჩვეულებრივ, ერთმაგ ბრჭყალში ისმება ('), მაგრამ ზოგიერთი სტრიქონის გამოყოფა ჩვეულებრივი ბრჭყალებითაც შეიძლება, მაგალითად, 'ეს ნიმუშია', „ცოდნა ძალაა“. სტრიქონი შეიძლება შედგებოდეს

ასოების, ციფრებისა და სხვა სიმბოლოების, მათ შორის, შუალედებისგანაც. სტრიქონი შეიძლება მიენიჭოს ცვლადს, მაგალითად:

```
a = 'აზერბაიჯანის', b = 'რესპუბლიკა'
```

როგორც რიცხვებზე, ისე სტრიქონებზეც, გარკვეული მოქმედებების ჩატარება შეიძლება, მაგალითად, ორი სტრიქონის შეკრებისას წარმოიქმნება მათი „შენჯნებისგან“ მიღებული სტრიქონი.

```
>>> c = a + b
>>> print (c)
```

აზერბაიჯანის რესპუბლიკა

```
>>> c = a + 'დემოკრატიული' + b
>>> print (c)
```

აზერბაიჯანის დემოკრატიული რესპუბლიკა

ყოველ სიმბოლოს, სტრიქონში მისი ადგილმდებარეობის მიხედვით, თავისი რიგითი ნომერი გააჩნია. ამ ნომრით ადვილად შეიძლება სიმბოლოს პოვნა სტრიქონში, ან მისი ამოჭრა სტრიქონიდან. Python-ის ენაზე სიმბოლოთა ნუმერაცია 0-დან იწყება. მეორე სიმბოლოს ნომერი იქნება 1, მესამისა – 2 და ა.შ. პროგრამირებაში ნომერს **ინდექსი** ეწოდება. ინდექსით შესაძლებელია სტრიქონის კონკრეტული სიმბოლოს განსაზღვრა, მაგალითად:



გვიდო ვან
როსუმი
(1956)

Python-ის ენის
შემქმნელი

```
>>> a = 'აზერბაიჯანი'
>>> a[4]
```

ორი ინდექსით შეიძლება სტრიქონის ფრაგმენტის გამოყოფა, რომელსაც სტრიქონის კვეთს უწოდებენ.

```
>>> a[0:4]
'აზერ'
```

როგორც ხედავთ, მიუხედავად იმისა, რომ დიაპაზონის მეორე ინდექსად მითითებულია რიცხვი 4, ამ პოზიციაზე მდგომი სიმბოლო სტრიქონის კვეთში არ შედის. სხვაგვარად რომ ვთქვათ, მეორე ინდექსის პოზიციაზე მდგომი სიმბოლო არასდროს შედის სტრიქონის კვეთში. დიაპაზონის მონიშვნისას მისი ერთი ინდექსი შეიძლება არ მიუთითონ. თუ კვეთში ერთი ინდექსი გამოტოვებულია, Python ავტომატურად ამოჭრის სტრიქონს ნულოვანი სიმბოლოდან. თუ მეორე ინდექსია გამოტოვებული, მაშინ კვეთში სტრიქონის დასასრულამდე ყველა სიმბოლო ხვდება.

```
>>> a[:5]
'აზერბ'
>>> a[7:]
'ჯანი'
```

სტრიქონის სიგრძეში იგულისხმება სტრიქონში შემავალი ყველა სიმბოლოს რაოდენობა, შუალედების ჩათვლით. სპეციალური ფუნქციის, len()-ის დახმარებით, შესაძლებელია სტრიქონის სიგრძის განსაზღვრა.

```
>>> len(a)
11
```

როგორც იცით, პროგრამის მუშაობის დროს მასში გამოყენებული ყველა მონაცემი კომპიუტერის ოპერაციულ მეხსიერებაში ინახება. სხვა მონაცემობებში განთავსებულ მონაცემებს პროგრამა ისე აღიქვამს, როგორც **გარეს**. გარე მონაცემების მიღების პროცესს შეტანა ეწოდება, ხოლო მონაცემების გარე მონაცემობებზე გადატანას – გამოტანა. როგორც რიცხობრივი, ისე სტრიქონული სიდიდეების კლავიატურიდან შესატანად გამოიყენება ფუნქცია input()- ეს ფუნქცია შეტანილ სიდიდეს სტრიქონად აღიქვამს. მაგალითად, ბრძანება =input()-ის მეშვეობით კლავიატურიდან რიცხვ 123-ის შეტანისას ცვლად a-ს ენიჭება არა 123-ის, არამედ '123'-ის მნიშვნელობა. ამიტომ ზოგჯერ საჭირო ხდება სიდიდის ერთი ტიპის მეორით შეცვლა.

Python-ის ენას, ისევე როგორც დაპროგრამების სხვა ენებს, სიდიდეთა ერთი ტიპიდან მეორეში გადასაყვანად ფუნქციები გააჩნია,

მაგალითად, რიცხვითი სიდიდის სტრიქონულში გადასაყვანად ფუნქცია `str()`-ს იყენებენ, ხოლო სტრიქონული სიდიდის რიცხვითში გადასაყვანად – ფუნქცია `int()`-ს.

```
>>> int('123')    >>> str(123)
123               '123'
```

Python-ში მონაცემების დასაბეჭდად, ან სხვაგვარად რომ ვთქვათ, მათ გამოსატანად, ფუნქცია `print()` გამოიყენება. ამ ფუნქციის დახმარებით შესაძლებელია ეკრანზე არა მარტო ტექსტების, არამედ ცვლადების მნიშვნელობათა გამოტანაც.

```
>>> a = 14
>>> print('ორჰანი', a, 'წლისა')
ორჰანი 14 წლისა
```

ამ მაგალითში ფუნქცია `print()` ყველა მონაცემს საბეჭდად ერთ სტრიქონში უშვებს, მათი ერთმანეთისგან გამოსაყოფად კი შუალედს იყენებს. მნიშვნელობათა გამოსაყოფად სხვა საშუალებებიც არსებობს, მაგალითად, გამოტანილ მნიშვნელობათა ერთმანეთისგან გასაცალკევებლად სიმბოლოს სახით შეიძლება მძიმის ან დეფისის დასმა. გამაცალკევებლად გამოყენებული სიმბოლო პარამეტერ `sep` -ის მეშვეობით აღინიშნება (მაგალითად, `sep = '-'`). თუ მითითებულია `sep = '\n'`, მაშინ თითოეული მნიშვნელობა ახალი სტრიქონიდან დაიბეჭდება.

```
>>> a = 14
>>> print('ორჰანი', a, 'წლისა', sep='\n')
ორჰანი
14
წლისა.
```

შევისწავლოთ დამოუკიდებლად

```
s = input()
print(s[2])
```

...

პროგრამის ეს ფრაგმენტი ალგორითმის პირველ ორ ნაბიჯს შეესაბამება. მიაწერეთ ამ ალგორითმის ოპერატორები.

ალგორითმი

1. კლავიატურიდან სიმბოლოთა ნებისმიერი სტრიქონის შეტანა.
2. შეტანილი სტრიქონის მესამე სიმბოლოს გამოტანა.
3. სტრიქონის უკანასკნელი სიმბოლოს გამოტანა.
4. სტრიქონის პირველი ხუთი სიმბოლოს გამოტანა.
5. სტრიქონის ყველა სიმბოლოს გამოტანა, ბოლო ორის გარდა.
6. სტრიქონის წველინდექსიანი სიმბოლოების გამოტანა.
7. სტრიქონის კენტიინდექსიანი სიმბოლოების გამოტანა.
8. სტრიქონის სიგრძის გამოტანა.

როგორ იცით, ნებისმიერ ობიექტს აქვს განსაზღვრული მახასიათებლები. ინფორმაციაში არჩევენ ობიექტის ორი სახის მახასიათებლებს: **ობიექტების თვისებებს** და **ობიექტის მეთოდებს**.

დაპროგრამების ენაზე ნებისმიერი ობიექტის მეთოდის საჩვენებლად სპეციალურ ჩანაწერს იყენებენ: ჯერ წერენ ობიექტის სახელს, შემდეგ კი

– მეთოდს. მათ შორის წერტილი ინერება (მაგალითად, `s1.upper()`, `S.find('a', 5, 10)`). Python-ის ენაზე სტრიქონები ობიექტებად აღიქმება და, როგორც ყოველ ობიექტს, მათაც თავიანთი მეთოდები გააჩნია. ყველაზე ხშირად მეთოდი `find` გამოიყენება. ეს მეთოდი მითითებულ სტრიქონში ქვესტრიქონის მოსაძებნად გამოიყენება. თუ ქვესტრიქონი მოიძებნება, მეთოდი საძებნი ქვესტრიქონის პირველი ჩართულის პირველი სიმბოლოს ინდექსს დააბრუნებს. თუ ქვესტრიქონი არ მოიძებნება, მეთოდი დააბრუნებს მნიშვნელობა `-1`-ს, მაგალითად:

```
>>>s = 'არცოდნა სირცხვილი არ არის, სირცხვილი არსწავლა'.
>>>print(S.find('სწავლა'))
39
>>>print(S.find('ცო'))
3
>>>print(S.find('წიგნი'))
-1
```

მეთოდ `find`-ს სხვა პარამეტრებიც შეიძლება ჰქონდეს. თუ გამოვიძახებთ `find`-ს სამი პარამეტრით `S.find(T, a, b)`, მაშინ ძებნა ქვესტრიქონ `S[a:b]` -ში განხორციელდება. მეთოდი `S.find(T, a, b)` ქვესტრიქონ `T`-ს პირველი ჩართულის პირველი სიმბოლოს ინდექსს `S` სტრიქონში დააბრუნებს.

```
>>>print(S.find('o', 5, 10))
8
```

მაგალითი

შეადგინეთ პროგრამა, რომელშიც კლავიატურიდან თქვენი გვარი და სახელი შედის, ხოლო გამოტანით, პრიქით გამოდის: ჯერ სახელი, მერე – გვარი.

პროგრამაში გამოყენებული ცვლადების აღწერა:

`s` – შესაყვანი სტრიქონი;

`lastname, firstname` სტრიქონის პირველი და მეორე სიტყვა; `k` – სიტყვებს შორის შუალედის პოზიცია.

ამოცანის ამოხსნის ალგორითმი:

1. პროგრამა შეიტანს სტრიქონს;
2. მეთოდ `find`-ის დახმარებით განისაზღვრება სიტყვებს შორის შუალედის პოზიცია;
3. სტრიქონის ნაწილი, დასაწყისიდან შუალედამდე, ენიჭება ცვლად `lastname`-ს;
4. სტრიქონის ნაწილი, შუალედის შემდგომი პოზიციიდან ბოლომდე, ენიჭება ცვლად `firstname`-ს;
5. წარმოიქმნება ახალი სტრიქონი, რომელიც შედგება მეორე სიტყვის (`firstname`-ს), შუალედისა და პირველი სიტყვისაგან;
6. ახალ სტრიქონს პროგრამა ბეჭდვაზე გამოიტანს.

```
s = input()
```

```

p = s.find(' ')
lastname = s[:p]
firstname = s[p + 1:]
s = firstname + ' ' + lastname
print(s)

```

პირველი სიტყვა – გვარი
მეორე სიტყვა – სახელი

ობიექტ „სტრიქონს“ სხვა მეთოდებიც გააჩნია. ისინი ქვემოთ მოცემულ ცხრილშია მოყვანილი.

მეთოდი	აღწერა
<code>s1.count(s0)</code>	განსაზღვრავს ქვესტრიქონ s0-ის სტრიქონ s1-ში ჩართულობის სიხშირეს. <pre>>>> s1 = 'abracadabra' >>> s1.count('ab') 2</pre>
<code>s1.strip()</code>	შლის შუალედებს სტრიქონის დასაწყისსა და ბოლოში, მაგალითად: <pre>>>> s1 = ' abc ' >>> s1.strip() 'abc'</pre>
<code>s1.replace(s2,s3)</code>	სტრიქონის ფრაგმენტ s2-ს ჩანაცვლებს ფრაგმენტ s3-ით, მაგალითად: <pre>>>> s1 = ' ყირგიზეთი ' >>> s1.replace('ყირგიზ' , 'ყაზახ') 'ყაზახეთი'</pre>
<code>s1.capitalize()</code>	პირველ ასოს მთავრულ ასოდ გარდაქმნის, მაგალითად: <pre>>>> s1 = 'ყურადღებით!' >>> s1.capitalize() 'ყურადღებით!'</pre>
<code>s1.upper()</code>	სტრიქონის ყველა ასოს მთავრულ ასოებად გარდაქმნის, მაგალითად: <pre>>>> s1 = 'ჰუსარი გუბა ხაჩმაზი ' >>> s1.upper() 'ჰუსარი გუბა ხაჩმაზი'</pre>
<code>s1.lower()</code>	სტრიქონის ყველა ასოს არამთავრულ ასოებად გარდაქმნის, მაგალითად: <pre>>>> s1 = 'ცისფერი-მინამული- ყვითელი-შავი ' >>> s1.lower() 'ცისფერი-მინამული- ყვითელი-შავი '</pre>

შეამოწმე შენი ცოდნა

- რა არის სტრიქონი?
- რას გულისხმობს პროგრამაში შეტანისა და გამოტანის ოპერაციები? რა ფუნქციებითაა ისინი წარმოდგენილი Python-ში?
- რა არის მეთოდი?
- შეადგინეთ პროგრამა, რომელსაც სიტყვის შემადგენელი ასოები უკუღმა თანამიმდევრობით გამოაქვს.
- გამოიტანეთ ეკრანზე სიმბოლო, რომელიც სტრიქონის შუაში მდებარეობს. თუ სიმბოლოთა რაოდენობა კენტია, მაშინ ორი სიმბოლო გამოიტანეთ.

16 სივსი



- როგორ იქმნება მარკირებული და დანომრილი ცხრილები ტექსტურ რედაქტორში?
- სიის რომელ სახეს ეკუთვნის აქ მოყვანილი შეკითხვები?

საქმიანოზა

1

```
s = 0
for i in range(1, 101):
    s = s + i
print(s)
```

ვიმსჯელოთ:

- რა მიიღება მოცემული ფრაგმენტის შესრულების შედეგად?
- რა სახის ალგორითმული სტრუქტურაა გამოყენებული ამ ფრაგმენტში?
- რას აკეთებს ფუნქცია `range()` ?

მრავალრიცხოვანი მონაცემების ერთობლივად შენახვის მიზნით Python-ის ენა ფართოდ იყენებს სიებს. **სია** არის გარკვეული წესით შენახული ობიექტების ნაკრები. სიის შემადგენელ ობიექტებს მის **ელემენტებს** უწოდებენ. სიებში შეიძლება რიცხვების, სტრიქონების, სხვა სიების ან ყველაფრის ერთად შენახვა. სიის ყოველ ელემენტს შეესაბამება რიგითი ნომერი (**ინდექსი**), რომლითაც მას შეიძლება მივმართოთ. ნებისმიერ მომენტში შეიძლება სიის შეცვლა მისთვის ახალი ელემენტების დამატების ან რომელიმე ელემენტების წაშლის გზით. Python-ის ენაზე სიის ელემენტები კვადრატულ ფრჩხილებში იწერება და ერთმანეთისგან მძიმით გამოიყოფა. მაგალითისათვის ქვემოთ მოყვანილია ცვლად `lst`-ში შენახული კვირის დღეების სია.

დაიმახსოვრე

სია
სიის ელემენტები
ინდექსი
ნაწილი
ციკლი

```
>>>lst = ['ორშაბათი', 'სამშაბათი', 'ოთხშაბათი', 'ხუთშაბათი',
'პარასკევი', 'შაბათი', 'კვირა']
```

Python-ის ენაზე სიის ელემენტები ნოლიდან იწმრება. ზემოთ მოყვანილ მაგალითში ელემენტ 'ორშაბათი'-ს ნომერი (ინდექსი) 0-ის ტოლია. იმისათვის, რომ მივმართოთ სია `lst` -ის ინდექს `i`-ს, საჭიროა, უბრალოდ, მივუთითოთ `lst[i]`.

```
>>>lst[2]
'ოთხშაბათი'
```

სიაში ადვილად შეიძლება ნებისმიერი ელემენტის შეცვლა. ამისათვის ელემენტს, რომელსაც შესაბამისი ინდექსი გააჩნია, უბრალოდ, ახალი მნიშვნელობა უნდა მივანიჭოთ. მინიჭება `lst[i] = x` ცვლის `i` ინდექსის

მქონე ელემენტი lst-ის მნიშვნელობას x -ით, მაგალითად, ბრძანება `lst [3] = 'მე-4 დღე'`-ის შესრულებით კვირის დღეების სიაში შეიცვლება მე-4 ელემენტი (ინდექსით 3).

```
>>>lst[3] = 'მე-4 დღე'
>>>lst
['ორშაბათი', 'სამშაბათი', 'ოთხშაბათი', 'მე-4 დღე', 'პარასკევი', 'შაბათი', 'კვირა']
```

ზოგჯერ საჭირო ხდება არა მთელი სიით, არამედ მისი რაღაც ნაწილით (ქვესიით) სარგებლობა. ასე მაგალითად, თუ სია lst-იდან ჩვენთვის აუცილებელია მხოლოდ i-დან j-მდე ინდექსის მქონე ელემენტები, ეს ასე შეიძლება ვაჩვენოთ: `lst [ i : j ]` (გაითვალისწინეთ, რომ ელემენტი ინდექსით i შედის ბოლო სიაში, ხოლო ელემენტი ინდექსით j – არა).

```
>>>lst[1:3]
['სამშაბათი', 'ოთხშაბათი']
```

თქვენ გაეცანით ციკლებს როგორც ALPLogo-ს პროგრამულ გარემოში, ისე Python-ში. დაპროგრამებისას, განსაკუთრებით, სიებზე (ზევრ ენაზე ისინი მასიურია), მუშაობის დროს ძალზე მოსახერხებელია ციკლების გამოყენება, მაგალითად, ამ პროგრამის ფრაგმენტის შესრულების დროს.

```
lst = ['ორშაბათი', 'სამშაბათი', 'ოთხშაბათი', 'მე-4 დღე', 'პარასკევი', 'შაბათი', 'კვირა']
for s in lst:
    print(s)
```

კვირის დღეებს პროგრამა შემდეგნაირად გამოიტანს:

```
ორშაბათი
სამშაბათი
ოთხშაბათი
ხუთშაბათი
პარასკევი
შაბათი
კვირა
```

ისეთი სიებისთვის, რომლებიც მხოლოდ რიცხვებისგან შედგება, სპეციალური ფუნქციები არსებობს. მაგალითად, ფუნქცია `sum ()` ითვლის სიის ელემენტების ჯამს.

```
>>>mas = [1, 2, 3, 4]
>>>sum(mas)
```

```
10
```


ცხრილში წარმოდგენილია სიებთან დაკავშირებული ძირითადი ფუნქციები (ოპერაციები) და მათი აღწერილობა.

ფუნქცია ანუ ოპერაცია	აღწერილობა
<code>len(lst)</code>	ადგენს სიის ელემენტების რაოდენობას, მაგალითად: <pre>>>> lst = [1, 2, 3] >>> len(lst) 3</pre>
<code>lst1 + lst2</code>	სია <code>lst 2</code> -ის ელემენტები ემატება სია <code>lst 1</code> -ს ბოლოში, მაგალითად: <pre>>>> lst1 = [1, 2, 3] >>> lst2 = ['ერთი', 'ორი', 'სამი'] >>> lst1 + lst2 [1, 2, 3, 'ერთი', 'ორი', 'სამი']</pre>
<code>min(lst)</code>	განსაზღვრავს სიის მინიმალურ ელემენტს, მაგალითად: <pre>>>> lst1 = [1, 2, 3, 85, -42, 33, 84] >>> min(lst1) -42</pre> სტრიქონულ სიდიდეებში საფუძვლად ანბანური მიმდევრობა ითვლება <pre>>>> lst2 = ['შვიდი', 'რვა', 'ცხრა'] >>> min(lst2) 'რვა'</pre>
<code>max(lst)</code>	განსაზღვრავს სიის მაქსიმალურ ელემენტს, მაგალითად: <pre>>>> lst = [1, 2, 3, 85, -42, 33, 84] >>> max(lst) 85</pre>
<code>del lst[i]</code>	სიიდან შლის ელემენტს ნომრით <code>i</code> , მაგალითად: <pre>>>> lst = [1, 2, 3, ['ერთი', 'ორი', 'სამი']] >>> del lst[2] >>> lst [1, 2, ['ერთი', 'ორი', 'სამი']]</pre>

სტრიქონის მსგავსად, სიასაც Python-ში თავისი მეთოდები აქვს. ქვემოთ ცხრილში ნაჩვენებია ზოგიერთი მეთოდი, რომლებიც ხშირად გამოიყენება სიებზე მუშაობის დროს.

მეთოდი	აღწერილობა
<code>lst.append(x)</code>	სიას ბოლოში ემატება ელემენტი <code>x</code> , მაგალითად: <pre>>>>lst = [1, 2, 3] >>>lst.append(4) >>>lst [1, 2, 3, 4]</pre>
<code>lst.count(x)</code>	განისაზღვრება <code>x</code> -ის ტოლი ელემენტების რიცხვი, მაგალითად: <pre>>>>lst = ['ერთი', 'ორი', 'ორი', 'სამი'] >>>lst.count(ორი) 2</pre>
<code>lst.index(x)</code>	განისაზღვრება ელემენტი <code>x</code> -ის სიაში პირველი ჩართვის ადგილი (ინდექსი), მაგალითად: <pre>>>>lst = ['ერთი', 'ორი', 'ორი', 'სამი'] >>>lst.index(ორი) 1</pre>
<code>lst.remove(x)</code>	შლის პირველ ელემენტს <code>x</code> -ის მითითებული მნიშვნელობით, მაგალითად: <pre>>>>lst = ['ერთი', 'ორი', 'ორი', 'სამი'] >>>lst.remove('ორი') >>>lst ['ერთი', 'ორი', 'სამი']</pre>
<code>lst.insert(i,x)</code>	ჩასვამს სიაში <code>i</code> პოზიციაზე ელემენტი <code>x</code> -ს ან სიას, მაგალითად: <pre>>>>lst = ['ერთი', 'ორი', 'ორი', 'სამი'] >>>lst.insert(2, 'ერთი') >>>lst ['ერთი', 'ორი', 'ერთი', 'ორი', 'სამი']</pre>
<code>lst.sort()</code>	დაახარისხებს სიას ზრდადობის მიხედვით (ალფაბეტის მიხედვით), მაგალითად: <pre>>>>lst = ['ერთი', 'ორი', 'ერთი', 'ორი', 'სამი'] >>>lst.sort() >>>lst ['ერთი', 'ერთი', 'ორი', 'ორი', 'სამი']</pre>
<code>lst.reverse()</code>	დაალაგებს სიის ელემენტებს უკუღმა თანამიმდევრობით, მაგალითად: <pre>>>>lst = [1, 2, 3, 'ერთი', 'ორი', 'სამი'] >>>lst.reverse() >>>lst ['სამი', 'ორი', 'ერთი', '3', '2', '1']</pre>

როგორც ხედავთ, Python-ის ენაზე შედგენილი სიები მონაცემთა ძალზე მოსახერხებელი სტრუქტურაა და მათი გამოყენებით სხვადასხვა ამოცანის გადაწყვეტა შეიძლება.

სტრიქონის სიად გარდასაქმნელად Python-ის ენაში ფუნქცია `list()`-ს იყენებენ, მაგალითად:

```
>>> s = 'სტეპი'
>>> lst = list(s)
>>> lst
['ს', 'ტ', 'ე', 'პ', 'ი']
```

გავეცნოთ კიდევ ორ მეთოდს, რომლებსაც ხშირად იყენებენ ამოცანების ამოხსნის დროს. ერთი მათგანია მეთოდი `join()`. ეს მეთოდი სიის ელემენტებს შორის მითითებულ სტრიქონს სვამს და ერთ სტრიქონს ქმნის (ანუ ამ სტრიქონის მეშვეობით სიის ელემენტები ერთიანდება). გთავაზობთ მაგალითს:

```
>>> lst = ['1', '2', '3']
>>> s = '---'.join(lst)
>>> s
'1---2---3'
```

ამ მეთოდის საპირისპიროდ, მეთოდი `split()` ყოფს სტრიქონს მოცემული გამყოფი სიმბოლოს დახმარებით და ნაწილებისგან ქმნის სიას.

```
>>> s = '12 34 56'
>>> lst = s.split(' ')           # გამყოფი სიმბოლო
>>> lst
['12', '34', '56']
```

მაგალითი

მართლწერის წესების თანახმად, ტექსტში მძიმეს ყოველთვის შუალედი მოსდევს. ქვემოთ მოცემული პროგრამა პოულობს მოყვანილ ტექსტში ამ ტიპის შეცდომებს და ასწორებს მათ. შეამოწმეთ პროგრამის მუშაობა.

