პრაქტიკული მეცადინეობა № 1

პრაქტიკული მეცადინეობის თემები:

- გამოსახულების ჩაწერა C++-ზე
- გამოსახულების ტიპის განსაზღვრა
- მონაცემთა ძირითადი ტიპები
- არითმეტიკული ოპერატორები
- გამოსახულების მნიშვნელობის გამოთვლა და ტიპის დადგენა.

მასალა დამოუკიდებლად გასაცნობისათვის:

1. რიცხვებისა და გამოსახულებების (ცხრილის მარცხენა სვეტი) C++-ზე ჩაწერის (ცხრილის მარჯვენა სვეტი) ნიმუშები:

№	გამოსახულება	გამოსახულება C++-ზე
1	0, 3845	0.3845
2	$\frac{1}{6}$	0.166667
3	$-\frac{18}{4}$	-4.5
4	-257,325	-257.325
5	0,0000568	5.68e-005
6	$3a-72b$	3*a-72*b
7	$-9xy$	-9*x*y
8	$0,6n+4,25$	0.6*n+4.25
9	$2b^2+3c-1$	2*b*b+3*c-1
10	$-8m^3+4m^2-5m+6$	-8*m*m*m+4*m*m-5*m+6
11	$1,25ay^2+0,8(b-y)$	1.25*a*y*y +0.8*(b-y)
12	$9a^2-4ab+5b^2$	9*a*a-4*a*b+5*b*b
13	$(5k+2n)(k-4,85n)$	(5*k+2*n)*(k-4.85*n)
14	$(a-b)^2$	(a-b)*(a-b) ან a*a-2*a*b+b*b
15	$(x-y)-7(x+y)^3$	(x-y)-7*(x+y)*(x+y)*(x+y)
16	$\frac{a-0,184b}{ab}$	(a-0.184*b)/(a*b)
17	$[12z-7(x+y)+1]\cdot\frac{y}{z}$	(12*z-7*(x+y)+1)*y/z
18	$\frac{ab}{b+c}-\frac{a+d}{a+b}$	a*b/(b+c)-(a+d)/(a+b)

2. C++-ზე ჩაწერილი გამოსახულებების (ცხრილის მარცხენა სვეტი) ჩვეულებრივი ფორმით ჩაწერის ნიმუშები:

№	გამოსახულება C++-ზე	გამოსახულება
1	0.25*x*x+4*x*y+3.7*y*y	$0,25x^2+4xy+3,7y^2$
2	k*m/(5*m-2*k)	$\frac{km}{5m-2k}$
3	(a+c)/x*y+z/(b+x)	$\frac{(a+c)y}{x}+\frac{z}{b+x}$

4	`d/(a*b)+(b+a)/d`	$\frac{d}{ab}+\frac{b+a}{d}$
5	`d/a*b+(b+a)/d`	$\frac{db}{a}+\frac{b+a}{d}$
6	`(a*b/5-27*(c+d))/(0.43*n)`	$\frac{\frac{ab}{5}-27(c+d)}{0{,}43n}$

3. ზოგიერთი საჭირო ფორმულა:
   1) სამკუთხედის ფართობი: $S=\frac{ah}{2}$, სადაც a სამკუთხედის გვერდია, h კი მასზე დაშვებული სიმაღლე.
   2) წრის ფართობი: $S=\pi R^2$, სადაც R წრის რადიუსია.
   3) წრეწირის სიგრძე: $L=2\pi R$.
   4) მართკუთხედის ფართობი და პერიმეტრი: $S=ab$ და $P=2(a+b)$, სადაც a და b მართკუთხედის გვერდებია.
   5) კვადრატის ფართობი და პერიმეტრი: $S=x^2$ და $P=4x$, სადაც x კვადრატის გვერდია.

სააუდიტორიო სამუშაო:

1. ჩაწერეთ C++-ზე შემდეგი რიცხვები და გამოსახულებები:
   ა) `1542; -67281;` $-\frac{17}{4}$; $\frac{29}{5}$;
   ბ) `36,2247; -0,0000892` - ფიქსირებული წერტილით და სამეცნიერო ფორმით
   გ) $3a+4b+5{,}75$; $-9(x+y)-2{,}3^2$; $5c^2-82ab^3+a^2$; $0{,}75(k-n)+36(m+n)$; $x[\,23(x+y)-3{,}5(y-x)\,]$;
   დ) $\frac{a-b}{15}$; $\frac{-u+3{,}4z}{uz}$; $\frac{y}{x}(n-k)-\frac{73x}{24y}$; $\frac{k(x-y)+m(x+y)}{(y-x)(k+m)}$; $\frac{-\frac{c+d}{3}-8cd}{d-c}$.