პროგრამაში გამოყენებული ცვლადების აღწერა

`s` – შესატანი ტექსტი;
`lst` – ტექსტის შესატყვისის სია;
`i` – სიის ელემენტების ინდექსი.

ამოცანის ამოხსნის ალგორითმი

1. პროგრამა შეიტანს ტექსტს (სტრიქონს);
2. ფუნქცია `list()`-ის დახმარებით სტრიქონი სიად გარდაიქმნება;
3. სიის ყველა ელემენტი, თავიდან ბოლომდე, სათითაოდ მოწმდება. თუ ელემენტი, რომელიც მოსდევს ელემენტს, რომლის სიმბოლოა `' '`, შუალედი არ არის, მაშინ სიის ამ ელემენტის შემდეგ ემატება ახალი ელემენტი – შუალედი;
4. ახალი სია სტრიქონად გარდაიქმნება;
5. პროგრამა გამოიტანს სახეშეცვლილ ტექსტს.

```
s = input('შეიტანეთ ტექსტი:')
lst = list(s)

i = 0
while i < len(lst):
    if (lst[i] == ',' and lst[i+1] != ' '):
        lst.insert(i+1, ' ')
    i = i + 1

s = ''.join(lst)
print(s)
```

შევისწავლოთ დამოუკიდებლად

გამოიყენეთ ფუნქციები `list()` და `count(x)` და დაწერეთ პროგრამა, რომელიც მოცემულ ტექსტში ასო 'ა'-ს რაოდენობას გამოითვლის.

სიის შემადგენელ ობიექტებს მის ელემენტებს უწოდებენ. სიებში შეიძლება რიცხვების, სტრიქონების, სხვა სიების ან ყველაფრის ერთად შენახვა. სიის ყოველ ელემენტს შეესაბამება რიგითი ნომერი (ინდექსი), რომლითაც მას შეიძლება მივმართოთ.

შაამოწაე შენი ცოდნა

1. რა არის სია და რა სიდიდეები შეიძლება იყოს მისი ელემენტები?
2. რა მოქმედებების შესრულება შეიძლება სიებზე?
3. გამოიყენეთ ფუნქციები `range()` და `list()` და 1-დან 100-მდე რიცხვებისგან შეადგინეთ სია.
4. დაწერეთ პროგრამა, რომელსაც შეუძლია, აღმოაჩინოს ტექსტში, სად არ არის დასმული სასვენი ნიშნების შემდეგ შუალედები და გაასწოროს ეს შეცდომები;
5. დაწერეთ პროგრამა, რომელსაც შეუძლია, მოცემულ სტრიქონში მოძებნოს ყველაზე გრძელი და ყველაზე მოკლე სიტყვა.

17

ფუნქცია



- რა არის ქვეპროგრამა და რა უპირატესობას ანიჭებს ის პროგრამისტს?
- რა ფუნქციას ასრულებს ბრძანება `input()` ?

საქმიანობა

1

გადახედეთ მოცემულ პროგრამულ კოდს და დაადგინეთ, რა ამოცანის შესრულებისთვისაა ის განკუთვნილი.

```
n = abs(int(input()))
a = (n // 100) % 10
b = (n // 10) % 10
c = n % 10
print(a, b, c)
```

ვიმსჯელოთ:

– როგორ აღწეროთ სიტყვებით ბრძანებათა თანამიმდევრობა პირველ სტრიქონში?

პროგრამისტებს შეუძლიათ, გაიგონ რამდენიმე ასეული სტრიქონისგან შედგენილი პროგრამული კოდი. მაგრამ რაც მეტი სტრიქონია პროგრამაში, მით უფრო რთულია მისი გაგება. მაშინაც კი, როდესაც ყოველი ოპერატორის ფუნქცია ნათელია, ასეთ პროგრამებში ძნელი განსასაზღვრია, რა მიზნითაა გამოყენებული მოცემულ ადგილას ესა თუ ის მათგანი. ასეთ პროგრამებში ცვლილებების შეტანა და შეცდომების გასწორებაც საკმაოდ ძნელია.

ამ პრობლემის მოგვარების მიზნით პროგრამას ყოფენ **ქვეპროგრამებად**, რომლებიც მარტივ ოპერაციებს ასრულებს. ამგვარად, საბოლოო სახის პროგრამა შედგება არა ცალკეული ოპერატორების, არამედ კოდების ცალკეული ბლოკებისაგან, რომელთაგან თითოეულს თავისი სახელი აქვს. ქვეპროგრამების გამოყენება ზოგავს პროგრამის დაწერისთვის საჭირო დროს, ამცირებს პროგრამული კოდის მოცულობას და პროგრამის სტრუქტურას უფრო გასაგებს ხდის. ამავე დროს, ასეთი მიდგომა პროგრამის დაწერისას დაშვებული შეცდომების რიცხვის შემცირებასაც განაპირობებს.

დამიხსოვრე

ქვეპროგრამა
ფუნქცია
ფუნქციის გამოძახება
ფორმალური პარამეტრები
ფაქტიური პარამეტრები

პროგრამებს, ჩვეულებრივ, ორ კატეგორიად: **პროცედურებად** და **ფუნქციებად** ყოფენ. **პროცედურა** მხოლოდ ოპერატორების თანამიმდევრობის შესრულებას ემსახურება. თქვენთვის ALPLogო-ს პროგრამული გარემოდან ნაცნობი ქვეპროგრამები პროცედურებს წარმოადგენს. **ფუნქციები** ს დანიშნულება გარკვეული მნიშვნელობების გამოთვლა და მათი პროგრამისთვის (ქვეპროგრამისთვის) დაბრუნებაა. დაპროგრამების ზოგიერთ ენაში, მათ შორის, Python-ში, ქვეპროგრამები პროცედურებად და ფუნქციებად არ იყოფა. ყველა მათგანი ფუნქციებად განიხილება. ასეთ ენებში პროცედურები არის ფუნქციები, რომლებიც არანაირ მნიშვნელობას არ აბრუნებს პროგრამაში. როგორც

აღინიშნა, ყოველ ფუნქციას თავისი სახელი აქვს. ნებისმიერი ფუნქციის შესასრულებლად საჭიროა, მიმართონ მის სახელს, ანუ ფუნქცია უნდა გამოიძახონ. ფუნქციის გამოძახება პროგრამის ნებისმიერი ადგილიდან შეიძლება.

Python-ის ენაში მრავალი სხვადასხვა დანიშნულების ფუნქცია არსებობს. ასე მაგალითად, ბლოკ „საქმიანობაში“ მოცემულ პროგრამაში გამოყენებულია 4 ფუნქცია: `abs()`, `int()`, `input()` და `print()`. ეს Python-ის სტანდარტული ფუნქციებია. მიუხედავად ამისა, ამოცანების ამოხსნისას ხშირად ჩნდება ახალი, საკუთარი ფუნქციების შექმნის აუცილებლობა. ბუნებრივია, Python-ის ენას გააჩნია ახალი ფუნქციების შექმნის შესაძლებლობა.

Python-ის ენაზე ფუნქციის აღწერა იწყება საკვანძო სიტყვა `def`, -ით, რომლის შემდეგ იწერება ფუნქციის სახელი. მრავალი ფრჩხილები, რომლებიც ფუნქციის სახელს მოსდევს, მას ჩვეულებრივი ცვლადისგან განასხვავებს. ორწერტილი მიუთითებს ფუნქციაში შემავალი ბრძანებების დასაწყისზე. ალვწეროთ, მაგალითად, მარტივი ფუნქცია, სახელწოდებით `greeting()`:

```
defgreeting():
    print('კეთილი იყოს თქვენი მობრძანება!')
```

პროგრამის ნებისმიერი ადგილიდან ფუნქციის, ისევე როგორც სტანდარტული ფუნქციის, გამოსაძახებლად აუცილებელია მისი სახელის (როგორც ცალკე ოპერატორის) მითითება.

```
greeting()
```

ამ ფუნქციის მუშაობის შედეგი ყოველთვის ერთნაირია, ანუ, მათემატიკის ენაზე რომ ვთქვათ, მას არგუმენტები არ აქვს. პროგრამირებაში ასეთი ფუნქციები, როგორც წესი, არ გამოიყენება. ჩვეულებრივ, ფუნქცია მონაცემებს იმ პროგრამიდან იღებს, რომელმაც ის გამოიძახა. მონაცემები ფუნქციას პარამეტრების სახით გადაეცემა. ფუნქციის შექმნისას მისი პარამეტრები ჯერ

```
defprivet(s):
    print(s)
    privet('დილა მშვიდობის!')
    privet('სადამო მშვიდობის!')
```

ფორმალური პარამეტრი

ფაქტობრივი პარამეტრი

ცნობილი არ არის. ფუნქციის აღწერისას მისი სახელის შემდეგ ე.წ. **ფორმალურ პარამეტრებს** მიუთითებენ. ფორმალური პარამეტრებია ნებისმიერი ცვლადები, რომლებიც გადასაცემ მონაცემებს განსაზღვრავს. ისინი მხოლოდ ფუნქციების მიერ შესრულებული მოქმედებების აღსაწერად გამოიყენება. ფუნქციის გამოძახებისას მისი სახელის შემდეგ მიუთითებენ ე.წ.

ფაქტობრივ პარამეტრებს, რომლებიც ამ ფუნქციას გადაეცემა. ფუნქციის ოპერატორების შესრულებისას ფორმალური პარამეტრები ფაქტიური მნიშვნელობებით იცვლება.

აღნიშნული განვიხილოთ ერთი პროგრამის მაგალითზე. ეს პროგრამა მუსულმანური კალენდრის, ჰიჯრის, შეყვანილ წელს გრიგორიანული (ევროპული) კალენდრის წლად (და პირიქით) გარდაქმნის და შედეგს ეკრანზე გამოიტანს. ამ დროს გამოთვლები შემდეგი ფორმულებით წარმოებს: $Q = X - X / 33 + 622$ და $X = Q - 622 + (Q - 622) / 32$. აქ ცვლადი Q გრიგორიანული კალენდრის წელზე მიუთითებს, ხოლო X – ჰიჯრის წელზე.

აღწეროთ ორი ფუნქცია: ერთი მათგანი – ფუნქცია `miladi_hicri()` ფორმულის საფუძველზე, პარამეტრის სახით გადაცემულ გრიგორიანული კალენდრის წელს ჰიჯრის წლად გარდაქმნის, მეორე ფუნქცია – `hicri_miladi()` პირიქით, ჰიჯრის წელს გრიგორიანულ წლად გადააქცევს და შედეგს ეკრანზე გამოიტანს.

```
def miladi_hicri(S):
    Q = int(S)
    X = Q - 622 + (Q - 622) // 32
    print(' ჰიჯრის წელი: ', X)

def hicri_miladi(S):
    X = int(S)
    Q = X - (X // 33) + 622
    print(' გრიგორიანული წელი: ', Q)

S = input(' გრიგორიანული წელი: ')
miladi_hicri(S)

S = input(' ჰიჯრის წელი: ')
hicri_miladi(S)
```

ამ მაგალითში თითოეული ფუნქცია გარედან მიღებული მნიშვნელობების საფუძველზე აწარმოებს გამოთვლას და შედეგს ეკრანზე გამოიტანს. მაგრამ უმეტეს შემთხვევაში პრაქტიკული ამოცანების გადაწყვეტისას უფრო მოსახერხებელია ფუნქციის მიღებული მნიშვნელობის იმ პროგრამისთვის დაბრუნება, რომელმაც ის გამოიძახა. ამ მიზნით პროგრამა Python-ში გათვალისწინებულია ბრძანება `return`. ეს ბრძანება ფუნქციის ბოლოში იწერება და მის შემდეგ ფრჩხილებში მიუთითებენ მნიშვნელობას, რომელიც ფუნქციამ უნდა დააბრუნოს.

როგორ იცით, ფუნქცია `input()`, კლავიატურიდან შეყვანილი სიდიდის ტიპისგან დამოუკიდებლად, ყოველთვის აბრუნებს სტრიქონს, ქვემოთ მოყვანილი ახალი ფუნქცია `num_input()` კი აბრუნებს რიცხვს.

```
def num_input(prompt):
    typed = input(prompt) # შესაყვანი სტრიქონი
                        # ინახება ცვლად typed-ში
    num = int(typed)      # სტრიქონი გარდაქმნება რიცხვად და
                        # ინახება ცვლადში
    return num           # ცვლადის მნიშვნელობა ბრუნდება

a = num_input(' შეიყვანეთ a: ')
b = num_input(' შეიყვანეთ b: ')
print('a * b =', a * b)
```

შავისნავლოთ დამოუკიდებლად

კალენდართან დაკავშირებული პროგრამა შეიძლება ისე შეიქცვალოს, რომ ახალი კალენდრის მნიშვნელობები ეკრანზე გამოდიოდეს არა ფუნქციის შემადგენლობაში, არამედ მის გამომძახებელ პროგრამაში.

```
def miladi_hicri(S):  
    Q = int(S)  
    X = Q - 622 + (Q - 622) // 32  
    return (X)  
  
def hicri_miladi(S):  
    X = int(S)  
    Q = X - (X // 33) + 622  
    return (Q)  
  
S = input(' გრიგორიანული წელი: ')  
X = miladi_hicri(S)  
print(' შიჯრის წელი: ', X)  
  
S = input(' შიჯრის წელი: ')  
Q = hicri_miladi(S)  
print(' გრიგორიანული წელი: ', Q)
```

გაანალიზეთ პროგრამის მუშაობა.

შეამოწმე შენი ცოდნა

1. რა უპირატესობას იძლევა ქვეპროგრამების გამოყენება?
2. რა ტიპის ქვეპროგრამები არსებობს?
3. რით განსხვავდება ფუნქცია პროცედურისგან?
4. Python-ის ენის რომელი ფუნქცია შეესაბამება კომპიუტერის გამომტან მოწყობილობებს?
5. გამოიყენეთ კალენდრის პროგრამა და დაადგინეთ თქვენი დაბადების წელი მუსულმანური წელთაღრიცხვით.
6. დაწერეთ ფუნქცია, რომელიც გამოთვლის სამკუთხედის პერიმეტრს მისი გვერდების მნიშვნელობის მონაცემების საფუძველზე.



- დაპროგრამების რომელი ენა უფრო მოგწონთ: ALPLogo თუ Python? რატომ?
- ამ ენებიდან რომელით ურჩევდით დაპროგრამების შესწავლის დაწყებას უმცროსკლასელებს?

საქმიანობა

1

მოცემულია პროგრამების ფრაგმენტები დაპროგრამების ორ სხვადასხვა ენაზე – BASIC და Pascal. გადახედეთ ფრაგმენტებს და შეეცადეთ დაადგინოთ, რას აკეთებს ისინი.

BASIC

```
S = 0
FOR I = 1 TO 100
    S = S + I
NEXT I
```

Pascal

```
a := 5;
b := 10;
if a > b then write(a)
else write(b);
```

ვიმსჯელოთ:

- რომელი ოპერატორებია თქვენთვის ნაცნობი?
- როგორ ჩაინერება ფრაგმენტების მონაცემები Python-ის ენაზე?

ის, ვინც პროგრამირებას სწავლობს, ადრე თუ გვიან იკითხავს: დაპროგრამების რომელი ენაა უკეთესი? მიუხედავად იმისა, რომ დაპროგრამების ათასზე მეტი ენაა ცნობილი, არ არსებობს არც ერთი ისეთი, რომელიც ყველა პროგრამისტს აწყობს. ყოველ ენას თავისი მომხრეები და მოწინააღმდეგეები ჰყავს.

სხვადასხვა სფეროში მოღვაწე ადამიანების – ინჟინრების, ბანკირების, სამხედროების – წინაშე სხვადასხვა ტიპის ამოცანები დგას, რომელთა გადაწყვეტისას უპირატესობა დაპროგრამების ამა თუ იმ ენას ენიჭება. ენა FORTRAN ფართოდ გამოიყენება მათემატიკური ამოცანების ამოსახსნელად, ენა COBOL („კობოლ“) – საბანკო საქმეში, ენები: PROLOG, LISP გამოიყენება ხელოვნური ინტელექტის სფეროში. პროგრამისტები, რომლებიც ინტერნეტისთვის ქმნიან პროგრამებს, უპირატესობას, ჩვეულებრივ, ენა JAVA-ს ანიჭებენ.

დაპროგრამების ყველა ჩამოთვლილი ენა **სპეციალური დანიშნულების ენაა**. თითოეულ მათგანს ისეთი ოპერატორები გააჩნია, რომელთა დახმარებით სპეციფიკური ამოცანების მარტივად გადაწყვეტა შეიძლება. ჩვეულებრივ, ამ ენებს უშუალოდ სამუშაო ადგილებზე სწავლობენ, ვინაიდან წინასწარ მათ ათვისებას აზრი არ აქვს.

დამიხსოვრე

სპეციალური დანიშნულების ენა ზოგადი დანიშნულების ენა

სპეციალური დანიშნულების დაპროგრამების ენების გარდა, არსებობს ზოგადი დანიშნულების ენებიც. მათი დახმარებით პრაქტიკულად ყველა ამოცანის გადანყვეტა შეიძლება. ასეთი ენების შესწავლა წინასწარვე შეიძლება – ისინი ყოველთვის შეიძლება დაგვჭირდეს. მათ შორის ყველაზე პოპულარულია Basic, Pascal, C++, Python.

ამჟამად მსოფლიოში 4000-ზე მეტი დაპროგრამების ენა არსებობს, რომელთაგან ბევრი დღეს აღარც გამოიყენება. გარდა ამისა, პროგრამების განსაკუთრებით მნიშვნელოვანი ნაწილი სრულდება არა ზოგადი დანიშნულების დაპროგრამების ენებზე, არამედ დაპროგრამების სპეციალური ინსტრუმენტების (მაგალითად, Visual Basic-ის) ან პროგრამული უზრუნველყოფის სპეციალური ტიპების (მაგალითად, Maple-ის) გამოყენებით.

დაპროგრამების სპეციალური ენების ქრონოგრაფი

დასახელება	წელი	ავტორი	შექმნის გეოგრაფია
FORTRAN	1954	ჯონ ბეკუსი	აშშ
LISP	1958	ჯონ მაკარტი	აშშ
Algol 60	1960	პიტერ ნაური და სხვ.	სხვადასხვა ქვეყანა
COBOL	1960	ავტორთა კოლექტივი	სხვადასხვა ქვეყანა
Simula	1962	კრისტენ ნიგარდი და სხვ.	ევროპა
BASIC	1963	ჯონ კემენი და სხვ.	აშშ
PL/I	1964	ჯორჯ რადინი	სხვადასხვა ქვეყანა
Algol 68	1968	აად ვან ვაინგარტენი	სხვადასხვა ქვეყანა
Pascal	1970	ნიკლაუს ვირტი	ევროპა
FORTH	1970	ჩარლზ მური	აშშ
C	1972	დენის რიჩი	აშშ
Smalltalk	1972	ალან კეი	აშშ
Prolog	1973	ალან კოლმერო და სხვ.	ევროპა
Ada	1980	ჯინ იშპია და სხვ.	აშს
C++	1984	ბორნ სტრაუსტრუპი	აშშ
Python	1991	გვიდო ვან როსუმი	აშშ
Java	1995	ჯეიმს გოსლინგი	აშშ

დანიშნულების მიხედვით დაპროგრამების ენები შეიძლება დაახლოებით ასე დაჯგუფდეს:

1. დიდი, რთული პროგრამების შესაქმნა: ზოგადი დანიშნულების პროგრამირების ენები (PL/L, C, C++, Pascal, Python, Modula-2, Ada, Java, C#).
2. მცირე პროგრამების შექმნა: ზოგადი დანიშნულების დაპროგრამების ენები (BASIC, Visual Basic, Pascal, Python).
3. მათემატიკური გამოთვლები, მეცნიერება და ინჟინერია: FORTRAN, APL, Maple და ზოგადი დანიშნულების ენები, რომლებიც ზემოთ არის დასახელებული.
4. მონაცემთა კომერციული დამუშავება: COBOL, RPG.
5. ხელოვნური ინტელექტის ამოცანები და ამოცანები რთული ლოგიკით: LISP და Prolog.
6. სტრიქონებზე და სცენარებზე მუშაობა: SNOBOL, REXX, Awk, Perl, Python, VBSCRIPT, JavaScript.

შავისნავლოთ დამოუკიდებლად

ზოგიერთი დაპროგრამების ენა არაორდინალური ამოცანების ამოსახსნელადაა შექმნილი. მაგალითისთვის, Piet-ის ენაზე დაწერილი პროგრამა აბსტრაქტული ნახატივით გამოიყურება. ქვემოთ წარმოდგენილი ნახატი არის პროგრამა „Hello World!“ („გამარჯობა, მსოფლიო!“), რომელიც ამ ენაზეა დაწერილი.



გამოარკვევით, საიდანაა აღებული ამ პროგრამის სახელწოდება. შეაგროვეთ ინფორმაცია სხვა ეგზოტიკური დაპროგრამების ენების შესახებ, როგორებიცაა **Chef, Ook!, Malkbolge**.

შეამოწმა შენი ცოდნა ?

1. დაპროგრამების რომელი ენაა უკეთესი?
2. რომელ ენას მიეკუთვნება Python: სპეციალური დანიშნულების თუ ზოგადი დანიშნულების დაპროგრამების ენას?
3. ვინ და როდის შექმნა Python-ის ენა?
4. რომელი ენები გამოიყენება, ძირითადად, მათემატიკური გამოთვლებისთვის?
5. დაპროგრამების რომელი ენები მიეკუთვნება სპეციალური დანიშნულების ენების კატეგორიას?

1. რომელი მათემატიკური მოქმედებები აღინიშნება // და %? სიმბოლოებით?

2. რა მნიშვნელობას მიიღებს მთელის ტიპის x და y ცვლადები პროგრამის ფრაგმენტის შესრულების შემდეგ?

$x = 823$

$y = 4$

$x = x // y$

$y = x \% y$

3. შეადგინეთ პროგრამა, რომელიც ადგენს მოცემული რიცხვის 4-ზე გაყოფადობას.

4. შეადგინეთ პროგრამა, რომელიც კლავიატურიდან შეიტანს ანდაზას „ქარის მოტანილს ქარი წაიღებსო“ და გამოიტანს:

ა) სიტყვას „ქარის“;

ბ) „ა“ ასოების რაოდენობას;

გ) მთელ ანდაზას მთავრული ასოებით.

5. რომელი მეთოდი გამოიყენება კონკრეტული ელემენტის სიიდან ამოსაშლელად?

5. შეადგინეთ პროგრამა, რომელიც უკულმა თანამიმდევრობით გამოიტანს შეტანილ წინადადებას.

6. აღწერეთ წინადადებაში ყველაზე მოკლე სიტყვის მოძებნისთვის განკუთვნილი ალგორითმი.

7. რას აკეთებს ეს პროგრამა?