2. C++-ზე ჩაწერილი გამოსახულებები ჩაწერეთ ჩვეულებრივი ფორმით:
   ა) `-4.38*a+123.5*b;` ბ) `1/(x*y)-7*y*y;` გ) `(n-m)/(2*m+1.5*n);`
   დ) `0.009*(2*(a+b)-a*b*b);` ე) `-d+2*(3*(a+b)+c);`
   ვ) `5.3/(1.9*(a-c)+d))+b;` ზ) `-36*x/(y*y+x*y+x*x);`
   თ) `u/(z-0.3*u)-0.05/u*z;` ი) `9.2*(k-2*n)/k-7.9/((n+1)*(n-2)+3).`

3. გაიკეთეთ განაცხადი C++-ის ზოგიერთი ტიპის ცვლადზე და განაცხადშივე მოახდინეთ ცვლადების ინიციალიზება (საწყისი მნიშვნელობების მინიჭება):

```
string name = "Giorgi";
int age = 17;
double consult_time = 15.30;
char decimal_point = '.';
bool is_odd = true;
```

```
string name("Giorgi");
int age(17);
double consult_time(15.30);
char decimal_point('.');
bool is_odd(true);
```

4. ცვლადების მნიშვნელობების მიხედვით განსაზღვრეთ მათი ტიპი
   ა) `0.000745;` ბ) `'A';` გ) `"I'm a first year student";` დ) `'?';`

ე) `false`; ვ) `-3275`; ზ) `true`; თ) `492.0275`;
ი) `"The C++ Programming Language"`; კ) `9`.

5. გამოთვალეთ C++-ზე ჩაწერილი შემდეგი გამოსახულებების მნიშვნელობები:

ა) `n/2`, თუ მოცემული გვაქვს განაცხადი `int n = 7;`
ბ) `k/2,` თუ მოცემული გვაქვს განაცხადი `double k = 7;`
გ) `7%2`; დ) `-12%3`; ე) `-(10%4)`.

6. განსაზღვრეთ C++-ზე ჩაწერილი გამოსახულების ტიპი (მთელი თუ ნამდვილი)

ა) `(4-9)*5`; ბ) `3+2.5`; გ) `25/4`; დ) `25./4`; ე) `19%4`; ვ) `-1.5*(7+3)`.

დამოუკიდებელი სამუშაო:

1. ჩაწერეთ C++-ზე შემდეგი რიცხვები და გამოსახულებები:

1) ა) `-6750`; ბ) `-π`; გ) `0,04`; დ) `921,054`; ე) `-0,000075`;
2) ა) `10x`; ბ) `y+48`; გ) `3,25a-14b`; დ) $42p^2-8$; ე) c^3;
3) ა) $\frac{b-a}{3}$; ბ) $(x+y)^2$; გ) c^2+cd+d^2; დ) $-5ax^2+4,25bx-3c$;
4) ა) $\frac{k}{mn}+\frac{mn}{k}$; ბ) $\frac{a-b}{y_1}+\frac{y_2}{a+b}$; გ) $\left(x_1^2+y_1^2\right)\left(x_2^2-y_2^2\right)$; დ) $\frac{c+b}{9a}(b-c)$;
ე) $[3(p-q)+5(p+q)]z$; ვ) $2,7\{x+[(x-2y)^2+5,09y]\}$; ზ) $\frac{(a+1)^2}{a^2-1}$;
5) ა) $\frac{1-2y+y^2}{y^3+1}$; ბ) $\frac{1}{ac+3b-ab}$; გ) $\frac{\frac{1}{n}+\frac{1}{m}}{n-m}$; დ) $10\frac{x+y}{\frac{1}{x}+\frac{1}{y}}$; ე) $-\frac{a-c}{b+d}-\frac{bc}{a+c}$;
6) ა) $\frac{a-b}{b}+2\frac{c+d}{d}+\frac{ad-bc}{cd}$; ბ) $7\frac{(y-z)^2}{y^2}+9\frac{(z+y)^2}{yz^2}$; გ) $\frac{m+\frac{n}{m-n}}{5-\frac{m+n}{m}}$.