`s = input()`

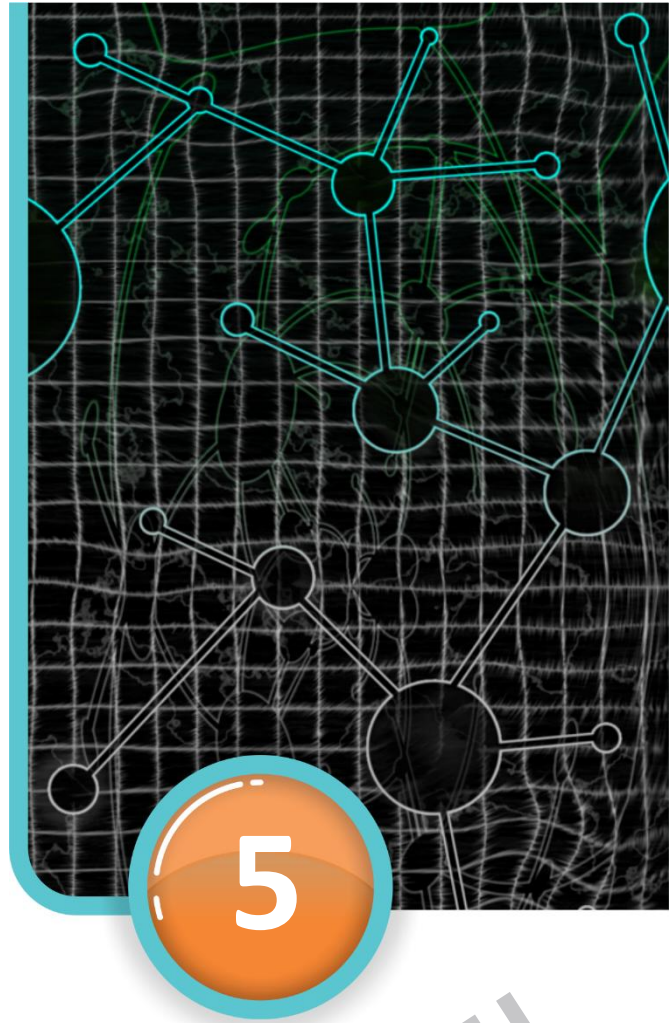
`l = len(s)`

`n = int(input("შეიცვანეთ სიტყვის რიგითი ნომერი წინადადებაში:"))`

`s = s.split(' ')`

`print(s[n-1])`

8. რა მიზნით იყენებენ ქვეპროგრამებს?



ინფორმაციული ტექნოლოგიები

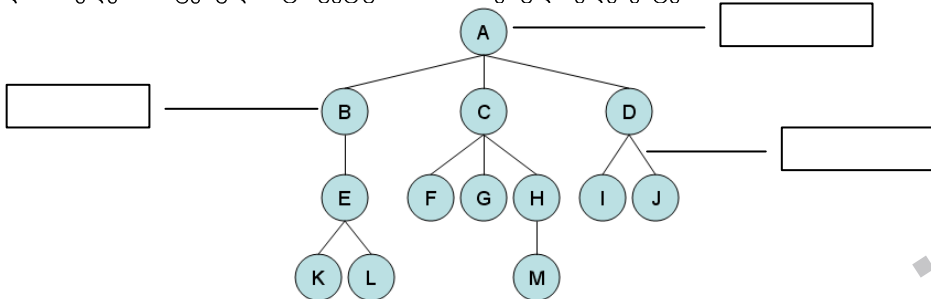
19. ინფორმაციული მოდელი გრაფებზე
20. გრაფებთან დაკავშირებული ამოცანები
21. კომპიუტერული ქსელები
22. ინფორმაციის გადაცემა
23. ვებსაიტების შექმნა
24. საინფორმაციო საზოგადოებისადმი მიძღვნილი საიტის შემუშავება
25. დამისამართება ინტერნეტში



- რა არის ინფორმაციული მოდელი?
- შეიძლება თუ არა, ერთ ობიექტს რამდენიმე სახის ინფორმაციული მოდელი ჰქონდეს?

საქმიანობა

დაასახელეთ მოცემული სტრუქტურის მითითებული ელემენტები.



ვიმსჯელოთ:

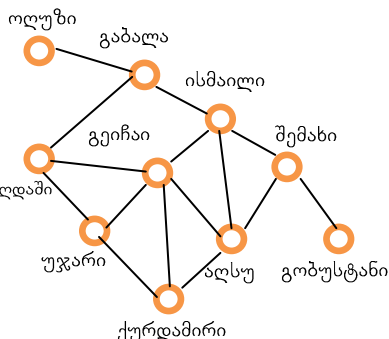
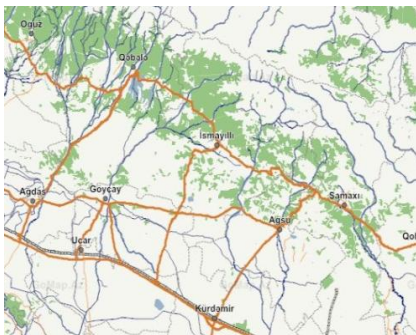
- რა სახის ინფორმაციულ მოდელს მიეკუთვნება მოცემული სტრუქტურა?
- რა არის გზა და რამდენი გზა არის მოცემული სტრუქტურის ნებისმიერ ორ წვეროს შორის?

მე-7 კლასში თქვენ გაეცანით სიის სახის, მე-8 კლასში კი – ხის სახის ინფორმაციულ მოდელს. ახლა კი მის კიდევ ერთ სახეს – გრაფს ანუ ქსელს – გაეცნობით.

რუკის მოცემულ ფრაგმენტზე ნაჩვენებია აზერბაიჯანის ერთ-ერთი რეგიონის რაიონული ცენტრების დამაკავშირებელი საავტომობილო გზები. თუ რაიონულ ცენტრებს მსხვილი ნერტილებით (ან პატარა რგოლებით) აღვნიშნავთ, გზას კი – მათი შემაერთებელი ხაზებით, შემდეგ სქემას მივიღებთ:

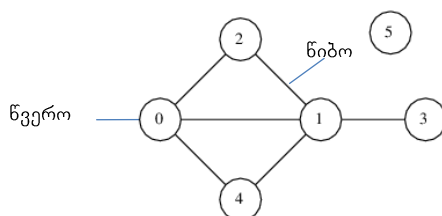
დაიმახსოვრა

გრაფი
წვერო
ნიბო
მოსაზღვრე წვეროების სია
მოსაზღვრეობის მატრიცა
ბმული გრაფი
ორიენტირებული გრაფი



რამდენიმე ობიექტის ერთმანეთთან ნებისმიერი სახით დამაკავშირებელ სტრუქტურას **გრაფს** უწოდებენ. პროგრამირებაში ტერმინ „გრაფის“ ნაცვლად ზოგჯერ სიტყვა „ქსელს“, ხმარობენ. გრაფის შემადგენელ ობიექტებს **წვეროები** ჰქვია, ხოლო ობიექტების შემაერთებელ ხაზებს – **ნიბოები**. არ არის სავალდებულო, რომ გრაფის ნებისმიერ ორ წვეროს შორის იყოს კავშირი. გრაფის წვეროებს, ჩვეულებრივ, ნომრავენ ან ასოებით აღნიშნავენ.

ჩვეულებრივი გრაფი



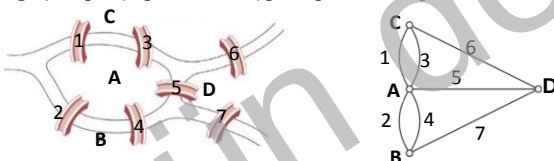
ეს საინტერესოა...



ლეონარდ ეილერი
(1707-1783)

გრაფების თეორიის შექმნის საბაზი ცნობილი მათემატიკოსის, ლეონარდ ეილერის, მიერ ერთი სახალისო ამოცანის ამოხსნის პროცესი გახდა. ეს 1736 წელს, მისი კენიგსბერგში (ახლანდელ კალინინგრადში) ყოფნის დროს მოხდა.

ამ ქალაქის ტერიტორიაზე გამავალი მდინარე მას ოთხ ნაწილად ყოფდა. ეს ნაწილები შვიდი ხიდით იყო ერთმანეთთან დაკავშირებული. ქალაქის გამარტივებულ გეგმაზე ხიდები ციფრებითაა აღნიშნული, ქალაქის ნაწილები კი – ასოებით.



ამოცანა (იმჟმად საკმაოდ პოპულარული) კი ასეთი იყო: როგორ გადავიაროთ ყველა ხიდი ისე, რომ არც ერთ მათგანზე ორჯერ არ მოვხვდეთ?

ეილერმა ქალაქის ნაწილები შესაბამისი A, B, C და D წერტილებით აღნიშნა, ხოლო ხიდები – ხაზებით, რომლებიც ამ წერტილებს აერთებდა (სქემა მარჯვნივ). ამგვარად, ამოცანა ასეთი დავალების ეკვივალენტური გახდა: შესაძლებელია თუ არა ამ ფიგურის შემოხაზვა ისე, რომ ფურცლიდან ხელი არ აიღო და თითოეულ ხაზს მხოლოდ ერთხელ გააყოლო? ეილერმა დაამტკიცა, რომ ამ ამოცანას ამონახსნი არა აქვს.

ჩვეულებრივ, გრაფებს ორი გზით წარმოადგენენ: მოსაზღვრე წვეროების სიით და მოსაზღვრეობის მატრიცით. **მოსაზღვრე წვეროების სიაში**

ჩამოთვლილია ის წვეროები, რომლებიც მოცემული წვეროებიდან ყველასთანაა შეერთებული. ამ ფორმაში გვხვდება გამეორებები, მაგალითად, თუ **A** და **B** წვეროები შეერთებულია, მაშინ **A** წვეროს მოსაზღვრე წვეროების სიაში იქნება **B** წვერო, ხოლო **B** წვეროს მოსაზღვრე წვეროების სიაში – **A** წვერო.

გრაფების წარმოდგენის მეორე გზის, მოსაზღვრეობის მატრიცის, შემთხვევაში გრაფს, რომელიც წვეროების n რაოდენობას შეიცავს, n -სტრიქონიანი და n -სვეტიანი ცხრილის სახით წარმოადგენენ (განზომილების მატრიცა $n \times n$). თუ ნებისმიერ x და y წვეროებს შორის არსებობს წიბო, მაშინ ელემენტი a_{xy} იქნება 1-ის ტოლი, საპირისპირო შემთხვევაში კი 0-ის ტოლი. მაგალითად, ზემოთ წარმოდგენილი გრაფის მოსაზღვრე წვეროების სია და მოსაზღვრეობის მატრიცა ამგვარი იქნება:

წვერო	მოსაზღვრეობის წვეროების სია
0	1, 2, 4
1	0, 2, 3, 4
2	0, 1
3	1
4	0, 1
5	

	0	1	2	3	4	5
0	0	1	1	0	1	0
1	1	0	1	1	1	0
2	1	1	0	0	0	0
3	0	1	0	0	0	0
4	1	1	0	0	0	0
5	0	0	0	0	0	0

კომპიუტერის მეხსიერებაში გრაფების შენახვის მიზნით, ამ მეთოდებთან ერთად, სხვა მეთოდებსაც იყენებენ. მაგალითად, თუ მოცემული გრაფის მოსაზღვრე წვეროების სიას Python-ის ენაზე დავარქმევთ `adjacency_list`-ს, ხოლო მისი წვეროების რაოდენობას – `num_vertices`-ს, მაშინ

```
adjacency_list = [[1, 2, 4],
                  [0, 2, 3, 4],
                  [0, 1],
                  [1],
                  [0, 1],
                  []]
```

```
num_vertices = len(adjacency_list)
```

თუ ამ გრაფისთვის მოსაზღვრეობის მატრიცას აღვნიშნავთ, როგორც `adjacency_matrix`-ს, მაშინ

```
adjacency_matrix = [[0, 1, 1, 0, 1, 0],
                    [1, 0, 1, 1, 1, 0],
                    [1, 1, 0, 0, 0, 0],
                    [0, 1, 0, 0, 0, 0],
                    [1, 1, 0, 0, 0, 0],
                    [0, 0, 0, 0, 0, 0]]
num_vertices = len(adjacency_matrix)
```

უნდა აღინიშნოს, რომ, ჩვეულებრივი (არაორიენტირებული) გრაფის მოსაზღვრეობის მატრიცა ყოველთვის სიმეტრიულია მთავარი დიაგონალის მიმართ. მატრიცის მთავარი დიაგონალი მარცხენა ზედა კუთხიდან და მარჯვენა ქვედა კუთხემდე ეშვება.

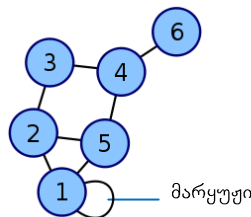
გრაფების შენახვა ნიბოების სიის ფორმითაც შეიძლება. ამ შემთხვევაში ყოველი ნიბო ორი რიცხვით – საწყისი და საბოლოო წვეროების ნომრებით – აღინიშნება.

გრაფის წვეროების რაოდენობა ცალკე ცვლადში ინახება, ვინაიდან იზოლირებული (ანუ ისეთი, რომელიც არც ერთ წვეროს არ უერთდება) წვეროები ნიბოების სიაში არ ხვდება.

```
num_vertices = 6          # წვეროების რაოდენობა
edges_list = [[0, 1],     # ნიბოების სია
               [0, 2],
               [0, 4],
               [1, 2],
               [1, 3],
               [1, 4],
               ]
```

გავეცნოთ გრაფებთან დაკავშირებულ კიდევ რამდენიმე ცნებას. ნიბოების თანამიმდევრობას, რომელშიც თითოეული ნიბოს ბოლო წვერო სხვა ნიბოს დასაწყისად გვევლინება (უკანსაკნელის გარდა) **გზა** ეწოდება. ჩაკეტილ გზას **ციკლი** ჰქვია. ზემოთ მოცემულ მაგალითში ნიბოები, რომლებიც 1-ელ, მე-2 და მე-3 წვეროებს აერთებს, ქმნის გზას, ხოლო გზა 1-ელ, მე-2 და მე-5 წვეროებს შორის ციკლია.

როგორც უკვე აღინიშნა, სრულიადაც არ არის სავალდებულო გრაფის ყველა წვეროს ერთმანეთთან შეერთება. თუ გრაფში არსებობს გზა მის ორ ნებისმიერ წვეროს შორის, მას **ბმულ** გრაფს უწოდებენ. თუ ნიბო წვეროს თავის თავთან აკავშირებს, ასეთ ნიბოს **მარყუჟს** უწოდებენ. თუ ნიბოს განსაზღვრული მიმართულება აქვს (მაგალითად, ნიბო A წვეროდან B წვეროსკენ კი არ მიდის, არამედ, A წვეროს შიგნიდან B წვეროში შედის), ასეთ ნიბოს **რკალს** უწოდებენ, ანუ ნიბო გრაფის ორ წვეროს აერთებს, ხოლო რკალი ერთი წვეროდან იწყება და მეორეში მთავრდება. გრაფს, რომლის ყველა ნიბო რკალია, **ორიენტირებული** გრაფი, ანუ **დიგრაფი** ჰქვია.



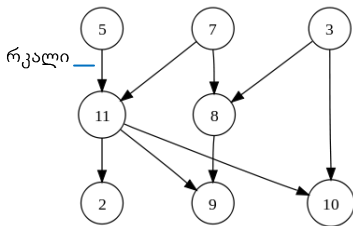
ორიენტირებული გრაფების კომპიუტერის მეხსიერებაში შენახვის მიზნით შეგვიძლია, ოდნავ შევცვალოთ ზემოთ აღწერილი გზები და წარმოდგენის შემდეგ საშუალებებს მივმართოთ:

- ყოველი წვეროს მოსაზღვრეობის სიაში შევინახოთ ის წვეროები, რომლებსაც ნიბოები მიემართება.
- მოსაზღვრეობის მატრიცაში, თუ არსებობს ნიბო i -დან j -მდე, ჩავთვალოთ, რომ $\text{adjacency_matrix}[i][j] == 1$, ხოლო თუ გრაფში საპირისპირო მიმართულების ნიბო არ არის, ჩავთვალოთ, რომ $\text{adjacency_matrix}[j][i] == 0$;
- ნიბოთა სიაში თითოეული ნიბო შევინახოთ [საწყისი, ბოლო]-ს სახით.

ბე (ხისებრი სტრუქტურა) გრაფის ერთ-ერთი სახეა. ეს არის ბმული გრაფი ციკლის გარეშე, ანუ ხის ნებისმიერ ორ წვეროს შორის არსებობს გზა, მაგრამ ჩაკეტილი გზები არ გვხვდება.

ინფორმაციული მოდელი გრაფებზე ჩვენი ცხოვრების მრავალ სფეროში გამოიყენება. მაგალითად, დასახლებულ პუნქტებში ახალმშენებლობანი, შენობები და სხვა ნაგებობები გრაფის წვერობად შეგვიძლია წარმოვიდგინოთ, ხოლო მათ შორის არსებული გზები, ელექტროგადამცემი ხაზები, წყალგაყვანილობა, კავშირგაბმულობის ხაზები და სხვა კომუნიკაციები – გრაფის წიბოებად. ასეთ გრაფებზე შესაძლებელია ტრანსპორტის ოპტიმალური მარშრუტების დაგეგმვა, ობიექტებს შორის უმოკლესი გზების განსაზღვრა და სხვ.

ორიენტირებული გრაფი



შევისწავლოთ დამოუკიდებლად

მოიძიეთ ინტერნეტში (მაგ., საიტ www.gomap.az-ის დახმარებით) თემის დასაწყისში მოცემულ სქემაზე მითითებულ აზერბაიჯანულ დასახლებულ პუნქტებს შორის არსებული მანძილი. მიღებული მონაცემების საფუძველზე ააგეთ გრაფი. განსაზღვრეთ მასში უმოკლესი მანძილი ალდაშსა და შემახსა შორის.

შეამოწმე შენი ცოდნა

1. რა არის გრაფი და რა ელემენტებისგან შედგება იგი?
2. რა ეწოდება ბმულ გრაფს რკალების გარეშე?
3. რა ჰქვია წიბოს, რომელიც ერთი წვეროდან გამოდის და იმავე წვეროში ბრუნდება?
4. რა ფორმით ინახება გრაფები კომპიუტერის მეხსიერებაში?
5. ააგეთ იმ რეგიონის გრაფი, რომელშიც ცხოვრობთ და მის საფუძველზე შეადგინეთ მოსაზღვრეობის მატრიცა.

20

პრაფუტან დაკავშირებული ამოცანები



- რა ეწოდება მიმართულების მქონე წიბოს?
- რა ეწოდება გრაფს, რომლის ყველა წიბოს აქვს მიმართულება?

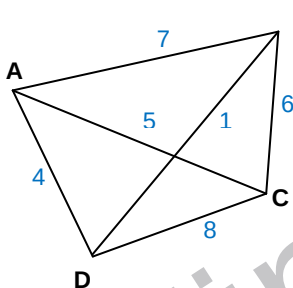
ზოგიერთი პრაქტიკული ამოცანის გადაწყვეტისას (მაგალითად, პუნქტებს შორის უმოკლესი გზის განსაზღვრისას) მნიშვნელობა აქვს არა მარტო წვეროებს შორის კავშირს, არამედ მათ შესაბამის რიცხვებსაც. ეს რიცხვი შეიძლება იყოს, ვთქვათ, მანძილი ქალაქებს შორის ან მგზავრობის ღირებულება. გრაფთა თეორიაში გრაფის თითოეული წიბოს შესაბამის რიცხვით მნიშვნელობას მის **წონას** უწოდებენ, ხოლო თვით ასეთ გრაფს – **შენონილ გრაფს**.

ამოცანა

A, B, C და D პუნქტებიდან თითოეული დანარჩენებს უკავშირდება გზებით, რომელთა სიგრძე ცნობილია: $AB=7$, $AC=5$, $AD=4$, $BC=6$, $BD=1$, $CD=8$. აუცილებელია რომელიმე პუნქტიდან დანყებული, ყველა პუნქტის შემოვლა, თითოეულ მათგანში მხოლოდ ერთხელ მოხვედრით, და ისევ საწყის პუნქტში დაბრუნება. რომელი მარშრუტი უნდა ავირჩიოთ, რომ გზა უმოკლესი აღმოჩნდეს?

ამოხსნა. შესაბამისი პუნქტები და მათ შორის არსებული გზების სქემა შეგვიძლია, შეწონილი გრაფის სახით წარმოვადგინოთ. როგორც ხედავთ, აქ 6 შესაძლებელი ციკლი გვაქვს: ABCDA, ACBDA, ABDCA, ACDBA, ADBCA, ADCBA. მათი სიგრძე, შესაბამისად, არის: 25, 16, 21, 21, 16, 25.

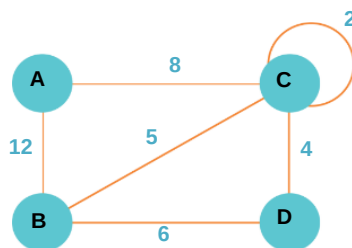
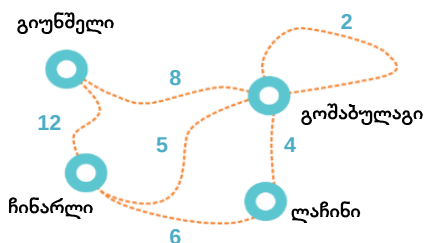
ამგვარად, ყველაზე მოკლე იქნება მარშრუტები ACBDA და ADBCA.



შეწონილ გრაფში მოსაზღვრეობის მატრიცის ნაცვლად **წონით მატრიცას** იყენებენ. წონითი მატრიცის უჯრებში უთითებენ წიბოების წონას და თუ ორ წვეროს შორის წიბო არ არის, შესაბამისი უჯრა ცარიელი რჩება. ნახატზე ნაჩვენებია სქემა, რომელზედაც მითითებულია გზების სიგრძე, მისი შესაბამისი გრაფი და წონითი მატრიცა.

ღამიასკოვა

შეწონილი გრაფი
წონითი მატრიცა



	A	B	C	D
A		12	8	
B	12		5	6
C	8	5		4
D		6	4	

რის განსაზღვრა შეიძლება წონითი მატრიცის დახმარებით? ჯერ ერთი, შეგვიძლია გავიგოთ, არსებობს თუ არა წიბო ორ წვეროს შორის, და თუ არსებობს, რა სიგრძისაა იგი (წონა). ამისათვის საკმარისია, შევხედოთ შესაბამის უჯრას. მაგალითად, B და C წვეროებს შორის არის წიბო და მისი წონა 5-ის ტოლია. მეორეც, თუ ვივარაუდებთ, რომ წიბოების წონა მანძილზე მიუთითებს, შესაძლებელი იქნება გზის სიგრძის განსაზღვრა. მაგალითად, ABCD გზის სიგრძე AB, BC და CD წიბოების სიგრძეთა ჯამის ტოლია: $12+5+4=21$. და ბოლოს, წონითი მატრიცის მეშვეობით შესაძლებელია თვით გრაფის დახაზვა.

ამოცანა

გალაქტიკა ირმის ნახტომში პლანეტა ნეპტუნზე 6 ქალაქი და ისინი თანამიმდევრობითაა დანომრილი, დაწყებული 1-დან. ზოგიერთი ქალაქი ერთმანეთთან გზებითაა დაკავშირებული. გალაქტიკის იმპერატორმა მაქსიმუსმა გადაწყვიტა, შეადგინოს ამ გზების სია პლანეტაზე, მაგრამ რადგანაც თავად მათემატიკაში მოიკოჭლებს, თქვენ გთხოვთ დახმარება.

ამოხსნა. თუ პლანეტა ნეპტუნის ქალაქებს წარმოვადგენთ წვეროების, ხოლო მათ შორის გზებს – წიბოების სახით, მივიღებთ ჩვეულებრივ გრაფს. ამოცანა მოითხოვს წიბოების რაოდენობის პოვნას. მოსაზღვრეობის მატრიცა, რომელიც ამ გრაფს შეესაბამება, დაახლოებით ასე შეიძლება გამოიყურებოდეს:

0	1	1	0	1	0
1	0	1	1	1	0
1	1	0	0	0	0
0	1	0	0	0	1
1	1	0	0	0	1
0	0	0	1	1	0

აქ 1 i სტრიქონისა და j სვეტის გადაკვეთაზე იმაზე მუთითებს, რომ შესაბამის ქალაქებს შორის არსებობს გზა. თუ ასეთი გზა არის, გამოდის, რომ j სვეტისა და i სტრიქონის გადაკვეთაზეც 1 უნდა იყოს. ანუ თუ: ქალაქ i -დან

ქალაქ j-მდე გზა არსებობს, ბუნებრივია, რომ ქალაქ j-დან ქალაქ i-მდეც ასევე არსებობს გზა. მაშასადამე, მოცემული ამოცანის ამოსახსნელად აუცილებელია მოსაზღვრეობის მატრიცაში ერთიანების რაოდენობის დადგენა და შედეგის 2-ზე გაყოფა. ამგვარად, ზემოთ მოცემული მოსაზღვრეობის მატრიცის შესაბამისად, პლანეტა ნეპტუნზე 8 გზაა.