2. C++-ზე ჩაწერილი გამოსახულებები ჩაწერეთ ჩვეულებრივი ფორმით:
ა) `k*n/(n+k);` ბ) `x-y/3*z;` გ) `a*b/(c-a)*(b+c);`
დ) `(12*t*(x+y)-z*p)/d;` ე) `m/(6*(k-n)+(3*m-n)/k);`
ვ) `(5.4*(a*a+b)-d)/b*b*d;` ზ) `0.06*(a-b)*(a-b)/((a+b)*(a+b)-1.25);`
თ) `(c/(b+d)-8*a)/(a/(d-b)+b/a*c)`.
3. ჩაწერეთ C++-ზე ფორმულები (იხ. **მასალა დამოუკიდებლად გასაცნობისათვის**, პუნქტი 3)
4. ცვლადების მნიშვნელობების მიხედვით განსაზღვრეთ მათი ტიპი
ა) `'a';` ბ) `-27;` გ) `true;` დ) `"I study C++";` ე) `-101.0037;` ვ) `'9';`
ზ) `false;` თ) `50074;` ი) `3.14159;` კ)`"Bjarne Stroustrup"`.

5. განსაზღვრეთ C++-ზე ჩაწერილი გამოსახულების მნიშვნელობა და ტიპი:
ა) `(3+7.)*12;`
ბ) `c/5+8.35,` თუ `c` მთელი რიცხვია და უდრის `15`-ს;
გ) `(14%3+2)/5;`
დ) `(a+b+c)/3,` თუ მოცემული გვაქვს აღწერა `int a=2, b=1, c=4;`
ე) `(m+n)/2.0,` თუ მოცემული გვაქვს აღწერა `int n=3, m=4;`
ვ) `14-x/4,` თუ `x` ნამდვილი რიცხვია და უდრის `9`-ს.

ლაბორატორიული მეცადინეობა № 1

ლაბორატორიული სამუშაოს გეგმა:

1. დაპროგრამების ინტეგრირებული გარემოს გაცნობა (20 წთ)
2. მარტივი ამოცანების აკრეფა, გამართვა, შესრულება, დამახსოვრება (შენახვა) (30 წთ)

დაპროგრამების ინტეგრირებული გარემოს გაცნობა

- გაააქტიურეთ Visual C++ 2010-ის გარემო.
- შექმენით ახალი პროექტი (გზა: Start Page -> New Project -> Win32, და Win32-ში გაავააქტიუროთ Win32 Console Application), Name ველში აკრიფეთ პროექტის სახელი, მაგალითად A და დააჭირეთ OK ღილაკს (ან Enter-ს). გამოსულ Win32 Application Wizard ფანჯრის Next ღილაკზე დაჭერით გადადით Application Settings ფანჯარაში, მონიშნეთ თვისება Empty Project და დააჭირეთ ღილაკს Finish.
- ჩამოშალეთ View მენიუ და აირჩიეთ Solution Explorer. გახსნილ Solution Explorer ფანჯარაში გაუხსნათ Source Files საქაღალდის კონტექსტური მენიუ (თაგვის მარჯვენა ღილაკით). აირჩიეთ Add -> New Item -> C++ File(.cpp), Name ველში აკრიფეთ პროგრამის სახელი, მაგალითად prog1 და დააჭირეთ Add ღილაკს (ან Enter-ს).
- გაიხსნება ტექსტური ფანჯარა სახელით prog1.cpp, რომელშიც უნდა აკრიფოთ თქვენი პროგრამის კოდი.
- აკრეფილი კოდის გაშვებისთვის: build მენიუში ირჩევთ build solution, შემდეგ, თუ შეცდომების სია ცარიელია, მიღებული შედეგების ეკრანზე გამოტანისთვის, Debug მენიუს ჩამონათვალში აირჩიეთ Start Without Debugging. კომპილერის კითხვაზე, გვინდა თუ არა პროგრამის გამართვა, უპასუხეთ დადებითად. შედეგად მიიღებთ პასუხს შავ ფანჯარაში (კონსოლში).

  შენიშვნა: პროგრამის გაშვება შეიძლება სხვადასხვანაირად. თუ რომელიმე სხვა გზას იყენებთ, მაშინ პროგრამაში უნდა გააააქტიუროთ დაკომენტირებული //system("PAUSE"); შეტყობინება.

ლაბორატორიული სამუშაო:

ამოცანა 1: დაბეჭდეთ ეკრანზე თქვენი სახელი და გვარი. პროგრამას დაურთეთ სათანადო კომენტარები.

შესაბამის C++-პროგრამას აქვს სახე:

```
//////////////////////////////////////////
//    ავტორი:
//    პროგრამა: სტუდენტის ვინაობის ბეჭდვა
//    თარიღი: 06/20/2011
//////////////////////////////////////////
#include <iostream>
using namespace std;
int main()
{
   cout<<"me var . . .\n";
   //system("PAUSE");
   return 0;
}
```

დავალება:

ა) შეიტანეთ პროგრამაში ცვლილება: დაბეჭდეთ სახელი, გვარი და მისამართი.

ბ) სახელი, გვარი და მისამართი დაბეჭდეთ ცალ-ცალკე სტრიქონზე.

გ) ჩასვით გამოსატანი ტექსტის – მისამართის – ბოლოში და შემდეგ შუაში რიგრიგობით სიმბოლოები: \n, \a, \t, \b, \r. თითოეულ შემთხვევაში გაუშვით პროგრამა და გააანალიზეთ შედეგი.

ამოცანა 2: მოცემული მთელი n რიცხვისთვის დაბეჭდეთ მისი მეხუთედის მთელი ნაწილი და ზუსტი მნიშვნელობა.

შესაბამის C++-პროგრამას აქვს სახე:

```
//////////////////////////////////////////////
//     ავტორი:
//     პროგრამა: მთელი n რიცხვის მეხუთედის ბეჭდვა
//     თარიღი: 06/20/2011
//////////////////////////////////////////////
#include <iostream>
using namespace std;
int main()
{
   int n =4;
   cout<< n/5 <<endl;
   cout<< n/5. <<endl;
   system("PAUSE");
   return 0;
}
```

დავალება:

ა) შეასრულეთ პროგრამა და ახსენით მიღებული შედეგი.

ბ) პროგრამა გაუშვით n–ის შემდეგი მნიშვნელობებისთვის: 9, 17, 33 და გააანალიზეთ შედეგები.

ამოცანა 3. მოცემული ნამდვილი d რიცხვისთვის დაბეჭდეთ მისი მეკხრედი.

დამოუკიდებელი სამუშაო:

1. დაწერეთ პროგრამა, რომელიც ეკრანზე გამოიტანს შემდეგ ტექსტს:

   გაიღიმე! გაიღიმე! გაიღიმე!
   გაიღიმე! გაიღიმე!
   გაიღიმე!

2. დაწერეთ პროგრამა, რომელიც მოცემული მთელი a რიცხვისთვის დაბეჭდავს მისი მესამედის მთელ ნაწილს და ზუსტს მნიშვნელობას. პროგრამა შეასრულეთ a–ს შემდეგი მნიშვნელობებისთვის: 2, 10, 25 და ახსენით მიღებული შედეგები.

პრაქტიკული მეცადინეობა № 2

პრაქტიკული მეცადინეობის თემები:

- სტანდარტული ტიპის ცვლადების ბეჭდვა
- არითმეტიკული და შედარების ოპერატორები

მასალა დამოუკიდებლად გასაცნობისათვის:

1. სტანდარტული ტიპის ცვლადების ბეჭდვის ნიმუშები:

№	პროგრამის ფრაგმენტი	შედეგი
1	`cout <<1111<<5555<<"AAAA";`	11115555AAAA
2	`cout<<1111<<endl<<5555<<endl<<"AAAA"<<endl;`	1111 5555 AAAA
3	`cout<<11.11<<endl<<false<<endl<<true<<endl<<'a'<<endl;`	11.11 0 1 a
4	`double d = 12.00357;` `cout<< d <<endl<< 'd'<<endl;`	12.0036 d
5	`int a = 1111;` `cout<< 'a' <<endl<<a<<endl;`	a 1111

შენიშვნა: აქ და ქვემოთ, იმისათვის რომ ვნახოთ პროგრამის ფრაგმენტის შედეგი, იგი უნდა შევავსოთ სრულ პროგრამულ კოდამდე. მაგალითად, მეხუთე ფრაგმენტისთვის უნდა დავწეროთ:

```
#include <iostream>
using namespace std;
int main()
{
    int a = 1111;
    cout<< 'a' <<endl<<a<<endl;

    //system("pause");
    return 0;
}
```

ხოლო სხვა ნიმუშებში შევცვალოთ ერთი ფრაგმენტი მეორეთი.