ამ ამოცანის ამოხსნის პროგრამა Python-ის ენაზე ასე შეგვიძლია წარმოვადგინოთ:

```
i = 1
w = 0
while i <= 6:
    s = input()

    w = w + s.count('1')

    i = i + 1
w = w // 2
print(w)
```

გზების რაოდენობა
შეგვყავს მატრიცის
ერთი სტრიქონი.
ითვლება ერთიანების რაოდენობა
ამ სტრიქონში და ეს რიცხვი ემატება
გზების საერთო რიცხვს.

შევისწავლოთ დამოუკიდებლად

გალაქტიკა ირმის ნახტომში პლანეტა ნეპტუნზე N ქალაქია და ისინი თანამიმდევრობითაა დანომრილი, დაწყებული 1-დან. ზოგიერთი ქალაქი ერთმანეთთან გზებითაა დაკავშირებული. გალაქტიკის იმპერატორმა მაქსიმუსმა გადაწყვიტა, შეადგინოს ამ გზების სია პლანეტაზე, მაგრამ რადგანაც თავად მათემატიკაში მოიკოჭლებს, თქვენ გთხოვთ, დაეხმაროთ გზების რაოდენობის დათვლაში (წყარო: informatika.edu.az).

ამოცანა

საჭადრაკო ტურნირი ტარდება წრიული სისტემით, რომლის თანახმად, თითოეული მონაწილე ყოველ მეტოქეს თითოჯერ ხვდება. ტურნირში 7 მოსწავლე მონაწილეობს. ცნობილია, რომ არიფმა ჩაატარა ექვსი პარტია, ბაქირმა – ხუთი, ჯეიჰუნმა და დადაშმა – სამ-სამი, ელჰანმა და ალიმ – ორ-ორი, ხოლო ილკინმა ერთი პარტია ითამაშა. ვისთან ითამაშა ჯეიჰუნმა?

ამოხსნა. ავაგოთ გრაფი G, რომელიც მოთამაშეთა შეხვედრებს ასახავს. ამ გრაფის წვეროები აღვნიშნოთ 1-დან 7-მდე რიცხვებით, ამგვარი შესაბამისობით: 1 – არიფი, 2 – ბაქირი, 3 – ჯეიჰუნი, 4 – დადაში, 5 – ელჰანი, 6 – ალი, 7 – ილკინი.



ვინაიდან წვერო 1-ის ხარისხი 6-ის ტოლია (არიფმა 6 მატჩი ჩაატარა), ეს წვერო ყველა დანარჩენ წვეროსთან უნდა იყოს შეერთებული, ხოლო რადგან წვერო 7-ის ხარისხი 1-ის ტოლია, მისი მოსაზღვრე მხოლოდ წვერო 1 იქნება.

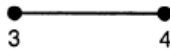
განვიხილოთ ქვეგრავი H_1 , რომელიც მრავალი წვეროსაგან $\{2,3,4,5,6\}$ შედგება. ამ ქვეგრავის მიღება შეიძლება G გრაფიდან ორი წვეროს – 1-ისა და 7-ის – და მათგან გამომავალი წიბოების მოშორებით. ამიტომ ქვეგრავ H_1 -ში, რომელიც 5 წვეროსაგან შედგება, წვეროების ხარისხი ამგვარი იქნება:

$$d(2) = 4, d(3) = d(4) = 2, d(5) = d(6) = 1.$$

გრაფ H_1 -ში, წვერო 2-ს აქვს მოსაზღვრეობა ყველა წვეროსთან, წვერო 5-სა და წვერო 6-ს კი – მხოლოდ ორ წვეროსთან.

ახლა ვნახოთ ქვეგრავი H_2 , რომელიც წვეროების სიმრავლე $\{3,4\}$ -საგან შედგება. ეს გრაფი ქვეგრავ H_1 -დან 2,5, და 6 წვეროებისა და მათგან გამომავალი წიბოების მოშორებით მიიღება.

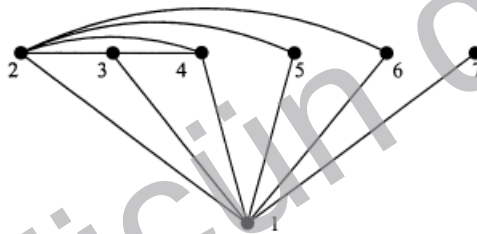
გრაფ H_2 -ში $d(3) = d(4) = 1$, ანუ ეს ამ ნახატზე წარმოდგენილი გრაფია.



თუ დავაბრუნებთ წაშლილ 2, 5 და 6 წვეროებს, მივიღებთ გრაფ H_1 -ს:



ახლა დავაბრუნოთ წაშლილი 1 და 7 წვეროები და მივიღებთ საჭირო გრაფ G -ს:



ეს გრაფი ასახავს მოსწავლეთა შეხვედრებს ტურნირის პროცესში. გრაფიდან ჩანს, რომ ჯეიკუნი (გრაფის მესამე წვერო) ეთამაშა არიფს, ბაქირსა და დადამს, რომლებსაც შეესაბამება წვეროები 1, 2 და 4. ცხადია, რომ ამ გრაფის დახმარებით ადვილი გასაგებია, ვისთან ითამაშეს ასპარეზობის დანარჩენმა მონაწილეებმა.

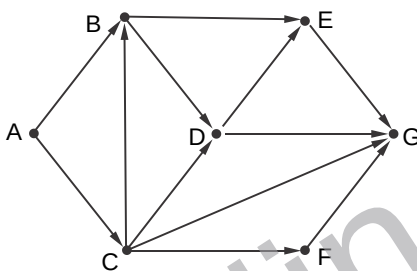
შეამოწმა შენი ცოდნა



1. რა ჰქვია გრაფს, რომლის ყოველ წიბოს გარკვეული რიცხვი შეესაბამება?
2. წარმოიდგინეთ გრაფი, რომლის წონითი მატრიცა არ არის სიმეტრიული მთავარი დიაგონალის მიმართ. თქვენი აზრით, როგორი იქნება იგი: შეწონილი, ორიენტირებული თუ ბმული? რა თვისებები ექნება მას, ექნება ციკლი თუ არა?
3. რამდენი წიბოა მოცემულ წონით მატრიცაში? რას უდრის A და E წვეროების შემადგენელი წიბოს წონა?

	A	B	C	D	E
A		5	2		6
B	5			5	
C	2			2	
D		5	2		3
E	6			3	

4. თუ წონით მატრიცაში რიცხვები პუნქტებს შორის დაშორებას გვიჩვენებს, რის ტოლი იქნება A-B-D-E გზის სიგრძე?
5. ნახატზე მოცემულია ABCDEFG ქალაქების დამაკავშირებელი გზების სქემა. თითოეულ გზაზე მხოლოდ მითითებული მიმართულებით შეიძლება გადაადგილება. რამდენი მარშრუტია A ქალაქიდან G ქალაქამდე?





- რა არის კომპიუტერული ქსელი?
- რას ნიშნავს ლოკალური ქსელი?

საქმიანობა

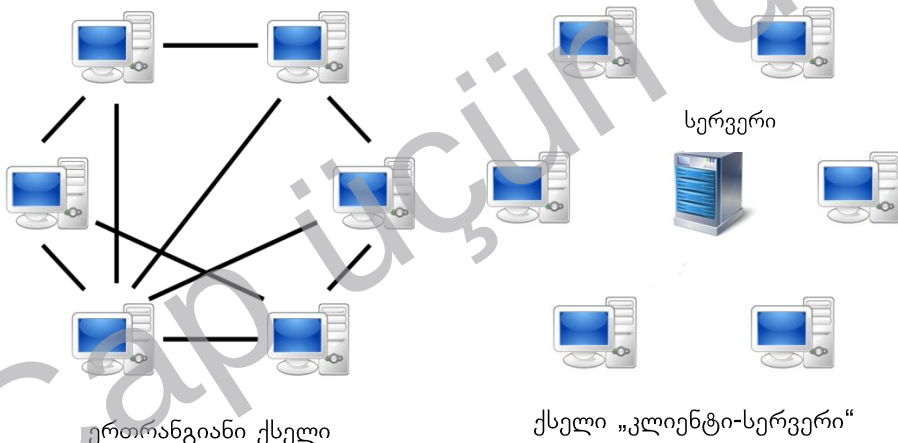
ააგეთ თქვენი სკოლის კომპიუტერული ქსელის შესაბამისი გრაფი. ამისათვის თითოეული კომპიუტერი გრაფის წვეროდ ჩათვალით. თუ ქსელი არ არის, მაშინ წარმოიდგინეთ, რომ კლასში ყოველი მოსწავლის წინ დგას კომპიუტერი, რომელიც დანარჩენ კომპიუტერებთან კაბელის მეშვეობითაა დაკავშირებული.

ვიმსჯელოთ:

- როგორი გრაფი დახაზეთ კომპიუტერული ქსელისთვის – ჩვეულებრივი თუ ორიენტირებული? პასუხი დაასაბუთეთ.
- თუ გრაფიდან რომელიმე წიბოს მოაშორებთ (ანუ განწყვეტთ კავშირს ორ კომპიუტერს შორის), როგორ აისახება ეს მთელი კომპიუტერული ქსელის მუშაობაზე?

როგორც იცით, კომპიუტერული ქსელი შეიძლება რამდენიმე კომპიუტერისგან შედგებოდეს ან მილიონობით კომპიუტერს აერთიანებდეს. ტერიტორიული პრინციპის მიხედვით, კომპიუტერული ქსელები რამდენიმე სახედ იყოფა, რომელთა შორის ყველაზე მეტად გავრცელებულია **ლოკალური** და **გლობალური** ქსელები. როგორც წესი, სპეციალისტები მათ შემოკლებული სახელებით, შესაბამისად, LAN-ით (lokal Area Networks) და WAN-ით (Wide Area Networks) მოიხსენიებენ. როგორც სახელწოდებებიდან ჩანს, ლოკალური ქსელები მოიცავს შეზღუდულ, ხოლო გლობალური ქსელები – უფრო ფართო გეოგრაფიულ ტერიტორიებს.

მიუხედავად იმისა, რომ სხვადასხვა ტიპის ლოკალური ქსელები არსებობს, ძირითადად გამოიყენება ორი მათგანი: ქსელი „კლიენტი-სერვერი“ და ერთრანგიანი ქსელი.



ქსელი „კლიენტი-სერვერი“ (client-server network).

ამ ტიპის ლოკალურ ქსელში მომუშავე სადგურებისა და სერვერების შესაძლებლობების მაქსიმალურად გამოყენების მიზნით მართვა მათ შორისაა განაწილებული. ამ არქიტექტურაში გამოყენებითი პროგრამების დამუშავება განაწილებულია კლიენტ-კომპიუტერსა და კომპიუტერ-სერვერს შორის. როგორც წესი, სერვერად გამოყენებულია უფრო მძლავრი კომპიუტერი. სერვერი კლიენტს მრავალმომხმარებლიან გარემოში მუშაობის ტრადიციულ უპირატესობებს ანიჭებს. ეს არის მონაცემების მართვა, ინფორმაციაზე კოლექტიური მუშაობა, მონაცემებისა და მათი დაცვის შესაძლებლობის ქსელური ადმინისტრირება.

დაიხასოვრა

ლოკალური ქსელი
გლობალური ქსელი
ქსელი „კლიენტი-სერვერი“
ერთრანგიანი ქსელი
ტოპოლოგია
ტოპოლოგია „სალტე“
ტოპოლოგია „რგოლი“
ტოპოლოგია „ვარსკვლავი“

ერთრანგიანი ქსელი (peer to peer networks). ამგვარი არქიტექტურის ქსელში ყველა კომპიუტერი ერთმანეთის თანასწორია. აქ არც ერთი კომპიუტერი არ თამაშობს სერვერის როლს. ქსელის მომხმარებლები თავად განსაზღვრავენ, კომპიუტერში არსებულ რომელ ფაილებს გაუზიარებენ სხვა მომხმარებლებს. ასეთი ქსელის მონყობა და მისი მართვა ძალიან ადვილია. მცირე ოფისები, როგორც წესი, უპირატესობას ერთრანგიან ქსელებს ანიჭებენ.

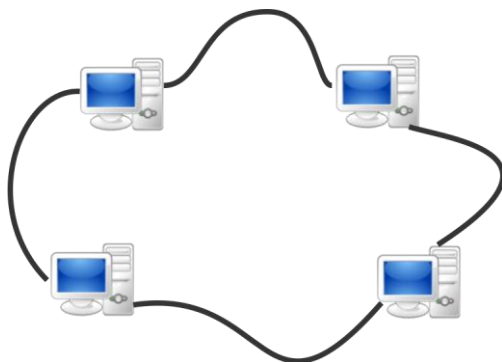
ქსელის შექმნისას დიდი მნიშვნელობა ენიჭება **ტოპოლოგიას**, ანუ ქსელური მოწყობილობებისა და კაბელების განლაგებისა და შეერთების სქემას. აუცილებელია ისეთი სტრუქტურის (გრაფის) არჩევა, რომ ქსელი საიმედოდ და ეფექტიანად მუშაობდეს და შესაძლებელი იყოს ინფორმაციის ნაკადის ადვილად მართვა. ქსელების უმრავლესობა სამი ტოპოლოგიის საფუძველზე იგება:

1. **ტოპოლოგია „სალტე“.** ამ ტოპოლოგიაში ლოკალური ქსელის ყველა კომპიუტერი ერთ საერთო კაბელთანაა მიერთებული. ამგვარ ქსელში მონაცემები ქსელის ყველა კომპიუტერს გადაეცემა. თითოეული კომპიუტერი ამოწმებს, მისთვისაა თუ არა განკუთვნილი ინფორმაცია. თუ ინფორმაცია მართლაც მისთვისაა განკუთვნილი, ის მას მიიღებს და დაამუშავებს. ამ ტოპოლოგიას სხვანაირად **ხაზოვან ტოპოლოგიასაც** უწოდებენ. ის ძალიან მარტივია და იაფიც ჯდება (ცოტა კაბელი სჭირდება), მაგრამ ზოგიერთი ნაკლოვანებით ხასიათდება. ამ ქსელის რომელიმე ნაწილის გაუმართაობა მთელი ქსელის მუშაობის შეფერხებას იწვევს. სწორედ ამ მიზეზის გამო წარსულში ძალზე პოპულარული ტიპოლოგია „სალტე“ დღეს პრაქტიკულად, აღარ გამოიყენება.

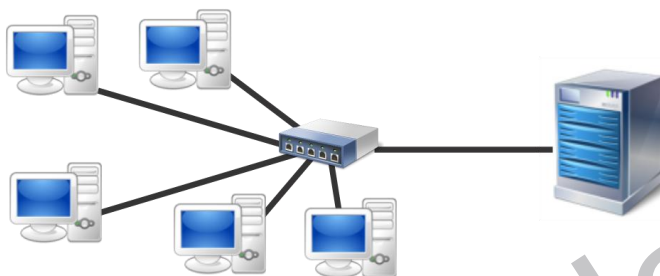


2. **ტოპოლოგია „რგოლი“.** ამ ტოპოლოგიის შემთხვევაში ყოველი კომპიუტერი ორ სხვა კომპიუტერთანაა შეერთებული, იღებს ინფორმაციას ერთიგან და გადასცემს მას მეორეს. ბოლო კომპიუტერი პირველთანაა შეერთებული და ასე იკვრება წრე. ამ ტოპოლოგიასაც უპირატესობაზე მეტი ნაკლი აქვს, მაგალითად, ძნელი დასადგენია, სად არის ლოკალიზებული გაუმართაობა. ერთი კომპიუტერის მწყობრიდან გამოსვლა გავლენას ახდენს მასთან

შეერთებული კომპიუტერების და, საბოლოოდ, მთელი ქსელი მუშაობაზე. ამიტომ დღესდღეობით რგოლურ ქსელსაც იშვიათად ვხვდებით.

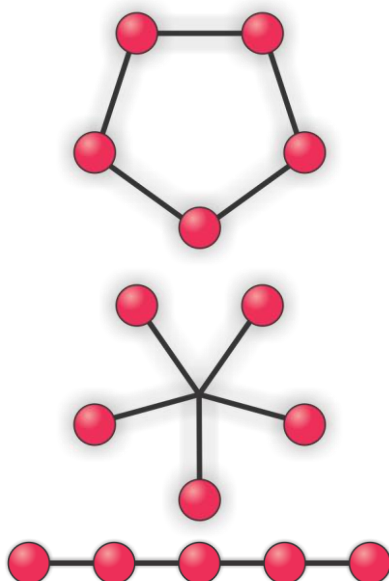


3. **ტოპოლოგია „ვარსკვლავი“.** ამ ტოპოლოგიაში, რომელიც გამოთვლითი ტექნიკის განვითარების საწყის ეტაპებზევე გაჩნდა, ყველა კომპიუტერი მძლავრ ცენტრალურ კომპიუტერთანაა მიერთებული. ასეთი კონფიგურაციისას ყველა მონაცემი მხოლოდ ცენტრალური კომპიუტერიდან გადაეცემა და ქსელის ყველა ნაწილს შორის ინფორმაციის გაცვლაზეც სწორედ ის აგებს პასუხს.



შავისნავლოთ დამოუკიდებლად

განსაზღვრეთ, რომელ ქსელურ ტოპოლოგიას შეესაბამება მოცემული გრაფები.



ამ გრაფებიდან რომელს ჰგავს გრაფი, რომელიც თქვენ ბლოკ „საქმიანობაში“ ააგეთ? მოცემული გრაფებისთვის შექმენით წონითი მატრიცა (თითოეული წიბოს წონად მისი სიგრძე ჩათვალით).

შეამოწმა შენი ცოდნა

1. ტერიტორიული პრინციპის მიხედვით, რა სახეობად იყოფა კომპიუტერული ქსელები?
2. საიდან მომდინარეობს სახელწოდებები: „კლიენტი-სერვერი“ და „ერთრანგიანი ქსელი“?
3. რა არის ქსელის ტოპოლოგია და რა სახის ტოპოლოგიებს იყენებენ კომპიუტერულ ქსელებში?
4. თუ თქვენს სკოლაში არის კომპიუტერული ქსელი, განსაზღვრეთ, რა ტოპოლოგიითაა იგი მოწყობილი?
5. არის თუ არა ტოპოლოგიის ყველა სახე გრაფი?



- როგორ წარმოგიდგენიათ გზა, რომელსაც გაივლის ინფორმაცია ჩვეულებრივი და მობილური ტელეფონებით ლაპარაკის დროს?
- როგორ გადაეცემა ინფორმაცია ინტერნეტის საშუალებით?

საქმიანობა

დაალაგეთ ნახატები ქრონოლოგიური თანამიმდევრობით. აღწერეთ ინფორმაციის გადაცემის საშუალება თითოეული ნახატის მიხედვით.



ვიმსჯელოთ:

- რა სახის ინფორმაცია გადაიცემა თითოეული ამ საშუალების მეშვეობით?
- როგორ ფიქრობთ, ინფორმაციის გადაცემის კიდევ რომელმა საშუალებებმა მოახდინა მნიშვნელოვანი ზეგავლენა საზოგადოების განვითარებაზე?

ინფორმაციის გაცემის საშუალებები რომ არა, საზოგადოება ვერ განვითარდებოდა. დაგროვილი ინფორმაციისა და შეძენილი ცოდნის გავრცელება და თაობიდან თაობისთვის გადაცემა ადამიანთა საზოგადოების განვითარების უმნიშვნელოვანესი ფაქტორია.

ინფორმაციის გადაცემასთან დაკავშირებული ნებისმიერი პროცესი შემდეგი მარტივი სქემის დახმარებით შეიძლება გადმოვიცეს:

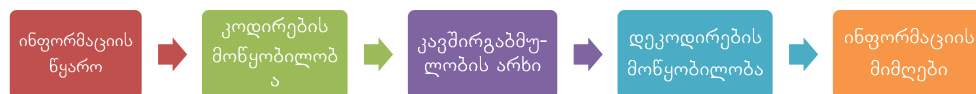
ინფორმაციის წყარო

კავშირგაბ-
მულობის
არხი

ინფორმაციის მიმღები

ინფორმაციის წყაროდან გამოსული ინფორმაცია **კავშირგაბმულობის არხის** გავლით აღწევს **ინფორმაციის მიმღებამდე**. ამ სქემის თითოეულ ნაწილს მისთვის დამახასიათებელი თვისებები, ანუ პარამეტრები, გააჩნია და გადაცემის ხარისხი ამ პარამეტრებზეა დამოკიდებული.

ზემოთ მოცემული სქემა ძალზე გამარტივებულია. სინამდვილეში ეს პროცესი შემდეგნაირად მიმდინარეობს:



ინფორმაციის წყარო შეიძლება იყოს როგორც ცოცხალი არსება, ისე ტექნიკური მოწყობილობა (მაგალითად, კომპიუტერი). წყაროდან ინფორმაცია **კოდირების** მოწყობილობას (მაგალითად, მოდემს) გადაეცემა. ეს მოწყობილობა მიღებულ ინფორმაციას იმ ფორმად გარდაქმნის, რომელიც მოსახერხებელია გადასაცემად. კავშირგაბმულობის არხში (მაგალითად, სატელეფონო ხაზში) გავლის შემდეგ ინფორმაცია **დეკოდირების** მოწყობილობაში (მაგალითად, მოდემში) ხვდება, რომელიც მას მიმღებისთვის გასაგებ ფორმად გარდაქმნის.

ინფორმაციის ერთი კომპიუტერიდან მეორეში გადასაცემად აუცილებელია დამაკავშირებელი რგოლის – **არხის** არსებობა. ამგვარი არხის მაგალითად შეიძლება მოვიყვანოთ სატელეფონო ქსელი, რომელიც დღესდღეობით ჩვენს ცხოვრებაში დიდ როლს თამაშობს. სატელეფონო სისტემა აერთიანებს ყველა სატელეფონო ქსელს – ლოკალურს, დაშორებულს, მობილურს, თანამგზავრულს. მართალია, სატელეფონო ქსელი ხმოვანი კავშირისთვისაა განკუთვნილი, მაგრამ ამჟამად მას სხვა მონაცემების (მაგ., ფაქსის, ელექტრონული ფოსტის, ცოცხალი აუდიოს და ვიდეოს) გადასაცემადაც ფართოდ იყენებენ.

არხებით მონაცემების გადასაცემად აუცილებელია კავშირგაბმულობის ან ფიზიკური, ან უსადენო საშუალებები.

ფიზიკური საშუალებები. მონაცემების გადასაცემად რამდენიმე სახის ფიზიკური საშუალება არსებობს.

კაბელი „გრეხილი წვეილი“. ეს კაბელი ყველაზე იაფია და სატელეფონო სისტემების უმრავლესობაში გამოიყენება. იგი ორი იზოლირებული და ერთმანეთზე გადაგრეხილი სადენისგან შედგება. ერთ-ერთ სადენს მოაქვს სიგნალი, მეორე კი დამინებულია. დამინებული სადენი (მისი ბოლო მიწაშია ჩაფლული) შთანთქავს რადიოდაბრკოლებებს, რითაც იცავს მეორე სადენით გადაცემულ სიგნალს.

დამატებითი

ფიზიკური საშუალებები უსადენო საშუალებები კაბელი „გრეხილი წვეილი“ კოაქსიალური კაბელი ფიბროპტიკური კაბელი რადიოსიგნალები მიკროტალღები თანამგზავრული კავშირგაბმულობის საშუალებები

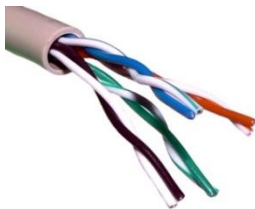
კოაქსიალური კაბელები. კაბელის ამ სახეობას, რომლითაც ძირითადად სატელევიზიო სფეროში სარგებლობენ, კომპიუტერულ ქსელებშიც ფართოდ იყენებენ. კოაქსიალური კაბელი შედგება ორი სადენისგან:

ცილინდრული ფორმის ეკრანისა და მასში მოთავსებული ცენტრალური იზოლირებული სადენისგან.