2. ოპერატორების გამოყენების ნიმუშები:

№	პროგრამის ფრაგმენტი	შედეგი

1	`int b = 2;`
	`b+=3;`
	`cout<<"b gaxda "<<b<<endl;`
		b gaxda 5

2	`double k =5.0148;`
	`k+=6;`
	`cout<<"k udris "<<k<<endl;`
		k udris 11.0148

3	`int b =2; double p =2.5;`
	`b*=2;  p*=2;`
	`cout<<"b udris "<<b<<endl`
	`<<"p udris "<<p<<endl;`
		b udris 4
		p udris 5

4	`int n =20; double q =-25;`
	`n/=6;  q/=6;`
	`cout<<"n udris "<<n<<endl`
	`<<"q udris "<<q<<endl;`
		n udris 3
		q udris -4.16667

5	`int c =15;`	c udris 6

	c%=9; cout<<"c udris "<<c<<endl;	
6	string name = "George"; cout<< name <<endl; cout<< name + ' ' + "Lukas"<<endl;	George George Lukas
7	string name = "George"; name+=" Lukas"; cout<< name<<endl;	George Lukas
8	char c = 'F'; c+=1; cout<< c<<endl;	G

9	int m = 11;
	cout<<m++<<endl;
	cout<<m<<endl;
	cout<<++m<<endl<<m<<endl;
		11
		12
		13
		13

10	char c = 'A';
	cout<<c++;
	cout<<c;
	cout<<++c<<c<<endl;
		ABCC

3. როგორ მუშაობს შედარების ოპერატორები

№	პროგრამის ფრაგმენტი	შედეგი
1	cout<< (37 == 42) <<endl;	0 // მცდარი
2	cout<< (25 > 17) <<endl;	1 // ჭეშმარიტი
3	int x =7, y =5; cout<< (x >= y) <<endl;	1 // ჭეშმარიტი
4	int a =55, b =15; cout<< (a <= b) <<endl;	0 // მცდარი
5	string name = "George"; cout<<(name == "George")<<endl;	1 // ჭეშმარიტი
6	string name = "George"; cout<<(name != "George")<<endl;	0 // მცდარი
7	double p =1.051, q =1.05; cout<< (p > q) <<endl;	1 // ჭეშმარიტი
8	double x =-0.000045, y =-0.0045; cout<< (x <= y) <<endl;	0 // მცდარი

საავტორიო სამუშაო:

1. იპოვეთ m ცვლადის მნიშვნელობა, თუ მოცემული გვაქვს განაცხადი int m =4;

   ა) m +=5; ბ) m *=10; გ) m -=6; დ) m /=8; ე) m %=5;

2. იპოვეთ c ცვლადის მნიშვნელობა, თუ მოცემული გვაქვს განაცხადი char c ='K';

   ა) c +=5; ბ) c -=5; გ) c ='M'; დ) --c;

3. იპოვეთ s ცვლადის მნიშვნელობა, თუ გვაქვს განაცხადი string s ="String";

   ა) s +=" AAAA"; ბ) s = s+' '+"AAAA"+'.'; გ) s ="UUUH!";

4. იპოვეთ x ცვლადის მნიშვნელობა, თუ გვაქვს განაცხადი double x = 11.907;

   ა) x++; ბ) x =(x-1.907)/3; გ) x -=11.807; დ) x +=-5;

5. ვთქვათ, x და y int ტიპის ცვლადებია. იპოვეთ მათი მნიშვნელობები:

   ა) x =(7-3)*9; ბ) y =(4+7)/3*2; გ) y =x =(10+8)/6; დ) y =4+5*(x =7/3);

ე) `x =(8+2)*2.5;` ვ) `y =(2+3)*2.5;` ზ) `y = (int)5.5+4.7;`

6. ა) `s` ცვლადს მიანიჭეთ მთელი `a`, `b` და `c` რიცხვების საშუალო არითმეტიკული. ზუსტი შედეგის მისაღებად როგორი ტიპის უნდა იყოს `s`?

ბ) `k` ცვლადს მიანიჭეთ მთელი `n` და `m` რიცხვების სხვაობის მეცხრედი. ზუსტი შედეგის მისაღებად როგორი ტიპის უნდა იყოს `k`?

გ) `C` ცვლადს მიანიჭეთ მთელი `A` და `B` რიცხვების ჯამის 4-ზე გაყოფისას მიღებული რიცხვის მთელი ნაწილი. დაადგინეთ `C` ცვლადის ტიპი.

7. როგორ გავარკვიოთ ორი ცვლადიდან რომლის მნიშვნელობაა მეტი? ახსენით რას დაბეჭდავს შემდეგი პროგრამა.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
   int n =3, m =2;
   cout<<(n == m)<<'\t'
       <<(n > m)<<endl;
   char p ='B', c ='Q';
   cout<<(p != c)<< '\t'
       <<(p <= c)<<endl;
   double x =10.5, y =-0.25;
   cout<<(x < y)<<'\t'
       <<(x >= y)<<endl;
   string S ="Anano", Q ="Aniko";
   cout<<(S > Q)<<'\t'
       <<(S < Q)<<endl;
   //system("PAUSE");
   return 0;
}
```

პასუხი:

```
0       1
1       1
0       1
0       1
Press any key to continue . . .
```

8. რა მნიშვნელობებს მიიღებს `x` და `y` ცვლადები?

```
int x =5, y =3;
x =y; y =x;
```

9. ახსენით სიტყვებით, რას აკეთებს კოდის შემდეგი ორი ფრაგმენტი:

ა)
```
int x =9, y =5, z;
z = x;
x = y;
y = z;
cout<<x<<' '<<y<<endl;
```

ბ)
```
int x =9, y =5;
x = y + x;
y = x - y;
x = x - y;
cout<<x<<' '<<y<<endl;
```

10. მოცემულია სამნიშნა რიცხვი `n`.

ა) დაბეჭდეთ `n` რიცხვის ციფრთა ჯამი.

```
#include <iostream>
using namespace std;
int main(){
   int n, sum;
   cout<<"Enter integer\n";
   cin>>n;
   sum =n/100+(n/10)%10+n%10;
   cout<<"sum = "<<sum<<endl;
   //system("PAUSE");
   return 0;
}
```

პროგრამის შედეგია:

```
Enter integer
537
sum = 15
Press any key to continue . . .
```

ბ) მთელ k ცვლადს მიანიჭეთ სამნიშნა m რიცხვის შებრუნებული რიგით აღებული ციფრებისგან შედგენილი რიცხვი და დაბეჭდეთ.

```
#include <iostream>
using namespace std;
int main(){
   int m, k;
   cout<<"Enter m"<<endl;
   cin>>m;
   k =100*(m%10)+10*((m/10)%10)+m/100;
   cout<<"k = "<<k<<endl;
   //system("PAUSE");
   return 0;
}
```

პროგრამის შედეგია:

```
Enter m
357
k = 753
Press any key to continue . . .
```

დამოუკიდებელი სამუშაო:

1. ა) A ცვლადს მიანიჭეთ მთელი B და D რიცხვების ჯამის მესამედი. ზუსტი შედეგის მისაღებად როგორი ტიპის უნდა იყოს A?
   ბ) განსაზღვრეთ d ცვლადის ტიპი და d–ს მიანიჭეთ p რიცხვის მეოთხედისა და q რიცხვის მეექვსედის სხვაობა, თუ :
   - p და q მთელი რიცხვებია;
   - p და q ნამდვილი რიცხვებია.
2. იპოვეთ y ცვლადის მნიშვნელობა, თუ მოცემული გვაქვს განაცხადი `int x = 7, y = 9;`

   ა) `y =(x == y);` ბ) `y =(x < y);` გ) `y =(x > y);` დ) `y = x <= y;`
   ე) `x = x != y;` ვ) `x += y;` ზ) `y *= x-1;` თ) `y /= x+1;` ი) `y += 2*x-y.`

3. შეამოწმეთ კომპიუტერზე და გაანალიზეთ მიღებული პასუხები (შეაჯამეთ ცხრილი):

№	პროგრამის ფრაგმენტი	შედეგი

1	`int x =1, z;`
	`z =x++;`
	`cout<<"x = "<<x`	

	<<"\tz = "<<z<<endl;	

2	int x =1, z;
	z =++x;
	cout<<"x = "<<x
	<<" z = "<<z<<endl;	

3	char s ='B', t; t =s--; cout<<"s = "<<s <<"\tt = "<<t<<endl;	

4	char s ='B', t;
	t =--s;
	cout<<"s = "<<s
	<<"\nt = "<<t<<endl;	

5	double d =15.674, q;
	q =++d -10;
	cout<<"d = "<<d
	<<" q = "<<q<<endl;	

6	double d =15.674, q;
	q =d-- +10;
	cout<<"d = "<<d
	<<" q = "<<q<<endl;	

7	string S = "Hello", K ="Students"; S +=", " + K + '!'; cout<<"S = "<<S<<endl;	

8	bool b =false;
	cout<<++b<<endl;
	b -=1;
	cout<<b<<endl;	

4*. რა შედეგი მიიღება პროგრამის შემდეგი ფრაგმენტის შესრულებით?