ამ იზოლაციის წყალობით, კოაქსიალურ კაბელში გამავალ სიგნალებს არ ემუქრება დაბრკოლებები, რომლებსაც სხვა გადაცემები და ელექტრონული მოწყობილობები წარმოქმნის. ამ უპირატესობის მიუხედავად, ეს კაბელი უფრო ძვირია, ვიდრე გრეხილი წყვილი.

ფიბროოპტიკური კაბელი. ფიბროოპტიკური (ბოჭკოვან-ოპტიკური) კაბელი თხელი ელასტიკური მინის მილაკებისგან მზადდება. ეს კაბელი კავშირგაბმულობის ტრადიციულ მეტალის ხაზებთან შედარებით გარკვეულ უპირატესობას ფლობს. სიგნალების გატარების ფართო არხის წყალობით მას შეუძლია ბევრად მეტი მონაცემის გადაცემა გაცილებით იოლად, ვიდრე მეტალის სადენებს. ამავე დროს ის ნაკლებად ექვემდებარება დაბრკოლებებს. ფიბროოპტიკური კაბელის ძირითადი ნაკლი მისი სიმყიფე და სიძვირეა.

მონაცემების გადაცემის ფიზიკური საშუალებები



კაბელი „გრეხილი წყვილი“



კოაქსიალური კაბელი



ფიბროოპტიკური კაბელი

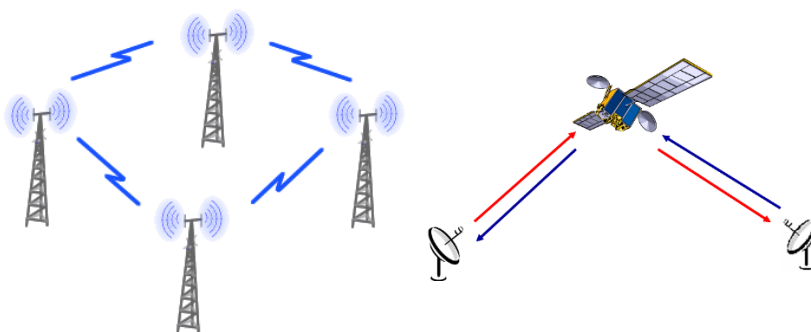
ამჟამად, ფიზიკურ საშუალებებთან ერთად, ფართოდ გამოიყენება **კავშირგაბმულობის უსადენო საშუალებებიც.**

- **რადიოსიგნალები.** რადიოსიგნალების დახმარებით მონაცემების გადაცემის დროს გადაცემის ანტენებსა და მიმღებს შორის პირდაპირი ხილვადი კავშირი უნდა იყოს, ანუ სიგნალი ერთი პუნქტიდან მეორეში პირდაპირი ხაზით უნდა მიდიოდეს. რადიოკავშირისთვის საჭიროა რადიოგადამცემი, რომელიც სიგნალებს გზავნის, და რადიომიმღები, რომელიც ამ სიგნალებს იჭერს.
- **მიკროტალღები.** მიკროტალღოვანი სიგნალები სივრცეში ელექტრომაგნიტური ტალღების სახით ვრცელდება. რადიოსიგნალების მსგავსად, ისინიც პირდაპირი ხაზით გადაეცემა ერთი მიკროტალღოვანი სადგურიდან მეორეს. დაბრკოლებების თავიდან ასაცილებლად მიკროტალღოვანი სადგურების უმრავლესობა შემალღებულ ადგილებზე ან მაღალსართულიან შენობებზე იდგმება.
- **თანამგზავრული კავშირგაბმულობის საშუალებები.** თანამგზავრული ანტენები დედამიწის ორიბიტაზე, მისი ზედაპირიდან ათეულობით ათასი კილომეტრის დაშორებით მდებარეობს. ასეთი სიმაღლის წყალობით თანამგზავრები დედამიწის ბრუნვის სიჩქარით ბრუნავს და

მისი ზედაპირის მიმართ ერთ წერტილში სტაბილურ მდებარეობას ინარჩუნებს. თანამგზავრები აღჭურვილია მონყობილობებით, რომლებიც აძლიერებს დედამიწიდან გაგზავნულ სიგნალებს და კვლავ უბრუნებს მათ დედამიწას.

- ქსელის კონსტრუირებისა და კომუნიკაციის ერთ-ერთი საშუალების არჩევისას აუცილებლად უნდა იქნეს გათვალისწინებული რამდენიმე ფაქტორი: ქსელის სახე, მისი ზომები (მასშტაბი) და ღირებულება.

მონაცემთა გადაცემის უსადენო საშუალებები



მიკროტალღოვანი კოშკურები თანამგზავრული კავშირგაბმულობა

შევისწავლოთ დამოუკიდებლად

შეკრიბეთ ინტერნეტიდან ინფორმაცია Wi-Fi და Wi-Max ტექნოლოგიების შესახებ. როგორია ამ ტექნოლოგიებით ინფორმაციის გადაცემის მაქსიმალური სიჩქარე? რა უპირატესობები და ნაკლოვანებები გააჩნია მათ?

შეამოწმა შენი ცოდნა ?

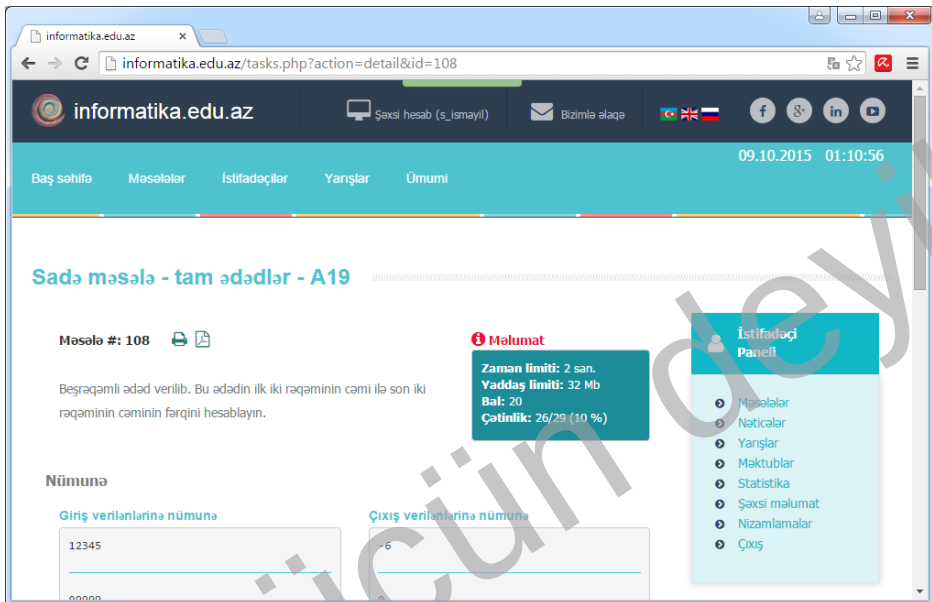
1. როგორია მონაცემთა გადაცემის ზოგადი სქემა?
2. რა არის კავშირგაბმულობის არხები და როგორ კლასიფიცირდება ისინი?
3. კავშირგაბმულობის რომელ უსადენო საშუალებებს იყენებენ მონაცემთა გადასაცემად?
4. კაბელის რომელი სახეობაა გამოყენებული თქვენი სკოლის კომპიუტერულ ქსელში?
5. როგორ ფიქრობთ, რომელი სიგნალების დახმარებით გადაიცემა ინფორმაცია მობილურ ტელეფონებში?



- რა არის ვებგვერდი და რით განსხვავდება იგი ვებსაიტისაგან?
- რას უწოდებენ ვებგვერდის სადემონსტრაციო პროგრამებს?

საქმიანობა

1. გახსენით ნებისმიერი ვებბრაუზერი;
2. გადადით საიტ *informatika.edu.az*-ზე;
3. რომელიმე ენაზე გადასასვლელად აირჩიეთ საიტის ზედა ნაწილში არსებული პიქტოგრამა შესაბამისი დროებით;
4. მენიუ ზომბალ-ში აირჩიეთ პუნქტი „პროექტის შესახებ“ და გაეცანით საიტის დანიშნულებას;
5. მენიუ ამოცანებ-ში აირჩიეთ პუნქტი ამოცანების საერთო სია. გაეცანით რამდენიმე ამოცანის პირობას. აირჩიეთ პუნქტი სტატისტიკა. გულდასმით დაათვალიერეთ დიაგრამები;
6. დაანკაპუნეთ სიაში პირველი მომხმარებლის სახელზე და გახსნილ გვერდზე გაეცანით მის შედეგებს.



ვიმსჯელოთ:

- პროგრამირების რომელი ენაა უფრო პოპულარული საიტის მომხმარებელთა შორის? მომხმარებელთა რა ნაწილი მიმართავს პროგრამირებისას Python-ის ენას?
- რამდენმა ადამიანმა ამოხსნა სწორად A50 მარტივი ამოცანა – მთელი რიცხვები?

კომპიუტერის ინტერნეტში შესვლა ხშირად სატელეფონო კავშირის საშუალებით ხორციელდება. სამწუხაროდ, სატელეფონო ქსელით ინფორმაციის გადაცემის სიჩქარე მეტად დაბალია. ამიტომ ინტერნეტში დროის ერთეულში გადაცემული სიმბოლოების რაოდენობის, ანუ ტრაფიკის, შემცირების მიზნით, თავად დოკუმენტს კი არა, სპეციალურ ენაზე დაწერილ მის აღწერილობას აგზავნიან. ვებდოკუმენტის აღსაწერად ენა **HTML-ს** (HyperText Markup Language, ჰიპერტექსტის მარკირების ენა) იყენებენ (იკითხება „ეიჩ-თი-ემ-ელ“). ამ ენაზე აღწერილობის მიღების შემდეგ ბრაუზერი ქმნის საწყის დოკუმენტს – აყალიბებს ტექსტს და სათანადო ადგილებზე განათავსებს ილუსტრაციებს. ყოველ ჯერზე, როცა კი ამა თუ იმ ვებგვერდს ვხსნით, სინამდვილეში **HTML-ის** ენაზე დაწერილ დოკუმენტს ვხედავთ. ყველა ვებდოკუმენტი **HTML-ის** ენის დახმარებითაა დაფორმატებული.

იმისათვის, რომ ბრაუზერს ან სხვა პროგრამას ვაჩვენოთ, როგორ გამოიტანოს ინფორმაცია ეკრანზე **HTML-ის** ენაზე, ტეგების ნაკრებს ვიყენებთ. **ტეგები** არის ბრძანებები, რომლებიც ბრაუზერში ტექსტის ასახვის წესებს განსაზღვრავს. ტეგი ყოველთვის გახსნილი კუთხური ფრჩხილით (ნიშანი <) იწყება და დახურული კუთხური ფრჩხილით (ნიშანი >) მთავრდება. ტეგების მეშვეობით დოკუმენტში შესაძლებელია განყოფილებების, აბზაცების, სიების, ნახატების, ცხრილების, კოლონ-ტიტულებისა და სხვა ობიექტების მითითება, თითოეული ბლოკის ფარგლებში შრიფტის, ზომების, სიმბოლოების ფერის შეცვლა, კურსივით ან მუქი შრიფტით გამოყოფა. **HTML-ის** ენის შემქმნელად **ტიმ ბერნერს-ლი** (Tim Berners-Lee) ითვლება.

დამატებით

ვებსაიტი
ვებგვერდი
HTML
ტეგი
მთავარი გვერდი



ტიმ
ბერნერს-ლი
(1955)

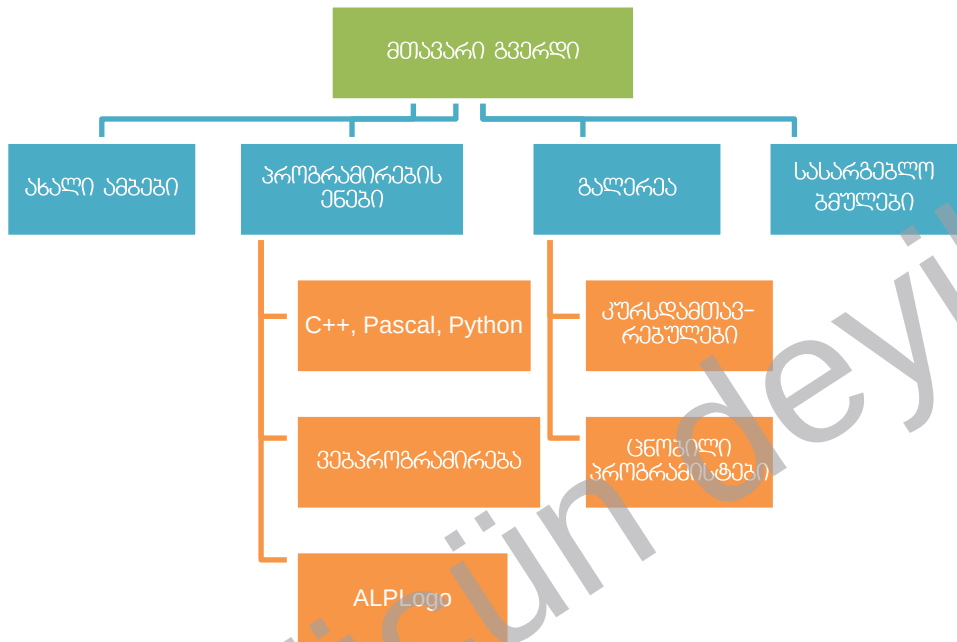
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Sample HTML Page</title>
  <meta http-equiv="Content-type" content="text
  <meta property="og:type" content="website" />
  <meta property="og:url" content="http://www.s
  <meta name="robots" content="index, follow" /
  <meta name="author" content="http://www.somed
  <link href="http://www.somedomain.com/" rel="
  <link href="http://www.somedomain.com/" rel="
```


ვებგვერდების ან ვებსაიტების შესაქმნელად რამდენიმე მეთოდის გამოყენება შეიძლება:

1. HTML დოკუმენტის შექმნა მარტივ ტექსტურ რედაქტორში (მაგალითად, Notepad-ში ან Wordpad-ში.
2. დოკუმენტის შექმნა ტექსტურ პროცესორში (მაგალითად, Microsoft Word-ში ან OpenOffice.org Writer-ში) და ვებგვერდის HTML-ის ფორმატით შენახვა.
3. სპეციალური პროგრამებით – ვიზუალური HTML-რედაქტორით (მაგალითად, Adobe Dreamweaver, Microsoft FrontPage, Nvu) ან **საიტების კონსტრუქტორით** (მაგალითად, uCoz-ით) სარგებლობა;

რა მეთოდიც არ უნდა იქნეს არჩეული ვებსაიტის შექმნისას, პირველ რიგში საჭიროა მისი **შინაარსისა და სტრუქტურის** განსაზღვრა.

დავუშვათ, რომ თქვენ „ნორჩი პროგრამისტის“ წრის წევრი ხართ და ამ წრის საქმიანობის ამსახველი საიტის სტრუქტურის გრაფის ფორმით წარმოდგენა შესაძლებლობას მოგცემთ, თვალნათლივ წარმოაჩინოთ მისი შინაარსი. გარდა ამისა, ის საიტის ერთი გვერდიდან მეორეზე გადასვლის ორგანიზებაშიც დაგეხმარებათ.



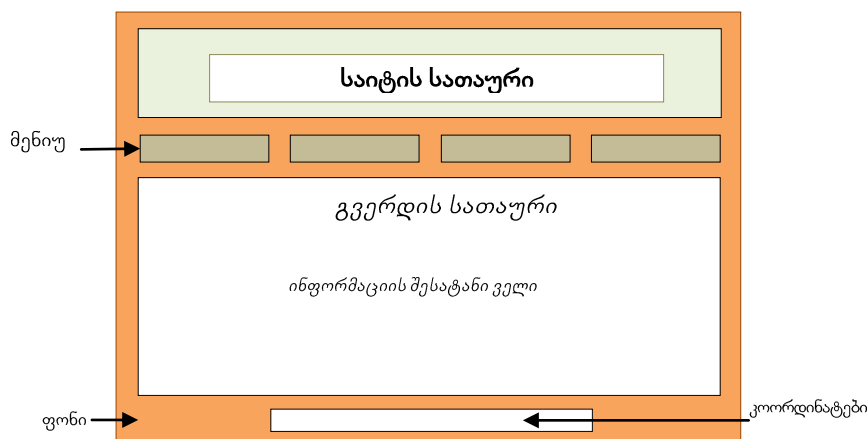
ნებისმიერი საიტის ჩატვირთვის დროს თავიდან მისი **მთავარი (საშინაო) გვერდი (home page)** იხსნება. ეს გვერდი, პირველ რიგში, პასუხობს კითხვაზე: „რას შეეხება ეს საიტი?“ ამისათვის მთავარ გვერდზე განათავსებენ საიტის დასახელებას, მისი თემის შესატყვის გრაფიკულ გამოსახულებას, საიტის შინაარსის ამსახველ მოკლე ტექსტს და ასევე მთავარ მენიუს, რომელიც საიტის ძირითად ნაწილებზე გადასვლას უზრუნველყოფს. მთავარ გვერდზე შეიძლება

განთავსდეს აგრეთვე მისი შემქმნელის სახელი, საკონტაქტო ინფორმაცია მასთან დასაკავშირებლად, ახალი ამბები და სხვა საინფორმაციო ბლოკები.

საიტის ყოველ გვერდზე, როგორც წესი, რამდენიმე მუდმივი ელემენტია, რომლებიც ერთსა და იმავე ადგილებზე მდებარეობს:

- საიტის სათაური, რომელიც გვერდის ყველაზე ზედა ნაწილში თავსდება;
- მთავარი მენიუ, რომელიც, ჩვეულებრივ, გვერდის ზედა ან მარცხენა ნაწილში მდებარეობს (მენიუს პუნქტების რაოდენობა, მათი სახელწოდებები და თანამიმდევრობა საიტის ყველა გვერდზე უცვლელია).

მიზანშეწონილია საიტის ყველა გვერდის ერთი და იმავე სტილით დამუშავება. სტილის შესანარჩუნებლად სასურველია საიტის ერთი გვერდის შაბლონის შექმნა და მასზე ყველა გვერდისთვის სავალდებულო ელემენტების ასახვა.



შეისწავლოთ დამოუკიდებლად

შეისწავლეთ საიტ informatika.edu.az-ის სტრუქტურა და წარმოადგინეთ იგი გრაფის ფორმით. რატომ არის საიტის სტრუქტურისთვის უფრო მისაღები არა ხისებრი, არამედ გრაფის ფორმის ინფორმაციული მოდელი?

შეამოწმა შენი ცოდნა ?

1. რატომ გამოიყენება სპეციალური ენა ვებდოკუმენტების აღსაწერად?
2. რა არის ტეგი და როგორ წარმოადგენს მას HTML-ის დოკუმენტში?
3. რა საშუალებებით ქმნიან ვებსაიტებს?
4. რას წარმოადგენს „მთავარი გვერდი“ და რა არის მასზე განთავსებული?

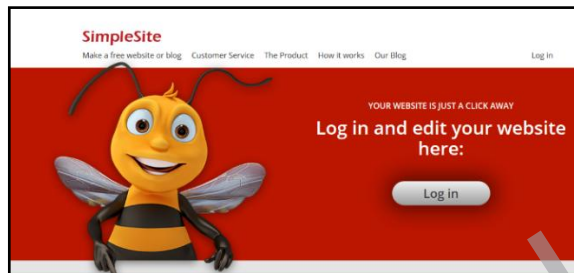
წინა გაკვეთილზე თქვენ შეიტყვეთ, რომ ვებსაიტების შექმნის მრავალი სხვადასხვა საშუალება არსებობს. ერთ-ერთი მათგანია იმ მომსახურებით სარგებლობა, რომელსაც ინტერნეტის გარკვეული საიტები გვთავაზობს. მომხმარებლებს, რომლებმაც HTML-ის ენა არ იციან და არც საიტების შექმნის გამოცდილება გააჩნიათ, ეს საიტები საშუალებას აძლევს, მოკლე დროში და ყოველგვარი ფინანსური დანახარჯების გარეშე შექმნან თავიანთი საკუთარი საიტები.

ახლა გაიგებთ, როგორ შექმნათ ერთ-ერთ ასეთ საიტზე არსებული მზა შაბლონების დახმარებით საინფორმაციო საზოგადოებისადმი მიძღვნილი საიტი. ამისათვის ჯერ გაეცანით ბლოკ „საქმიანობის“ შემდეგ მოცემულ მასალებს და განსაზღვრეთ თქვენ მიერ შესაქმნელი საიტის სტრუქტურა.

საქმიანობა

ვებსაიტის შექმნა შაბლონების საფუძველზე

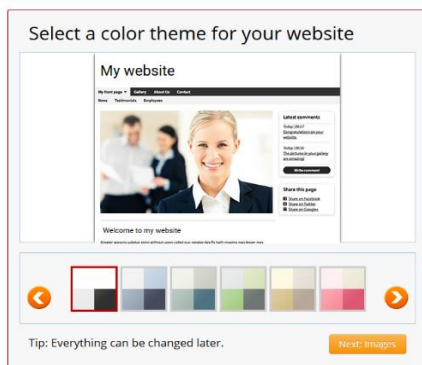
1. ჩაწერეთ ბრაუზერის მისამართების სტრიქონში მისამართი www.simplesite.com და დააჭირეთ კლავიმ <enter>-ს. გაიხსნება საიტის მთავარი ფანჯარა;



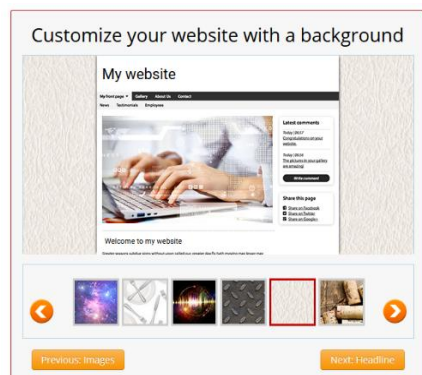
2. მენიუში აირჩიეთ პუნქტი Make a free website or blog. გაიხსნება შემდეგი გვერდი.



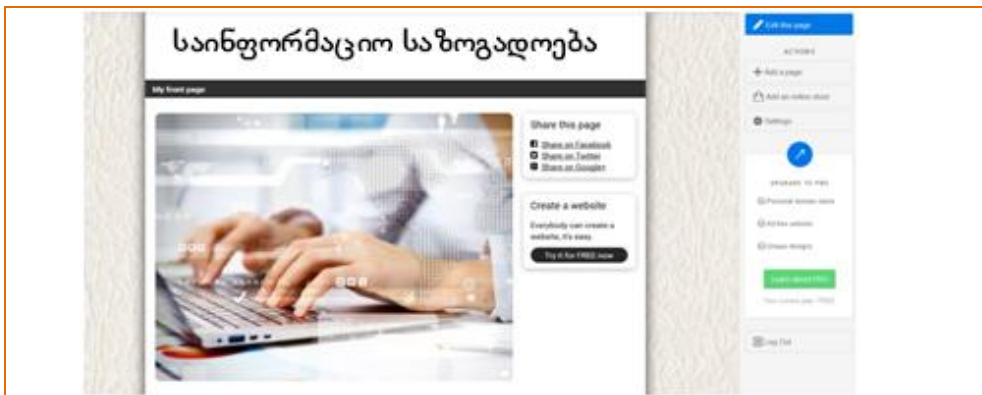
3. დაანკაპუნეთ ლილაკ Personal website or blog-ზე;
4. გახსნილ გვერდზე აირჩიეთ ფერების ცხრილი და დაანკაპუნეთ ლილაკ Next: Images-ზე;



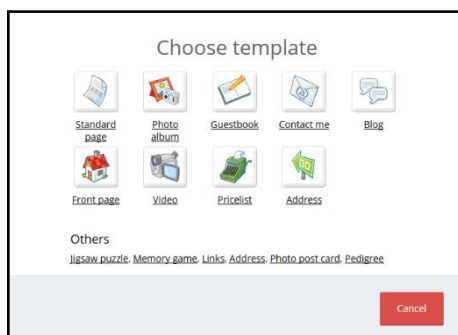
5. პროგრამა შემოგთავაზებთ სხვადასხვა გამოსახულებას მთავარი გვერდისთვის. აირჩიეთ ის, რომელიც თქვენს თემას შეესაბამება და დააწკაპუნეთ ლილაკ Next: Backgrounds-ზე;
6. შემოთავაზებულთაგან ერთი გამოსახულება აირჩიეთ საიტის გვერდის ფონად;

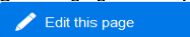


7. დააწკაპუნეთ ლილაკ Next: Headline-ზე. გახსნილ ახალ გვერდზე აკრიფეთ საიტის სათაური, მაგალითად, **საინფორმაციო საზოგადოება**.
8. დააწკაპუნეთ ლილაკ Save and continue-ზე;
9. გახსნილ გვერდზე საიტისთვის სახელის დასარქმევად მითითებულ ველში შეიყვანეთ შესაბამისი სათაური (მაგალითად, **info-sazogadoeba**). თუ ასეთი სახელი ვინმეს მიერ უკვე გამოყენებულია, გაჩნდება შეტყობინება ამის შესახებ;
10. საიტისთვის სახელის დარქმევის შემდეგ გამოვა შეტყობინება საიტის წარმატებით შექმნის თაობაზე;
11. მომდევნო გვერდზე შეიტანეთ პაროლი საიტის ადმინისტრირებისათვის;
12. მიუთითეთ ელექტრონული ფოსტის მისამართი და დააწკაპუნეთ შესაბამის ლილაკზე;
13. თუ ყველა ნაბიჯი სწორად შეასრულეთ, ბრაუზერი გამოიტანს თქვენი საიტის მთავარ გვერდს;
14. საიტის მისამართის დასადასტურებლად გაეცანით თქვენი ელექტრონული ფოსტით მიღებულ წერილს და გადადით მითითებულ ბმულზე. თქვენი საიტის მისამართში იქნება სიტყვა **simplesite** და ის სახელი, რომელიც თავად აირჩიეთ: **info-sazogadoeba.simplesite.com**.



15. საიტზე თემის შესატყვისი მასალების განსათავსებლად შეგიძლიათ, დამატებითი გვერდები შექმნათ. ამისათვის მარჯვნივ მდებარე პანელზე დააწკაპუნეთ ლილვს Add a page-ზე. გაიხსნება სხვადასხვა ტიპის გვერდების ნიმუშების შემცველი ფანჯარა. აირჩიეთ ვარიანტი Standard page;



16. გვერდზე მასალების განთავსებისა და მათი რედაქტირების მიზნით დააწკაპუნეთ ლილვს -ზე, რომელიც მარჯვენა ზედა კუთხეში მდებარეობს. გაეცანით გახსნილ ფანჯარაში ნაჩვენებ რედაქტორის ინსტრუმენტებს და გაარკვიეთ თითოეული ინსტრუმენტის ფუნქცია;
17. განათავსეთ გვერდზე ინფორმაცია საინფორმაციო საზოგადოების შესახებ;
18. შეკრიბეთ ინტერნეტში ინფორმაცია საინფორმაციო საზოგადოების ფორმირების ძირითადი მიმართულებების შესახებ და განათავსეთ იგი საიტის მეორე გვერდზე;
19. შექმენით გვერდები **ფოტოალბომი** და **ვიდეო**. მოძებნეთ თემის შესაფერისი ნახატები და ვიდეომასალა და განათავსეთ ამ გვერდებზე.

გასული საუკუნის მეორე ნახევარში დიდი მოცულობის ინფორმაციის შენახვის, გადაცემისა და დამუშავების მიზნით აუცილებელი გახდა თანამედროვე საშუალებების გამოყენება. დედამიწის მოსახლეობის უდიდესმა ნაწილმა მატერიალური წარმოების სფეროდან ინფორმაციაზე მუშაობის სფეროში გადაინაცვლა. გაჩნდა მოსახლეობის ახალი სოციალური ფენა, რომელიც უშუალოდ არ აწარმოებს მატერიალურ ფასე-

ულობებს. გონებრივი მუშაობის ხვედრითი წილის გაზრდამ იმ ადამიანთა რიცხვიც გაზარდა, რომლებიც შრომის ამ სახეობასთან დაკავშირებულ საქმიანობის სფეროს ირჩევენ. დაიწყო ახალი საზოგადოების ჩამოყალიბება, რომელსაც მეცნიერებმა „საინფორმაციო საზოგადოება“ უწოდეს.

დამატებითი

საინფორმაციო საზოგადოება ელექტრონული მთავრობა

საინფორმაციო საზოგადოება ისეთი საზოგადოებაა, რომლის წევრთა უმრავლესობა ინფორმაციის შექმნით, შენახვით, დამუშავებითა და გამოყენებითაა დაკავებული.

ჩამოგითვლით საინფორმაციო საზოგადოების ზოგიერთ დამახასიათებელ თავისებურებას:

1. ინფორმაციის მოცულობა ძალზე სწრაფად იზრდება და მისი დამუშავებისა და შენახვის მიზნით ადამიანი სპეციალურ ტექნიკურ საშუალებებს იყენებს;

ინფორმაციის მოცულობა იმ ზომამდე გაიზარდა, რომ ადამიანს არ ძალუძს, თავად დაამუშავოს იგი. ამიტომ ამ სამუშაოს სპეციალურ ტექნიკურ საშუალებებს აკისრებს. კომპიუტერები უზრუნველყოფს სანდო საინფორმაციო წყაროების გამოყენებას, ამცირებს არაპროდუქტიული საქმიანობის ხვედრით წილს, აჩქარებს ოპტიმალური გადაწყვეტილებების მიღებას და განაპირობებს ინფორმაციის ავტომატურ დამუშავებას.

2. ძალზე სწრაფად ვითარდება კომპიუტერული ტექნიკა, კომპიუტერული ქსელები, საინფორმაციო ტექნოლოგიები. ინტერნეტის ქსელის მომხმარებელთა რიცხვი მილიონებს ითვლის. თანამედროვე მულტიმედია სისტემების გამოყენება, რომლებიც თავის თავში აერთიანებს სხვადასხვა მონოპოლიზაციის (კომპიუტერის, ტელევიზორის, რადიოს, ტელეფონისა და ა.შ.) ფუნქციებს, ინფორმაციული ტექნოლოგიების უნივერსალიზაციას განაპირობებს. ინფორმაციის შესანახავად განკუთვნილი მონოპოლიზაციები კი განუწყვეტილად პატარავდება და უკვე ხელის გულზეც კი ეტევა. ამ მონოპოლიზაციებში თავისუფლად თავსდება როგორც პირადი, ისე რამდენიმე ენციკლოპედიის ტოლფასი უნივერსალური ცნობარები. ამ მონოპოლიზაციის ინტერნეტის ქსელთან მიერთებით შესაძლებელია ოპერატიული ინფორმაციის მიღება, მაგალითად, ამინდის ან გზებზე არსებული საცობების შესახებ.

3. მიმდინარეობს ფასეულობათა გადაფასება, ცხოვრების და კულტურული დასვენების ყაიდის შეცვლა.

თანამედროვე ადამიანის თავისუფალი დროის უდიდეს ნაწილს კომპიუტერული თამაშები იკავებს. ამჟამად ისინი ქსელურ თამაშებად ტრანსფორმირდება და მათში ერთდროულად რამდენიმე, ტერიტორიულად ერთმანეთისგან მნიშვნელოვნად დაშორებულ ადამიანს შეუძლია მონაწილეობის მიღება. იზრდება ინტერნეტში გატარებული დრო, სადაც შესაძლებელია „მოგზაურობა“ საგანმანათლებლო საიტებზე, ვირტუალურ მუზეუმებში, ლიტერატურის კითხვა და ა.შ.

დიდი პოპულარობით სარგებლობს ონლაინ ურთიერთობის მომსახურებანი, რომლებიც მომხმარებლებს საშუალებას აძლევს, იურთიერთონ შორს მყოფ ადამიანებთან რეალური დროის რეჟიმში.

4. ადამიანების სახლებში გაჩნდა სხვადასხვა ელექტრონული ხელსაწყო და კომპიუტერიზებული მონოპოლიზაცია.

ე.წ. „ჭკვიანი სახლის“ სისტემა, სადენების ერთობლიობის ნაცვლად, ერთი ძალური და ერთი საინფორმაციო კაბელით აღიჭურვება, რომელიც თავის თავზე იღებს ყველა საინფორმაციო კავშირს, საკაბელო ტელევიზიისა და ინტერნეტში გასვლის ჩათვლით. სპეციალური ელექტრონული ბლოკი აკონტროლებს საყოფაცხოვრებო ტექნიკას და ცხოვრების უზრუნველყოფის ყველა სისტემას. „ჭკვიან სახლებს“ მიემატება „ჭკვიანი მანქანები“. მათში, გარდა კომპიუტერისა, რომლებიც ავტომობილის ტექნიკურ ნაწილს გაუწევს მეთვალყურეობას, იფუნქციონირებს ასევე ქალაქის საინფორმაციო სამსახურებთან დაკავშირებული სისტემა. ასეთი ავტომობილი „ჭკვიან სახლთან“ იქნება დაკავშირებული და მის მართვას შეძლებს.

5. ენერგიისა და მატერიალური ფასეულობების წარმოება მანქანებს დაეკისრება, ადამიანი კი, ძირითადად, ინფორმაციის დამუშავებით იქნება დაკავებული.

წარმოებაში ხდება ადამიანთა კადრების შემცირება და რობოტებისა და მანიპულატორების დანერგვა.

6. განათლების სფეროში უწყვეტი განათლების სისტემა შეიქმნება.

ადამიანს მიეცემა საშუალება, მთელი სიცოცხლის განმავლობაში ისწავლოს, რათა ცხოვრებას არ ჩამორჩეს, შეიცვალოს პროფესია და საზოგადოებაში ღირსეული ადგილი დაიკავოს.

7. ბავშვებს საშუალება ეძლევათ, სახლში მიიღონ განათლება კომპიუტერული პროგრამებისა და ტელეკომუნიკაციის საშუალებით.

ამასთან დაკავშირებით, სწავლების პროცესში იცვლება მისი ფორმები და წარმოიქმნება პრობლემები აღზრდის ასპექტში.

8. წარმოიქმნება და ვითარდება საინფორმაციო მომსახურების ბაზარი.

ინფორმაცია იქცევა საქონლად და მომსახურების საგანგებო სახედ, რომლის გაყიდვა და შეძენა ისევე შესაძლებელია, როგორც ნებისმიერი სხვა საქონლისა.

შავისნავლოთ დამოუკიდებლად

შედით ელექტრონული მთავრობის პორტალზე (<https://www.e-gov.az/>) და გაეცანით სხვადასხვა სახელმწიფო სტრუქტურის მიერ შემოთავაზებულ მომსახურების სახეებს. დარეგისტრირდით პორტალზე, ისარგებლეთ განათლების სამინისტროს მომსახურებით „ელექტრონული ცნობა განათლების მიღების ადგილის შესახებ ზოგადსაგანმანათლებლო სკოლების მოსწავლეთათვის“ და მიიღეთ ელექტრონული ცნობა თქვენ შესახებ.

შაამონეა შანი ცოდნა

1. რა უპირატესობა აქვს საიტების შექმნას მზა შაბლონების დახმარებით?
2. როგორ განათავსეთ ტექსტი თქვენ მიერ შექმნილ საიტზე?
3. რა გავიგეთ საინფორმაციო საზოგადოების შესახებ?
4. დაასახელეთ საინფორმაციო საზოგადოების ჩამოყალიბების ძირითადი მიმართულებები



- რა როდენობის სხვადასხვა ობიექტს შეიძლება მიეცეს მისამართი 8 თანამიმდევრული ბიტის მეშვეობით? 32 ბიტის მეშვეობით?
- რით განსხვავდება ქსელში კომპიუტერები ერთმანეთისგან?

საქმიანობა

1. ბრაუზერის მისამართების სტრიქონში აკრიფეთ <http://www.speedguide.net/ip/> და დააჭირეთ კლავიმ <Enter>-ს;
2. გახსნილი გვერდის ველ Search IP address or hostname-ში აკრიფეთ informatika.edu.az და დააჭირეთ კლავიმ <Enter>-ს. გაიხსნება ინფორმაცია იმის შესახებ, თუ სად მდებარეობს საიტ informatika.edu.az-ის სერვერი და მისი განთავსების რუკა.

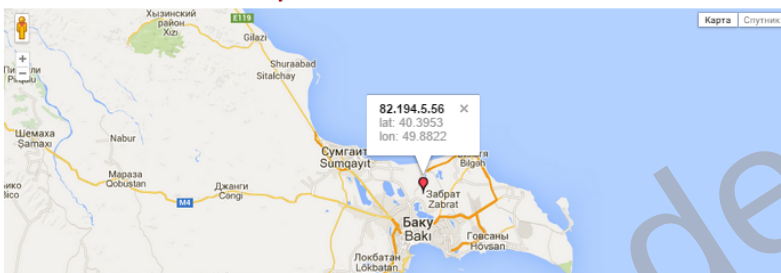
Search IP address or hostname:
Your IP address: 188.72.183.59

82.194.5.56 IP address Information

The IP address 82.194.5.56 was found in Baku, Baki, Azerbaijan. It is allocated to Azedunet LLC. Additional IP location information, as well as network tools are available below.

IP address: 82.194.5.56
ISP: Azedunet LLC
City: Baku
Region: Baki
Country: Azerbaijan (AZ)
Latitude: 40.3953
Longitude: 49.8822

[traceroute](#) [check latency](#) [whois](#) [BGP routing info](#) [blacklist check](#)

82.194.5.56 Location Map**ვიმსჯელოთ:**

- რაზე მიუთითებს აბრევიატურა ISP, რომელიც „Internet Service Provider“-ის შემოკლებული ფორმაა?
- სად მდებარეობს საიტ informatika.edu.az-ის სერვერი?

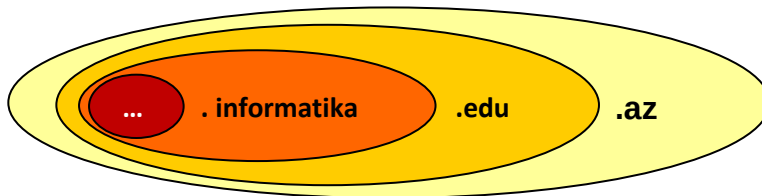
ქსელის ყოველ კომპიუტერს თავისი უნიკალური IP-მისამართი აქვს (იკითხება „აი-პი- მისამართი). თითოეული მისამართი 32-ბიტის ორნიშნა რიცხვია. მოხერხებულობისათვის ამ რიცხვს წერტილებით გამოყოფილი ოთხი ათობითი რიცხვის სახით წარმოადგენენ. ამ ათობითი რიცხვებიდან თითოეულს შეიძლება მიენიჭოს მნიშვნელობა 0-255-ის ფარგლებში.

172 . 16 . 254 . 1
↓ ↓ ↓ ↓
10101100 . 00010000 . 11111110 . 00000001

ცხადია, ციფრების ასეთი ნაკრების დამახსოვრება ძნელია. ამიტომ შეიქმნა **სახელთა დომენური სისტემა – DNS (Domain Name Service)**, რომელიც მომხმარებლებს საშუალებას აძლევს, IP მისამართების ნაცვლად, უფრო გასაგები სახელწოდებები (მაგალითად, www.informatika.edu.az) გამოიყენონ. მაშასადამე, როცა მომხმარებელი მისამართების სტრიქონში შესაბამის სახელს წერს, DNS-სერვერი მას ავტომატურად IP-მისამართად გარდაქმნის.

ქსელში ყველა სახელი სხვადასხვა დონის **დომენებად** არის დაყოფილი. ყველა ქვეყნისთვისაა გამოყოფილი პირველი დონის დომენი. მაგალითად, დომენი „az“ აზერბაიჯანს ეკუთვნის, „de“ – გერმანიას, „ge“ – საქართველოს. ე.წ. ტერიტორიული დომენებისთვის პირველი დონის სხვა დომენებიცაა დამახასიათებელი, მაგალითად, დომენი „.com“ აერთიანებს კომერციულ ორგანიზაციებს, დომენი „.gov“ – სახელმწიფო ორგანიზაციებს. პირველი დონის დომენები ცოტაა და ახალი დომენები იშვიათად ჩნდება. პირველი დონის თითოეული დომენი შეიძლება იყოს უფრო დაბალი დონის უამრავი დომენის მისამართი.

დომენის მფლობელს თავისი დომენის ქვედომენების შექმნა შეუძლია.



ყოველი ფიზიკური თუ იურიდიული პირი, რომელსაც სურს, წარდგენილი იყოს ინტერნეტში, ვალდებულია, განსაკუთრებული ყურადღება მიაქციოს დომენური სახელის შერჩევას. დომენური სახელები შემდეგ პრინციპებს უნდა ექვემდებარებოდეს:

- შერჩეული სახელი საიტის არსს უნდა შეესაბამებოდეს;
- სახელი, შეძლებისდაგვარად, ძალიან გრძელი არ უნდა იყოს;
- უნდა ვეცადოთ, რომ დომენის სახელი არ შეიცავდეს იმ ასოებს, რომლებიც ინგლისურ ანბანში არ არის.

როგორც თითოეულ კომპიუტერს აქვს თავისი მისამართი ინტერნეტში, ისე ყოველ რესურსს – ვებგვერდს აქვს თავისი მისამართი. მას URL-მისამართს („იუ-არ-ელ“) უწოდებენ (აბრევიატურა URL წარმოქმნილია ინგლისური ფრაზის, „Uniform Resource Locator“-ის, შემოკლების შედეგად და **რესურსების ერთიან მაჩვენებელს** ნიშნავს. ბრაუზერი ამ მისამართს ნებისმიერი ინტერნეტრესურსის – დოკუმენტის თუ მისი ნაწილის – მოსაძებნად იყენებს.

URL-მისამართი ერთმნიშვნელოვნად განსაზღვრავს რესურსების ადგილმდებარეობას ინტერნეტში. თითოეული ასეთი მისამართი ორი ნაწილისგან შედგება. პირველი მათგანი რესურსის **იდენტიფიკატორია**, მეორე კი მოცემული რესურსის ადგილმდებარეობაზე მიუთითებს. ტიპური URL-მისამართი ასე გამოიყურება:

<http://informatika.edu.az/tasks.php?action=result>

მისამართის პირველი ნაწილი <http://> პროტოკოლს წარმოადგენს და გვიჩვენებს, რომ რესურსი ვებსაიტია. **პროტოკოლი** წესების ან სტანდარტების ნაკრებია, რომელიც უზრუნველყოფს კომპიუტერებს შორის ურთიერთობისა და მათ შორის ინფორმაციის გაცვლის მონესრიგებას მინიმალური შეცდომებით. იმისათვის, რომ უკეთ გავიგოთ დარჩენილი

მეორე ნაწილი, გავყოთ იგი ორად. პირველი ნაწილი (informatika.edu.az) – იმ კომპიუტერის დომენური სახელია, რომელშიც რესურსია მოთავსებული; მეორე ნაწილი ([/tasks.php?action=result](http://tasks.php?action=result)) კონკრეტული ვებრესურსის (ვებგვერდის) სახელზე მიუთითებს.

დავუშვათ, რომ თქვენ შექმენით საიტი და განუსაზღვრეთ მას დომენური სახელი. მაგრამ როგორ განათავსებთ მას ინტერნეტში? პირველ რიგში, თქვენ უნდა დაარეგისტრირდეთ დომენური სახელი. გაითვალისწინეთ, რომ სათანადო დომენური სახელის შერჩევა საქმის მხოლოდ ნაწილია. სავსებით შესაძლებელია, ეს სახელი უკვე ჰქონდეს ვინმეს დარეგისტრირებული თავის სახელზე.

ღამასწოვრა

დომენური სახელი
დომენური სახელების სისტემა
IP მისამართი
DNS მისამართი
პროტოკოლი
პროვაიდერი
ჰოსტინგი

დომენური სახელების რეგისტრაციას კერძო კომპანიები აწარმოებს. ასე მაგალითად, დომენურ სახელ „az“-ს კომპანია „ინტრანს“ გასცემს. იქ რეგისტრირდება მეორე დონის დომენები. მაგალითად, ასეთი დომენებია საიტების ინტერნეტის ქსელის ნებისმიერ სერვერზე განთავსებითა და მათი მომსახურებითაა დაკავებული ასევე კერძო კომპანიები – **პროვაიდერები**. ასეთ მომსახურებას **ჰოსტინგს** უწოდებენ. ჰოსტინგის მომსახურება ფასიანია, თუმცა არსებობს კომპანიები, რომლებიც უფასო მომსახურებას ეწევა.

შავისნავლოთ დამოუკიდებლად

IP დამისამართების (IPv4) არსებული სისტემის მეშვეობით 4 294 967 296 (256x256x256x256) კომპიუტერს შეიძლება მიეცეს განუმეორებელი სახელი. რატომ გახდა აუცილებელი IP-დამისამართების მე-6 ვერსიაზე გადასვლა? ჩაატარეთ კვლევა და მოამზადეთ პრეზენტაცია IP?6-დამისამართების სისტემის მნიშვნელობის შესახებ.

შეამოწმა შენი ცოდნა

1. რა ფორმითაა წარმოდგენილი კომპიუტერების მისამართები ქსელში?
2. რა IP მისამართი აქვს განათლების სამინისტროს ოფიციალურ საიტს?
3. რას გვიჩვენებს URL-მისამართი?
4. რაში გამოიხატება ჰოსტინგის მომსახურება?
5. წარმოადგინეთ 32-ბიტისანი IP-მისამართი 1100110110100010000101001111010 ოთხი ათობითი, ერთმანეთისგან წერტილებით გამოყოფილი რიცხვის სახით.
6. ჩაწერეთ ოთხი ათობითი რიცხვისგან შედგენილი IP მისამართი: 145.120.30.56 32-ბიტისანი სახით.

შემაჯავებელი კითხვები და დავალებები

1. ჩამოთვალეთ ინფორმაციის მიღებისა და გადაცემის საშუალებები.
2. ახსენით ერთრანგიანი ქსელის მუშაობის პრინციპი.
3. დაადგინეთ თქვენს რეგიონში (ქალაქში) მოქმედი პროვაიდერი კომპანიების სახელწოდებები.
4. რა შემთხვევაში წარმოადგენენ ობიექტის ინფორმაციულ მოდელს გრაფის ფორმით?
5. მოიფიქრეთ მოცემული თემებიდან ერთ-ერთის სტრუქტურა და გამოსახეთ იგი გრაფის ფორმით:
 - ა) „ჩემი სკოლა“;
 - ბ) „ჩემი ოჯახი“;
 - გ) „ჩემი სამშობლო“.
6. რისთვისაა საჭირო IP-მისამართი?
7. დომენური სახელების რომელი ჯგუფებისა და სტრუქტურების კუთვნილებაზე მიუთითებს დომენი „.gov“?
8. გააანალიზეთ შემდეგი დომენური სახელები:
 - a) bakimektebleri.edu.az
 - b) az.wikipedia.org

ცნობილი ალგორითმები

ევკლიდეს ალგორითმი (უდიდესი საერთო გამყოფის პოვნა)

ევკლიდეს ალგორითმი – არის ორი მთელი რიცხვის უდიდესი საერთო გამყოფის პოვნის ალგორითმი.

როგორც მათემატიკის კურსიდან იცით, ორი რიცხვის **უდიდესი საერთო გამყოფი (უსბ)** არის ის უდიდესი რიცხვი, რომელზედაც ეს რიცხვები უნაშთოდ იყოფა.

გამოკლებით უსბ-ის პოვნის ალგორითმის აღწერა

გამოკლებით უსბ-ის პოვნის ალგორითმის აღწერა

1. დიდ რიცხვს გამოვაკლოთ პატარა;
2. თუ 0-ს მივიღებთ, ე.ი. ეს რიცხვები ერთმანეთის ტოლია და უსბ-ს წარმოადგენს (საჭიროა ციკლიდან გამოსვლა);
3. თუ გამოკლების შედეგი 0-ს არ უდრის, მაშინ დიდ რიცხვს შევცვლით გამოკლების შედეგით;
4. გადავალოთ პირველ ნაბიჯზე.

მაგალითი

ვიპოვოთ 30-ისა და 18-ის უსბ.

$$30 - 18 = 12$$

$$18 - 12 = 6$$

$$12 - 6 = 6$$

$$6 - 6 = 0 \quad \text{დასასრული: უსბ საკლებისა და მაკლების ტოლია.}$$

$$\text{უსბ } (30, 18) = 6$$

პროგრამა Python-ის ენაზე