```
cout << (0.3 == 3*0.1)<< endl;
```

მითითება: მიღებული პასუხის სახსნისთვის ჩაამატეთ პროგრამაში

```
#include <iomanip>
```

და დაბეჭდეთ 3*0.1 –ის მნიშვნელობა 20 ათწილადი ნიშნის სიზუსტით:

```
cout << fixed << setprecision(20);
cout << 3*0.1 << endl;
```

5. ახსენით, რას დაბეჭდავს პროგრამის შემდეგი ფრაგმენტი და რატომ:

ა)
```
string A ="stumari", B ="studenti";
cout << (A < B) << endl << (A != B) << endl;
```

ბ)
```
string C ="kargi ";
cout << (C + "amindi" >= C + "ambavi") << endl;
```

გ)
```
string K ="Word", M ="Excel";
cout << (K > M) << '\t'<<(K =M) << endl;
cout << (K == M) << endl;
```

6. ახსენით, რას აკეთებს კოდის შემდეგი სამი ფრაგმენტი:

ა)
```
char p ='!', c ='?', d;
d = p; p = c; c = d;
cout << p << ' ' << c << endl;
```

ბ)
```
bool t = true, f = false, d;
d = f; f = t; t = d;
cout << boolalpha << "t gaxda " << t
```

```
        << endl << "f gaxda " << f << endl;
```

გ)
```
string S ="Gel", Q ="Euro", P;
P = S; S = Q; Q = P;
cout<<S<< '\t'<<Q<<endl;
```

7. შეამოწმეთ, რას დაბეჭდავს შემდეგი პროგრამა და ახსენით:

```
#include <iostream>
using namespace std;
int main()
{
   char x ='*';
   cout<<x<<' '<<(int)x<<endl;
   int p =127;
   cout<<(char)p<<" simbolos kodi = "
       << p << endl;
   //system("PAUSE");
   return 0;
}
```

8. d ცვლადს რიგრიგობით მიანიჭეთ a რიცხვის ყველა ციფრი, თუ:
   ა) a არის ოთხნიშნა მთელი რიცხვი;
   ბ) a არის ხუთნიშნა მთელი რიცხვი.

9. მოცემულია სამნიშნა N რიცხვი. დაბეჭდეთ N რიცხვის ციფრები შებრუნებული რიგით.

ლაბორატორიული მეცადინეობა № 2

ლაბორატორიული სამუშაოს ამოცანა:

- სტრიქონების წაკითხვა და ბეჭდვა
- მთელი რიცხვების წაკითხვა
- სამი რიცხვის საშუალო არითმეტიკულის გამოთვლა

ლაბორატორიული სამუშაო:

ამოცანა 1. სტრიქონის შეტანა კლავიატურიდან და გამოტანა ეკრანზე.

```
//////////////////////////////////////
//  ავტორი:
//  პროგრამა: სახელის წაკითხვა და ბეჭდვა
//  თარიღი:
//////////////////////////////////////
#include <iostream>
#include <string>
using namespace std;
int main()
{
   cout<<"Please enter your first name"
         " (followed by 'Enter'):\n";
   string first_name;
   cin >> first_name;
   cout << "Hello, " << first_name << "!\n";
   //system("PAUSE");
   return 0;
}
```

დავალება: გაუშვით პროგრამა შესრულებაზე: შეიტანეთ კლავიატურიდან თქვენი სახელი, ბოლოს დააჭირეთ Enter –ს (რითაც დაადასტურებთ შეტანილ ინფორმაციას). გააანალიზეთ მიღებული შედეგი.

ამოცანა 2. სტრიქონის და მთელი რიცხვის შეტანა კლავიატურიდან და გამოტანა ეკრანზე.

```
//////////////////////////////////////////////
//  ავტორი:
//  პროგრამა: სახელის და ასაკის წაკითხვა და ბეჭდვა
//  თარიღი:
//////////////////////////////////////////////
#include <iostream>
#include <string>
using namespace std;
int main(){
   cout<<"Please enter your first name"
         " and age:\n";
   string first_name;
   int age;
   cin >> first_name >> age;
   cout << "Hello, " << first_name << "!\n"
        << "You are " << age << " years old" << endl;
   //system("PAUSE");
   return 0;
}
```

დავალება: გაუშვით პროგრამა შესრულებაზე: შეიტანეთ კლავიატურიდან თქვენი სახელი და ასაკი, შეტანა დაასრულეთ Enter –ზე დაჭერით. გააანალიზეთ მიღებული შედეგი.

ამოცანა 3. პროგრამა "გეკითხებათ" თქვენ სახელსა და გვარს და შემდეგ ბეჭდავს მისალმების სტრიქონს.