```
a = int(input(' შეიყვანეთ პირველი რიცხვი '))
b = int(input(' შეიყვანეთ მეორე რიცხვი '))

while a != b:
    if a > b:
        a = a - b
    else:
        b = b - a
print(' უსბ = ', a)
```

ორი რიცხვის უსბ-ის პოვნის ეს ალგორითმი ყველაზე ოპტიმალური არ არის, მაგალითად, როცა

$a = 1000\,000$ და $b = 2$ ეს ალგორითმი 500000-ჯერ შესრულდება.

გაყოფით უსბ-ის პოვნის ალგორითმის აღწერა

1. გაყოფით დიდი რიცხვი პატარაზე;
2. თუ ნაშთი ნულია, მაშინ უსბ პატარა რიცხვი იქნება (საჭიროა ციკლიდან გამოსვლა);
3. თუ განაყოფი ნული არ არის, მაშინ დიდი რიცხვი უნდა შევცვალოთ გაყოფის ნაშთით.
4. გადავიდეთ პირველ ნაბიჯზე;

მაგალითი:

ვიპოვოთ 30-ისა და 18-ის უსბ.

$$30/18 = 1 \text{ (ნაშთი 12)}$$

$$18/12 = 1 \text{ (ნაშთი 6)}$$

$$12/6 = 2 \text{ ნაშთი 0. ბოლო გამყოფი არის უსბ: უსბ (30, 18) = 6}$$

პროგრამა Python-ის ენაზე

```
a = int(input(' შეიყვანეთ პირველი რიცხვი ' ))
b = int(input(' შეიყვანეთ მეორე რიცხვი ' ))

while a!=0 and b!=0:
    if a > b:
        a = a % b
    else:
        b = b % a

print (a+b)
```

შენიშვნა:

ციკლში გაყოფის შედეგად დარჩენილი ნაშთი ენიჭება ცვლად a-ს ან ცვლად b-ს. თუ ნაშთი ნულის ტოლია (ჩვენ არ ვიცით, a-შია ის თუ b-ში, ამიტომ ორივე პირობას შევამოწმებთ), ციკლი მთავრდება. ბოლოში, რადგან უცნობია, რომელ ცვლადს მიენიჭა უსბ, გამოიტანება მათი ჯამი – a + b (ერთ-ერთი ცვლადი ყოველთვის ნულის ტოლია და ჯამზე გავლენას ვერ მოახდენს).

$$\begin{aligned} f_g(a, b) &= f_g(a-b, b) \\ &\text{თუ } a > b \\ f_g(36, 48) &= f_g(36, 48-36) = \\ &= f_g(36, 12) = f_g(12, 36-12) = f_g(12, 24) = \\ &= f_g(12, 24-12) = f_g(12, 12) = 12 \end{aligned}$$

პროგრამა Python-ის ენაზე ფუნქციის გამოყენებით

```
def nod(a,b):  
    while a != 0 and b != 0:  
        if a > b:  
            a = a % b  
        else:  
            b = b % a  
    return a + b  
a = int(input(' შეიყვანეთ პირველი რიცხვი '))  
b = int(input(' შეიყვანეთ მეორე რიცხვი '))  
print (nod(a, b))
```

ამოცანები დამოუკიდებელი მუშაობისათვის

1. დაწერეთ პროგრამა, რომელიც პოულობს სამი რიცხვის უდიდეს საერთო გამყოფს.
2. გამოიყენეთ ევკლიდეს ალგორითმი ორი რიცხვის უდიდესი საერთო გამყოფის პოვნისათვის და დაწერეთ პროგრამა N მოცემული რიცხვების უსაზღვროდ.
3. გამოიყენეთ ფორმულა $შსჯ=ab/შსბ(a,b)$ და დაწერეთ პროგრამა ორი ნატურალური რიცხვის, a-სა და b-ს, უმცირესი საერთო ჯერადის (შსჯ-ის) საპოვნელად;
4. მოცემულია N მთელი რიცხვები. იპოვეთ მათ შორის ორი ისეთი რიცხვი, რომელთა შსბ უდიდესი იქნება.
5. მოცემულია N მთელი რიცხვები. იპოვეთ მათ შორის ორი ისეთი რიცხვი, რომელთა შსჯ უმცირესი იქნება.

ფაქტორიალის გამოთვლა

ნატურალური რიცხვის ფაქტორიალს უწოდებენ 1-დან ამ რიცხვამდე ყველა რიცხვის ნამრავლს, მაგალითად რიცხვ 5-ის ფაქტორიალია $1 \times 2 \times 3 \times 4 \times 5 = 120$. ფაქტორიალის გამოთვლის ფორმულა შეიძლება ასე ჩაიწეროს:

$$n! = 1 \times 2 \times \dots \times n,$$

სადაც n მოცემული რიცხვია, ხოლო n! – ამ რიცხვის ფაქტორიალი. ამ ფორმულისა და ციკლ while -ის გამოყენებით, რიცხვის ფაქტორიალის გამოთვლის ფორმულა შეიძლება ასე ჩაწეროთ:

პროგრამა Python-ის ენაზე

```
n = input("რიცხვი, რომლის ფაქტორიალის პოვნაა საჭირო ")  
n = int(n)  
f = 1  
i = 1  
while i < n:  
    f = f * i  
    i = i + 1  
print (n, "! = ", f)
```

დავუშვათ, რომ $n = 5$, ე.ი უნდა გამოვთვალოთ $5!$ ციკლ `while` -ს სხეულში პირველი გავლისას ცვლად `f`-ს ენიჭება მნიშვნელობა 1×1 , მეორე გავლის შემდეგ – 1×2 , შემდეგ – 2×3 , 6×4 და 24×5 . მეექვსეჯერ ციკლი `while` არ შესრულდება, ვინაიდან ცვლად `i`-ს მნიშვნელობა 5 -ს გაუტოლდება და პირობა $i < n$ არ შესრულდება. შევნიშნავთ, რომ პირობის თანახმად, მიღებულია: $0! = 1$ и $1! = 1$

$$0! = 1$$

ამოცანები დამოუკიდებელი მუშაობისათვის

1. დაწერეთ ფორმულის მიხედვით გამოთვლის პროგრამა მოცემული n და k ($k < n$) ნატურალური რიცხვებისათვის.

$$C_n^k = \frac{n!}{k!(n-k)!}$$

2. დაწერეთ ნამრავლ $1 \times 3 \times \dots \times (2n-1)$ -ის გამოთვლის პროგრამა მოცემული ნატურალური რიცხვისათვის.
3. დაწერეთ ნამრავლ $2 \times 4 \times \dots \times (2n)$ -ის გამოთვლის პროგრამა მოცემული n ნატურალური რიცხვისათვის.
4. დაწერეთ პროგრამა უმცირესი n ნატურალური რიცხის საპოვნელად, რომლის დროსაც $n!$ იყოფა 990 -ზე.
5. დაწერეთ პროგრამა, რომელიც მოცემული n ნატურალური რიცხის მიხედვით განსაზღვრავს რიცხვ $n!$ -ში შემავალი ერთიანების რიცხვს.

რიცხვის თვლის ათობითი სისტემიდან ორობითში გადაყვანა

თქვენ იცნობთ რიცხვის თვლის ათობითი სისტემიდან ორობითში გადაყვანის ალგორითმს:

1. მოცემული ათობითი რიცხვი იყოფა 2 -ზე (თვლის ორობითი სისტემის ფუძეზე);
2. განაყოფი მთელი რიცხვის სახით ენიჭება ერთ ცვლადს, ხოლო ნაშთი სტრიქონის სახით – მეორეს (თუ ნაშთი არ არის, იწერება 0);
3. თუ განაყოფი ნულის ტოლი არ არის, ის იყოფა ორზე და შედეგი ენიჭება ცვლადს, რომელშიც ჩანერილი იყო განაყოფი (ცვლადის წინა მნიშვნელობა იშლება), ხოლო სტრიქონული ცვლადის დასაწყისში, რომელიც ნაშთების ჩასანერად იყო გათვალისწინებული, ემატება ახალი ნაშთი;
4. ნაბიჯი 3 მეორდება მანამდე, სანამ ნაშთი ნულს არ გაუტოლდება;
5. სტრიქონულ ცვლადში მოთავსებული ნაშთები იქნება მოცემული ათობითი რიცხვის ორობითი ჩანაწერი. შევნიშნავთ, რომ ნაშთები (0 ან 1) სტრიქონულ ცვლადში განლაგდება მარჯვნიდან მარცხნივ, მათი გამოთვლის თანამიმდევრობის მიხედვით.

პროგრამა Python-ის ენაზე

```
x = int(input("შეიყვანეთ ნატურალური რიცხვი"))
n = ""
while x > 0:
    y = str(x % 2)
    n = y + n
    x = x // 2

print(n)
```

$$10_2 = 2_{10}$$

დავალებები დამოუკიდებელი მუშაობისათვის

1. დაწერეთ პროგრამა, რომელიც პოულობს ერთიანების რიცხვს ათობით რიცხვ 123456-ის ორობით ჩანაწერში.
2. დაწერეთ პროგრამა, რომელიც განსაზღვრავს, რა უფრო მეტია ათობით რიცხვ 135797-ის ორობით ჩანაწერში: ერთიანები თუ წულები.
3. დაწერეთ პროგრამა, რომელიც პოულობს ერთიანების რიცხვს $[1,100]$ ინტერვალის n ნატურალური რიცხვის ორობით ჩანაწერში. n რიცხვი კლავიატურიდან შედის.
4. დაწერეთ პროგრამა $[1,100]$ ინტერვალის იმ ნატურალური რიცხვების რაოდენობის საპოვნელად, რომელთა ორობითი ჩანაწერი მხოლოდ ერთიანებს შეიცავს.
5. დაწერეთ პროგრამა $[a,b]$ ინტერვალის იმ ნატურალური რიცხვების რაოდენობის საპოვნელად, რომელთა ორობითი ჩანაწერი მხოლოდ ერთიანებს შეიცავს.

ფიბონაჩის რიცხვები

იმ მიმდევრობის ელემენტებს, რომელშიც ყოველი მომდევნო რიცხვი, დაწყებული მესამიდან, წინა ორი რიცხვის ჯამს უდრის, ფიბონაჩის რიცხვებს უწოდებენ: 1, 1, 2, 3, 5, 8, 13 და ა.შ.

ფორმულა:

$$F_1 = 1$$

$$F_2 = 1$$

$$F_n = F_{n-1} + F_{n-2}$$

მაგალითი

$$F_3 = F_2 + F_1 = 1 + 1 = 2$$

$$F_4 = F_3 + F_2 = 2 + 1 = 3$$

$$F_5 = F_4 + F_3 = 3 + 2 = 5$$

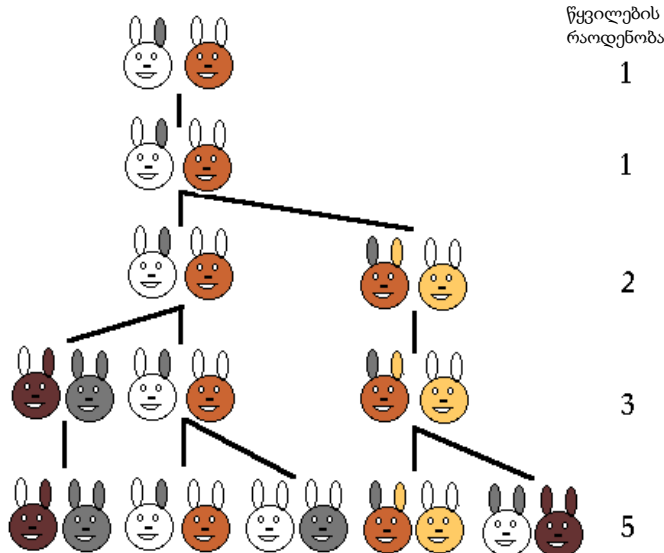
$$F_6 = F_5 + F_4 = 5 + 3 = 8$$

ალგორითმი ფიბონაჩის მიმდევრობის № n რიცხვის გამოსათვლელად

1. შეგვყავს განსასაზღვრავი n ელემენტის ნომერი;
2. ვამოწმებთ: თუ $n < 3$, მაშინ გამოგვაქვს პასუხი 1;
3. ხოლო თუ $n > 3$, მაშინ
4. მიმდევრობის პირველ ორ წევრს fib1 და fib2 ენიჭება საწყისი მნიშვნელობა (1 და 1).
5. შემდეგ, დაწყებული ნომერი 3-დან n-მდე (ჩათვლით), ვასრულებთ:
 - ა. გამოითვლება ჯამი fib1 და fib2 და შედეგი ენიჭება მესამე ცვლადს fib_sum.
 - ბ. წარმოებს ჩანაცვლება $\text{fib1} = \text{fib2}$ და $\text{fib2} = \text{fib_sum}$.
6. გამოგვაქვს პასუხი.

პროგრამა Python-ის ენაზე

```
fib1 = 1
fib2 = 1
n = int(input("რიცხვის ნომერი ფიბონაჩის მიმდევრობაში "))
if n < 3:
    print(1)
else:
    fib1 = 1
    fib2 = 1
    i = 3
    while i <= n:
        fib_sum = fib2 + fib1
        fib1 = fib2
        fib2 = fib_sum
        i = i + 1
    print (fib_sum)
```



ამოცანები დამოუკიდებელი მუშაობისათვის

1. დაწერეთ პროგრამა, რომელსაც ეკრანზე გამოაქვს ფიბონაჩის მიმდევრობის პირველი n რიცხვი. რიცხვი n კლავიატურიდან შედის.
2. დაწერეთ პროგრამა, რომელსაც ეკრანზე გამოაქვს ფიბონაჩის მიმდევრობის n_1 და n_2 ელემენტები. რიცხვები n_1 და n_2 კლავიატურიდან შედის.
3. დაწერეთ პროგრამა, რომელიც განსაზღვრავს, არის თუ არა კლავიატურიდან შეტანილი რიცხვი ფიბონაჩის რიცხვი.
4. დაწერეთ პროგრამა, რომელიც განსაზღვრავს ფიბონაჩის რიცხვების რაოდენობას $[a, b]$ ინტერვალში.
5. დაწერეთ პროგრამა, რომელიც პოულობს $[0, 100]$ ინტერვალში არსებული ყველა ფიბონაჩის რიცხვის ჯამს.

ტერმინების ლექსიკონი

abacus ~ abakus ~ cõtka, abak აბაკი, საანგარიშო – ზოლებად დაყოფილი საანგარიშო დაფა. იყენებდნენ ძველ საბერძნეთსა და რომში არითმეტიკული გამოთვლებისათვის. გამოთვლების დროს პატარა კოჭებს ან ქვებს გადაადგილებენ ზოლებში, რომლებიც რიცხვების თანრიგებს შეესაბამება.

absolute address ~ salt adres ~ mütləq ünvan აბსოლუტური მისამართი
– ელექტრონულ ცხრილებში: უჯრის მისამართი, რომელიც არ იცვლება ფორმულის კოპირების ან გადაადგილების დროს.
– **აბსოლუტური მნიშვნელობა** – რიცხვის აბსოლუტური მნიშვნელობა, ანუ მისი სიდიდე ნიშნის გარეშე.

absolute value (abs) ~ mutlak deęer ~ mütlaq qiymət აბსოლუტი უჯრა – ელექტრონული ცხრილის მიმდინარე (მონიშნული) უჯრა, ანუ უჯრა, რომელიც დროის მიმდინარე მომენტში მუშაობს.

active cell ~ etkin hücre ~ aktiv xana აქტიური ფანჯარა – მრავალფანჯრიან სისტემებში: ფანჯარა, რომელიც მოცემულ მომენტში გამოიყენება მომხმარებლის მონაცემების შესატანად, გამოსახულების შესაქმნელად, ან მენიუს პუნქტის ასარჩევად.

address ~ adres ~ ünvan – მისამართი – 1. გვიჩვენებს უფრას კომპიუტერის მდებარეობაში. ყოველ უფრას თავისი უნიკალური მისამართი აქვს.

2. ელექტრონულ ცხრილში: ასო და რიცხვი, რომლებიც გვიჩვენებს სტრიქონს და სვეტს, რომელთა გადაკვეთაზე მდებარეობს უფრა.

3. რიცხვების ნაკრები ინტერნეტის ქსელში კომპიუტერის აღსანიშნავად.

4. ელექტრონული ფოსტის მისამართი.

5. ვებგვერდის ადგილი ინტერნეტში.

administrator ~ yönetici ~ inzibatçı, administrator ადმინისტრატორი – სპეციალისტი, რომელიც პასუხისმგებელია ქსელისა და მონაცემთა ბაზის მართვის სისტემის კონტროლსა და მომსახურებაზე, მათ დაპროექტებაზე, შექმნასა და კონფიგურაციაზე.

algorithm ~ Igoritma ~ algoritm ალგორითმი – მოქმედებათა სასრული თანამიმდევრობა, მიმართული გარკვეული მიზნის მიღწევისაკენ.

alignment ~ выравнивание ~ hizalama ~ düzləndirmə გასწორება –
 ობიექტების განთავსება ვერტიკალურად და ჰორიზონტალურად ერთსა
 და იმავე დონეზე.

alignment ~ **выравнивание** ~ **hizalama** ~ **düzləndirmə** ანალოგური – ტერმინი ეხება ისეთ ობიექტებს (მაგალითად, ლაპარაკს), რომლებიც მუდმივად (ცვლადი ფიზიკური სიდიდეებით (მაგალითად, ამპლიტუდით, ძაბვით, სიხშირით) აღიწერება.

და F-მდე), ამასთან, 10-დან 15-მდე ციფრებს ინგლისური (ლათინური) ალფაბეტის ასოები შეესაბამება. რიცხვის ზოგიერთ სისტემაში ჩანერისას მის ფუძეს ამ რიცხვის შემდეგ ქვედა ინდექსად ფრჩხილებში წერენ, მაგალითად: $24A_{E(16)}$

bit (binary digit) ~ бит (двоичная цифра) ~ ikil ~ bit (ikilik rəqəm)
ბიტი (ორობითი ციფრი) – ერთ-ერთი სიმბოლო ორიდან (0, 1), რომლებიც
 თვლის ორობით სისტემაში გამოიყენება.

board ~ **пlаtа** ~ **kart** ~ **lövħə**, **kart** პლატა – ზოგადი ტერმინი. ჩვეულებრივ, აღნიშნავს ბრტყელ ელექტრონულ პლატას ელექტრონული კომპონენტებით, რომლებიც ნაბეჭდი ელექტროგამტარი ბილიკებითაა შეერთებული. კომპიუტერის ძირითად პლატას, რომელზეც მიკროპროცესორია მთავსებული, დედაპლატას უწოდებენ.

buffer ~ ნუფერ ~ **arabellek** ~ **bufer** ბუფერი – მონაცემების შუალედური შენახვის ადგილი: მეხსიერების დარეზერვებული უბანი, რომელშიც მონაცემები დამახსოვრებელ მოწყობილობაში გადატანამდე ინახება. ხშირად გამოიყენება მონაცემების გადაცემის ან შეტანა-გამოტანის ოპერაციების დაჩქარების მიზნით. ზოგიერთ მოწყობილობას, მაგალითად, პრინტერს ან მის მხარდამჭერ ადაპტერს, საკუთარი ბუფერი აქვს.

bullet ~ მარკერ ~ madde imi ~ nişan მარკერი – ვიწრო გაგებით: ნერტილი (რგოლი), რომელიც გამოყოფს ტექსტის მცირე ფრაგმენტს ან სიის ყოველ ელემენტს. ამ მიზნით ხშირად სხვა სახის ნიშნებიც გამოიყენება, მაგალითად, კვადრატები და სხვა დეკორატიული ნიშნები.

bus ~ шина ~ veri yolu ~ şin –სალტე – ელექტრული შეერთებების ჯგუფი კომპიუტერის სხვადასხვა კომპონენტს შორის მონაცემების გადაცემისათვის.

byte (bunary term) ~ байт (двоичный элемент) ~ bayt, sekiz ikil ~ bayt (ikilik element) ბაიტი (ორბითი ელემენტი) – ინფორმაციის საზომი ერთეული. შედგება 8 ბიტისგან. ტექნიკასა და მონაცემთა შენახვის სისტემებში: ცალკეული სიმბოლოს, მაგალითად, ასოს, ციფრის ან პუნქტუაციის ნიშნის ეკვივალენტი.

cache ~ кэш ~ önbellek ~ keş კეში – მეხსიერების სპეციალური ქვესისტემა, რომელიც იმასოვრებს განსაკუთრებით ხშირად გამოყენებული მონაცემების მნიშვნელობებს, რის გამოც მათთან წვდომა ძალიან გაადვილებულია.

calculator ~ калькулятор ~ hesap makinesi ~ kalkulator
კალკულატორი – მოწყობილობა, რომელიც რიცხვებზე აწარმოებს ოპერაციებს. ლათინურ ენაზე სიტყვა calculus „კენჭს“ ნიშნავს (უძველეს გამოთვლელ მოწყობილობა აბაკში კენჭებს იყენებდნენ). პირველი ელექტრონული ხელის კალკულატორი ფირმა Texas Instrument-მა გამოუშვა 1967 წელს.

call ~ вызывать ~ çağırmaq ~ çağırmaq გამოძახება – პროგრამაში კოდის რომელიმე განყოფილებისთვის (როგორც წესი, რომელიმე ქვეპროგრამისთვის) ნიშნის მიცემა ძირითადი პროგრამის შესრულების განახლებისათვის აუცილებელი მთელი ინფორმაციის შენახვით.

caption ~ заголовок ~ başlık ~ başlıq სათაური – 1. დოკუმენტის სათაური.

2. ფანჯრის სათაური.

card ~ плата ~ kart ~ kart პლატა – ტერმინი, რომელსაც ხშირად იყენებენ ნაბეჭდ პლატასთან ან ადაპტერთან კავშირში.

cell ~ ячейка ~ hücre, göze ~ xana უჯრა – 1. ელექტრონულ ცხრილში: სვეტისა და სტრიქონის გადაკვეთის ადგილი.

2. მეხსიერების დამისამართებული (ნომრით ან სახელით) ელემენტი.

central processing unit (CPU) ~ центральный процессор (ЦП) ~ merkezi işlem birimi ~ mərkəzi prosessor – ცენტრალური პროცესორი – კომპიუტერის მიერ გამოთვლების შესრულებისა და მართვის ბლოკი; მონყობილობა, რომელიც იცებს და ასრულებს ბრძანებებს.

channel ~ канал ~ kanal ~ kanal არხი – 1. გზა ან კავშირი, რომლითაც ინფორმაცია ერთი მონყობილობიდან მეორეს გადაეცემა. კომპიუტერთან დაკავშირებით არხი შეიძლება იყოს შიდა და გარე. შიდა გარემოში კომპიუტერი იყენებს არხს, რომელსაც ხშირად სალტეს უწოდებენ. მისი საშუალებით სისტემის კომპონენტებს, მაგალითად, კლავიატურასა და კომპიუტერის მეხსიერებას, შორის მონაცემების გაცვლა ხორციელდება.

2. კავშირგაბმულობის სისტემებში: არხი ინფორმაციის გადაცემის მატარებელია. გამოიყენება აგრეთვე ტერმინი „ხაზი“.

character ~ символ ~ karakter ~ simvol სიმბოლო – ასო, ციფრი, პუნქტუაციის ან რომელიმე სხვა სპეციალური ნიშანი, ან მმართველი კოდი, მაგალითად, A, 3 ან &. აუცილებელი არ არის, სიმბოლოს ვიზუალური გამოსახულება ჰქონდეს ეკრანზე ან ქაღალდზე, მაგალითად, შუალედიც ისეთივე სიმბოლოა, როგორიც ასო ან ციფრია.

click ~ щелчок, щелкнуть ~ tık, tıklamak ~ çiqqıltı, çiqqıldatmaq დაწკაპუნება („დაკლიკვა“) – თავის ღილაკზე თითის ერთხელ დაჭერა და მაშინვე აღება (თავის გადაადგილების გარეშე). ჩვეულებრივ, ამას რომელიმე ელემენტის ასარჩევად (ამ დროს მართვის მონყობილობის, მაგალითად, თავის, მაჩვენებელს ასარჩევი ობიექტის „ზემოთ“ ათავსებენ და თავის აქტიურ ღილაკს აჭერენ), ან ამ მოქმედების გასაუქმებლად, ასევე, პროგრამის ან რომელიმე ფუნქციის გასააქტიურებლად აკეთებენ.

click ~ щелчок, щелкнуть ~ tık, tıklamak ~ çiqqıltı, çiqqıldatmaq დაკლიკება – ინფორმაციის დუბლირება ან მისი ასლის შექმნა დოკუმენტის სხვა ნაწილში, სხვა ფაილში, მეხსიერების სხვა უბანში ან სხვა მატარებელზე.

cursor ~ **курсор** ~ **imleç** ~ **kursor** **კურსორი** – სპეციალური ეკრანული ინდიკატორი, მაგალითად, მოციმციმე მოძრავი ნიშანი ან მართკუთხედი, რომელიც არჩეული სიმბოლოს გამოჩენის ადგილს მიუთითებს.

desktop ~ **рабочий стол** ~ **masaüstü** ~ **iş masası** **სამუშაო მაგიდა** – სამუშაო ველი ეკრანზე, რომელიც პიქტოგრამებისა და მენიუს მეშვეობით მაგიდის ზედაპირის მოდელირებას ახდენს. მომხმარებელს შეუძლია, თავისი დახმარებით ზუსტად ისე გადაადგილოს ობიექტები სამუშაო მაგიდაზე, როგორც ამას ჩვეულებრივ მაგიდასთან მუშაობის დროს აკეთებს. Windows-ის სამუშაო მაგიდა სპეციალური კატალოგია თითოეული მომხმარებლისთვის. ჩვეულებრივ, მასზე უამრავი პროგრამული ფაილის იარლიყია განთავსებული.

dialog box ~ **диалоговое окно** ~ **iletişim kutusu** ~ **dialog boxu** **დიალოგური ფანჯარა** – გრაფიკულ სამომხმარებლო ინტერფეისში: სპეციალური ფანჯარა, რომელსაც პროგრამა ან აპლიკაცია გამოაჩენს, როდესაც აუცილებელია მომხმარებლის პასუხი.

digit ~ **цифра** ~ **rakam** ~ **rəqəm** **ციფრი** – ერთ-ერთი სიმბოლო, რომელიც თვლის სისტემაში რიცხვის ჩასაწერად გამოიყენება. თვლის ნებისმიერ სისტემაში შესაძლებელი ციფრების რაოდენობა სისტემის ფუძის ტოლია.

digital-to-analog converter (DAC) ~ **цифро-аналоговый преобразователь (ЦАП)** ~ **sayısal-örneksel dönüştürücü** ~ **rəqəm-analog çeviricisi** – **ციფრულ-ანალოგური გარდაქმნელი** – მოწყობილობა, რომელიც ციფრულ მონაცემებს ანალოგურ სიგნალებად გარდაქმნის.

directory ~ **каталог** ~ **dizin** ~ **katalog** **კატალოგი** – დისკზე შენახული ფაილებისა და კატალოგების ჩამონათვალი; დისკზე ფაილების ორგანიზებისა და დაჯგუფების საშუალება, რომელიც მომხმარებელს თავიდან ააცილებს დიდი რაოდენობის ფაილების დახვეწებას მაგიდაზე. ყველაზე მაღალი დონის კატალოგს ფესვი ეწოდება. კატალოგებს, რომლებსაც სხვა კატალოგები შეიცავს, ქვეკატალოგებს უწოდებენ. Windows-ისა და Mac OS-ის ოპერაციულ სისტემებში კატალოგებს საქალაქდები ჰქვია.

download ~ **загрузка** ~ **indirmek** ~ **endirmek** **ჩამოტვირთვა** – კავშირგაბმულობის სისტემებში: მოდემის ან ქსელური ხაზების მეშვეობით ფაილის ასლის გადაცემის პროცესი დაშორებული კომპიუტერიდან იმ კომპიუტერში, რომელმაც ის მოითხოვა.

drag ~ **перетаскивать** ~ **sürüklemek** ~ **sürüklmek** **გადათრევა** – კომპიუტერულ გრაფიკაში: გამოსახულების ან ობიექტის ეკრანის ერთი აგილიდან მეორეში გადატანა მისთვის „ხელის ჩავლებით“ და საჭირო ადგილზე „გადათრევით“.

extension ~ расширение ~ uzanti ~ uzanti გაფართოება – სიმბოლოების ნაკრები ფაილის სახელწოდებაში, რომელიც მას სახელის შევსების ან ფაილის აღწერის მიზნით ემატება.

file ~ файл ~ dosya ~ fayl ფაილი – ინფორმაციის დასრულებული, სახელდებული ერთობლიობა (მაგალითად, პროგრამა), პროგრამის მიერ გამოყენებულ მონაცემთა ნაკრები, ან მომხმარებლის მიერ შექმნილი დოკუმენტი.

folder ~ папка ~ dizin ~ qovluq ფოლდერი – გრაფიკულ სამომხმარებლო ინტერფეისში: პროგრამებისა და ფაილების შესანახი ადგილი; ეკრანზე სიმბოლურად საქალაქის სახის გრაფიკული ნიშნით გამოსახება. საქალაქად გარკვეული აზრით კატალოგის ეკვივალენტურია.

font ~ шрифт ~ yazı tipi ~ şrift შრიფტი – ერთი სტილის (მაგალითად, Courier), მოხაზულობის (მაგალითად, კურსივი), სისქის (მაგალითად, მუქი) სიმბოლოთა ნაკრები.

gigabyte (GB) ~ гигабайт ~ gigabayt ~ gıqabayt გიგაბაიტი – მკაცრი გაგებით: ერთი მილიარდი ბაიტი. ზუსტი მნიშვნელობა ხშირად იცვლება კონტექსტის მიხედვით: კომპიუტერებში ბაიტების რაოდენობა ხშირად ორის ჯერადი რიცხვით გამოიხატება. ამიტომ გიგაბაიტი შეიძლება ნიშნავდეს როგორც 1000 მბაიტს, ისე 1024 მბაიტს, ანუ 2^{30} (1 048 576) ბაიტს.

graph ~ граф ~ çizge ~ qraf გრაფი – პროგრამირებაში: მონაცემთა სტრუქტურა, რომელიც გარკვეული რაოდენობის (შეიძლება ნულოვანისაც) ერთმანეთთან ნებისმიერი სახით (ნიშნების მეშვეობით) შეერთებული წვეროებისგან შედგება.

hexadecimal (hex) ~ шестнадцатичная система счисления ~ onaltılı sayı sistemi ~ onaltılıq say sistemi თვლის თექვსმეტობითი სისტემა – თვლის სისტემა 16-ის ფუძით, რომლის შემადგენლობაში შედის ციფრები 0-დან 9-მდე და მთავრული ან ჩვეულებრივი ასოები A-დან (რიცხვ 10-ის ეკვივალენტი) F-მდე (რიცხვ 15-ის ეკვივალენტი).

Internet Service Provider (ISP) ~ поставщик услуг Internet, Интернет-провайдер ~ İnternet servis sağlayıcısı ~ İnternet xidmət provayderi ინტერნეტმომსახურების მომწოდებელი, ინტერნეტ-პროვაიდერი – კომპანია, რომელიც ინტერნეტმომსახურებით სარგებლობის საშუალებას აძლევს როგორც ფიზიკურ, ისე იურიდიულ პირებს.

italic ~ курсив ~ italik ~ kursiv კურსივი – შრიფტის მოხაზულობა ოდნავ მარჯვნივ გადახრით. კურსივით, ჩვეულებრივ, უცხოურ სიტყვებს, სათაურებს, ტექნიკურ ტერმინებსა და ციტატებს გამოყოფენ.

modem ~ модем ~ modem ~ modem მოდემი – კავშირგაბმულობის მოწყობილობა, რომელიც კომპიუტერიდან ინფორმაციის ჩვეულებრივი სატელეფონო ხაზით გადაცემის საშუალებას იძლევა.

motherboard ~ материнская плата ~ anakart ~ ana lövhə დედაპლათა – ძირითადი პლათა, რომელიც შეიცავს პროცესორს, ოპერაციულ მეხსიერებას, სალტის კონტროლიორს, შემაერთებლებს და კომპიუტერული სისტემის სხვა კომპონენტებს.

network ~ (вычислительная) сеть ~ ағ ~ şəbəkə (გამომთვლელი) ქსელი – ერთმანეთთან კავშირგაბმულობის საშუალებებით შეერთებული კომპიუტერებისა და მათთან დაკავშირებული მოწყობილობების ჯგუფი. ქსელი შეიძლება ჰქონდეს მუდმივი შეერთება, მაგალითად, კაბელების საშუალებით, და დროებითი – ტელეფონით ან სხვა დროებით კომუტირებული კავშირგაბმულობის ხაზით; მცირე ზომები (მასშტაბი), მაგალითად, ლოკალური გამომთვლელი ქსელი, რომელიც მცირე რაოდენობის კომპიუტერების, პრინტერებისა და სხვა მოწყობილობებისაგან შედგება. არსებობს საკმაოდ დიდი ქსელებიც, მაგალითად, გლობალური. ისინი ფართო გეოგრაფიულ ზონაში განთავსებული მრავალი კომპიუტერისგან შედგება.

password ~ пароль ~ parola ~ parol –პაროლი – კომპიუტერულ სისტემასთან და მის ფაილებთან წვდომის შეზღუდვის ერთ-ერთი საშუალება.

peer-to-peer network ~ одноранговая сеть ~ eşler arası ağ ~ “tay-tuşlar” şəbəkəsi ერთრანგიანი ქსელი – ქსელური არქიტექტურა, დაფუძნებული მასში შემავალი ყველა კომპიუტერის თანასწორობაზე.

peripheral ~ периферийный ~ çevresel, yanbirim ~ çevrəsəl, periferiya პერიფერიული – კომპიუტერთან მიერთებული და მისი მიკროპროცესორით მართული მოწყობილობა, მაგალითად, დისკი, პრინტერი, მოდემი ან ჯოისტიკი.

personal computer (PC) ~ персональный компьютер ~ kişisel bilgisayar ~ fərdi kompüter პერსონალური კომპიუტერი – ერთი ადამიანისთვის განკუთვნილი კომპიუტერი.

pixel ~ пиксель ~ piksel ~ piksel პიქსელი – წერტილი, რომლის მსგავსთა ერთობლიობა ქმნის გამოსახულებას ეკრანზე ან ბეჭდვისას.

pointer ~ указатель ~ göstergeç ~ göstərici მაჩვენებელი – 1. გრაფიკულ გარემოში: ნიშანი ან სიმბოლო, მაგალითად, ისარი, რომელიც თავიდან ან სხვა მოწყობილობით იმართება გამოსახულების ელემენტების, ეკრანის ან მენიუს ჩვენების, მონიშვნის ან არჩევის მიზნით.

port ~ **порт** ~ **port** ~ **port** პორტი – კომპიუტერის პერიფერიული მოწყობილობების მისაერთებელი (ზოგ შემთხვევაში სისტემური ბლოკის უკანა პანელზე მდებარეობს).

printer ~ принтер ~ yazıcı ~ printer პრინტერი - ინფორმაციის ქალაქის მატარებელზე გამოტანი (საბეჭდი), ანუ დოკუმენტის მყარი (ქალაქის) ასლის შექმნილი მოწყობილობა.

RAM (random access memory) ~ оперативная память, оперативное запоминающее устройство (ОЗУ), память прямого доступа, память с произвольной выборкой ~ rastgele erişimli bellek ~ ixtiyari erişimli yaddaş, operativ yaddaş ოპერაციული მეხსიერება, ოპერაციული დამამახსოვრებელი მოწყობილობა (მდმ), პირდაპირი წვდომის მეხსიერება – მეხსიერების მოწყობილობა მონაცემთა შესანახავად, რომელიც დაკავშირებულია მიკროპროცესორთან და მონაწილეობს მის მიერ მონაცემებზე ჩატარებულ ოპერაციებში.

resolution ~ разрешающая способность ~ çözünürlük ~ çözümlülük
გარჩევადობის უნარი – მაჩვენებელი, რომელიც გამოსახულების ხარისხს განსაზღვრავს. გამოსახულების ეკრანზე გამოტანისას მის გარჩევადობას განაპირობებს პიქსელების რაოდენობა ერთ დუიმიზე, ხოლო ბეჭდვისას – ნერტილების რაოდენობა ერთ დუიმიზე.

screen saver ~ хранитель экрана ~ ekran koruyucusu ~ ekran qoruyucusu **ერანის დამცველი** – პროგრამა, რომელიც კომპიუტერის უმოქმედობის (მოცდენის) შემთხვევაში თიშავს ეკრანს ან მასზე ანიმაციური სურათი გამოაქვს.

screenshot ~ снимок экрана ~ ekran görüntüsü ~ ekran görüntüsü – ეკრანის ან მისი რომელიმე ნაწილის გამოსახულების კოპირების შედეგად მიღებული ნახატი.

scroll bar ~ полоса прокрутки ~ kaydırma çubuğu ~ fırlatma zolağı
გადაადგილების ზოლი – გრაფიკულ სამომხმარებლო ინტერფეისში: დოკუმენტის მარჯვენა ან ქვედა ნაწილში არსებული ვერტიკალური ან ჰორიზონტალური ზოლი. იმ შემთხვევაში, როდესაც დოკუმენტი ფანჯარაში არ ეტევა, გამოიყენება მის გადასადგილებლად.

search ~ поиск ~ arama ~ arama, axtarış ძიება – ფაილის ან გარკვეული მონაცემების ადგილმდებარეობის მოძებნა.

search engine ~ поисковая система ~ arama motoru ~ axtarış sistemi
საძიებო სისტემა – პროგრამა, რომელიც უზრუნველყოფს ინფორმაციის მოძიებას საკვანძო სიტყვების, თემების ან სხვა ნიშნების მიხედვით

(მაგალითად, AltaVista, Excite, Google, HotBot, Infoseek, Lycos, Magellan, Rambler, Yahoo!).

sector ~ сектор ~ kesim ~ sektor სექტორი – დისკზე მონაცემების შესანახი უბნის ნაწილი. დისკს აქვს ზედა და ქვედა მხარეები, რომლებზეც წრიული ბილიკები და სექტორები არსებობს. დაფორმატების დროს თითოეული ბილიკი გარკვეული ზომის (ჩვეულებრივ, 512 ბაიტის) სექტორებად იყოფა.

setup ~ установка ~ kurmak ~ quraşdırma დაყენება – პროგრამულ უზრუნველყოფაში: კომპიუტერსა და მასთან მიერთებულ მოწყობილობებში პროგრამის მუშაობისთვის მომზადების პროცედურა.

shortcut ~ ярлык ~ kısayol ~ qısayol იარლიყი – ობიექტის, პროგრამის ან ბრძანების მანიშნებელი ფაილი, რომელიც დამატებით ინფორმაციას შეიცავს. იარლიყებს, ძირითადად, სამუშაო მაგიდაზე ქმნიან ძნელად მისაწვდომ ადგილებში მდებარე პროგრამების სწრაფად გახსნის მიზნით.

software ~ программное обеспечение ~ yazılım ~ program təminatı პროგრამული უზრუნველყოფა – გამომთვლელი მანქანის პროგრამები; ბრძანებები, რომლებიც მანქანის აპარატურულ საშუალებებს ამუშავებს.

source code (SRC) ~ исходный код ~ kaynak kodu ~ ilkin kod საწყისი კოდი – მაღალი დონის ან ასემბლერის ენაზე დაწერილი პროგრამების ოპერატორები, რომლებსაც ადამიანი კითხულობს.

statement ~ оператор ~ deyim ~ deyim /ოპერატორი – ყველაზე მცირე შესრულებადი ელემენტი პროგრამირების ენაში. ზოგადად, პროგრამის თითოეული სტრიქონი ინდივიდუალური ოპერატორია და ცალკე ბრძანებად ითვლება.

status bar ~ строка состояния ~ durum çubuğu ~ status zolağı მიმდინარე მდგომარეობის სტრიქონი – ჰორიზონტალური ზოლი, რომელიც პროგრამის ფანჯრის ქვედა ნაწილში მდებარეობს და მომხმარებლისთვის მნიშვნელოვან ინფორმაციას ასახავს.

storage ~ запоминающее устройство (ЗУ) ~ depolama, bellek ~ yaddasaxlama qurğusu დამმახსოვრებელი მოწყობილობა – გამომთვლელ ტექნიკში: გარე მოწყობილობა, რომელშიც კომპიუტერის ინფორმაცია ინახება.

taskbar ~ панель задач ~ görev çubuğu ~ tapşırıqlar zolağı ამოცანათა პანელი – Windows-ის ოპერაციულ სისტემაში: ჰორიზონტალური პანელი ეკრანის ქვედა ნაწილში, რომელიც ამოცანების სიას შეიცავს. ამოცანათა პანელის მოთავსება ეკრანის ზედა ნაწილში, მის მარჯვენა ან მარცხენა მხარესაც, შეიძლება.

title bar ~ строка заголовка ~ başlık çubuğu ~ başlıq zolağı სათაურის სტრიქონი – გრაფიკულ სამომხმარებლო ინტერფეისში: ჰორიზონტალური ზოლი ფანჯრის ზედა ნაწილში; მასზე მითითებულია ფანჯრის (პროგრამის) სახელი. აქვე სხვა ღილაკებიცაა მოთავსებული.

topology ~ топология ~ topoloji ~ topologiya ტოპოლოგია – ლოკალური ქსელში მონყობილობათა შორის არსებული კავშირით შექმნილი კონფიგურაცია. ქსელების უმრავლესობა სამი სახის ტოპოლოგიას – სალტეს, რგოლსა და ვარსკვლავს – ეფუძნება.

tree ~ дерево ~ ağac ~ ağac ხე – მონაცემთა იერარქიული სტრუქტურა, რომელსაც აქვს ფესვი და რამდენიმე განტოტება. ჩვეულებრივი ხისგან განსხვავებით, როგორც წესი, გადმოზრუნებული სახით გამოისახება.

turtle ~ черепашка ~ kaplumbağa ~ bağa კუ – სამკუთხედის ფორმის ან კუს გამოსახულების მქონე პატარა ფიგურა, რომელიც კომპიუტერის ეკრანზე მაჩვენებლის როლს ასრულებს.

utility ~ утилита ~ yardımcı ~ utilit უტილიტა – მცირე პროგრამა, რომელიც განსაზღვრული ფუნქციის შესასრულებლად არის განკუთვნილი, მაგალითად, რედაქტორი ან გამმართველი.

wallpaper ~ обои ~ duvar kağıdı ~ divar kağızı შპალერი – კომპიუტერულ ტექნიკაში: გამოსახულება სამუშაო მაგიდის გასაფორმებლად გრაფიკულ ოპერაციულ სისტემებში. სამუშაო მაგიდის შპალერად ხშირად იყენებენ რასტრულ გამოსახულებებს, რომელთა ზომები შეესაბამება ეკრანის გარჩევადობას (მაგალითად, 1024X768 ან 1280X1024 პიქსელი).

web browser (= browser) ~ браузер, веб-обозреватель ~ web tarayıcısı, ağ gözeticisi ~ veb brauzer ბრაუზერი, ვებმომხილველი – პროგრამული უზრუნველყოფა, განკუთვნილი ვებსაიტების დათვალიერების, ძიების, დამუშავებისა და ეკრანზე გამოსახვისთვის, ინტერნეტის ქსელში ერთი ვებგვერდიდან მეორეზე გადასვლისათვის.

web page ~ веб-страница ~ ağsayfası ~ veb-səhifə ვებგვერდი – მსოფლიო ქსელში განთავსებული HTML-ის დოკუმენტი.

web server ~ веб-сервер ~ web sunucusu ~ veb-server ვებსერვერი – კომპიუტერი, რომელიც უზრუნველყოფს მომხმარებელთა წვდომას ინტერნეტისა და ინტრანეტის ქსელების სამსახურებთან და ვებგვერდებთან.

web site (site) ~ веб-сайт ~ web sitesi ~ veb-sayt ვებსაიტი – საერთო სახელის, თემისა და სანავიგაციო სისტემის მქონე ვებგვერდების ერთობლიობა.

window ~ окно ~ pencere ~ pəncərə ფანჯარა – მართკუთხა ველი მონიტორის ეკრანზე, რომელიც განკუთვნილია გამოყენებით პროგრამასთან ან მის ნაწილთან დაკავშირებისათვის. დანიშნულების მიხედვით, არჩევენ პროგრამულ ფანჯრებს, დოკუმენტების ფანჯრებს და დიალოგურ ფანჯრებს.

wizard ~ магистр ~ sihirbaz ~ sehirbaz ოსტატი – დამხმარე ფუნქცია. პროგრამაში მუშაობის დროს მომხმარებელს კარნახობს და საჭირო ინფორმაციას აწვდის. როგორც წესი, „ოსტატის“ დახმარება დამწყებ მომხმარებლებს სჭირდებათ ხოლმე.

word processor ~ текстовый процессор ~ kelime işlemci ~ mətn prosessoru ტექსტური პროცესორი – გამოყენებითი კომპიუტერული პროგრამის ნაირსახეობა, განკუთვნილი ყოველნაირ ბეჭდურ დოკუმენტებზე სამუშაოდ (ტექსტის აკრეფა, რედაქტირება, დაფორმატება, დაბეჭდვა).

worksheet ~ рабочий лист ~ işlem tablosu ~ iş vərəqi სამუშაო ფურცელი – ელექტრონული ცხრილის მიერ შექმნილი მონაცემების ფაილი ან მისი ნაწილი. ელექტრონული ცხრილის ფაილს წიგნი ეწოდება. წიგნი რამდენიმე სამუშაო ფურცლისგან შედგება. შესაძლებელია მათი დამატებაც.

workstation ~ рабочая станция ~ iş istasyonu ~ iş stansiyası – სამუშაო სადგური – ზოგადი ტერმინი ლოკალურ ქსელში ან კლიენტ-სერვერის გარემოში პერსონალური კომპიუტერების აღსანიშნავად.

World-Wide Web (WWW) ~ Всемирная паутина, Сеть ~ dünya çapında ağ ~ Dünya hörümçək toru მსოფლიო ქსელი – დანაწილებული სისტემა, რომელიც უზრუნველყოფს ინტერნეტში ჩართულ სხვადასხვა კომპიუტერში განთავსებულ ერთმანეთთან დაკავშირებულ დოკუმენტებთან წვდომას. მსოფლიო ქსელის აღსანიშნავად ხშირად იყენებენ სიტყვა „ვებს“ და აბრევიატურა www-ს.

İNFORMATİKA – 9
Ümumtəhsil məktəblərinin 9-cu sinfi üçün
“İnformatika” fənni üzrə dərslik
Gürcü dilində

Tərtibçi heyət:

Müəlliflər:	Ramin Əli Nazim oğlu Mahmudzadə İsmayıl Calal oğlu Sadıqov Naidə Rizvan qızı İsayeva
Elmi redaktor	R.Əliquliyev , akademik, t.e.d., professor
Tərcüməçi	L.Bakradze
Redaktoru	S.No zadze
Bədii redaktor	T.Məlikov
Texniki redaktor	Z.İsayev
Dizayner	E.Çikarışvili
Rəssamlar	M.Hüseynov, E.Məmmədov
Korrektor	N.Kusraşvili

Dərsliyin gürcü dilində nəşri “Deogene” nəşriyyatı ilə birgə həyata keçirilmişdir.

Azərbaycan Respublikası Təhsil Nazirliyinin
03.06.2016-cı il tarixli 369 №-li əmri ilə təsdiq olunmuşdur.

© Azərbaycan Respublikası Təhsil Nazirliyi – 2017

Müəlliflik hüquqları qorunur. Xüsusi icazə olmadan bu nəşri və yaxud onun hər hansı hissəsini yenidən çap etdirmək, surətini çıxarmaq, elektron informasiya vasitələri ilə yaymaq qanuna ziddir.

Hesab-nəşriyyat həcmi 7,4. Fiziki çap vərəqi 8,5. Səhifə sayı 136.
Kağız formatı 70 × 100^{1/16}. Tiraj 200. Pulsuz. Bakı – 2017.

“YAZ” nəşriyyatı
Bakı, AZ 1001, H.Seyidbəyli küç. 30