```
////////////////////////////////////////////////
//  ავტორი:
//  პროგრამა: სახელისა და გვარის წაკითხვა და ბეჭდვა
//  თარიღი:
////////////////////////////////////////////////
#include <iostream>
#include <string>
using namespace std;
int main(){
   cout<<"Please enter your first name:\n";
   string first_name;
   cin >> first_name;
   cout << "Please enter your last name:\n";
   string last_name;
   cin >> last_name;
   cout << "Hello, " << first_name <<' '
       << last_name<<endl;
   //system("PAUSE");
   return 0;
}
```

დავალება:

ა) გაუშვით პროგრამა შესრულებაზე: შეიტანეთ კლავიატურიდან თქვენი სახელი და შემდეგ გვარი. გააანალიზეთ მიღებული შედეგი.

ბ) გადაწერეთ პროგრამა: პირველი 6 სტრიქონი შეცვალეთ შემდეგი 3 -ით

```
cout<<"Please enter your first and last names\n";
string first_name, last_name;
cin>>first_name>>last_name;
```

კვლავ გაუშვით პროგრამა შესრულებაზე და შეიტანეთ კლავიატურიდან თქვენი სახელი და გვარი. გააანალიზეთ მიღებული შედეგი.

ამოცანა 4. დაბეჭდეთ სხვადასხვა ტიპებისთვის მათი წარმოდგენის უდიდესი და უმცირესი მნიშვნელობები

```
////////////////////////////////////////////
//  ავტორი:
//  პროგრამა: უდიდესი და უმცირესი სიდიდეები
//  თარიღი:
////////////////////////////////////////////
```

ა) მთელი ტიპებისა:

```
#include <iostream>
#include <climits>
using namespace std;
int main()
{
   cout << INT_MIN << '\t' << INT_MAX << endl;
   cout << 0 << '\t' << UINT_MAX << endl;
   cout << CHAR_MIN <<'\t'<< CHAR_MAX << endl;
   cout << 0<<'\t' << UCHAR_MAX << endl;
   cout << SHRT_MIN <<'\t' << SHRT_MAX << endl;
   cout << 0<<'\t' <<USHRT_MAX<<endl;
   cout << LONG_MIN <<'\t' << LONG_MAX << endl;
   cout << 0 <<'\t' <<ULONG_MAX << endl;
```

```
    cout << sizeof(long long) << endl;
    cout << LLONG_MIN << '\t' << LLONG_MAX << endl;
    cout << 0 << '\t' << ULLONG_MAX << endl;
    //system("PAUSE");
    return 0;
}
```

ბ) ნამდვილი ტიპებისა:

```
#include <iostream>
#include <cfloat>
using namespace std;
int main()
{
    cout << FLT_MIN <<' \t' << FLT_MAX << endl;
    cout << DBL_MIN<<'\t' << DBL_MAX << endl;
    //system("PAUSE");
    return 0;
}
```

დავალება: გაუშვით პროგრამა შესრულებაზე: შეიტანეთ კლავიატურიდან სამი მთელი რიცხვი და გააანალიზეთ მიღებული შედეგი.

დამოუკიდებელი სამუშაო:

1. მოცემული სამი მთელი რიცხვისთვის, x ცვლადს მიანიჭეთ მათი საშუალო არითმეტიკული და დაბეჭდეთ იგი.

შესაბამისი C++ –პროგრამის სახეა:

```
//////////////////////////////////////////////
//  ავტორი:
//  პროგრამა: სამი რიცხვის საშუალო არითმეტიკული
//  თარიღი:
//////////////////////////////////////////////
#include <iostream>
using namespace std;
int main(){
    int a, b, c;
    cout<<"Enter 3 integers :\n";
    cin>>a>>b>>c;
    double x;
    x =(a+b+c)/3.;        // საშუალო
    cout << "The average is " << x << endl;
    //system("pause");
    return 0;
}
```

დავალება: გაუშვით პროგრამა შესრულებაზე: შეიტანეთ კლავიატურიდან სამი მთელი რიცხვი და გააანალიზეთ მიღებული შედეგი.

2. რა მოხდება, თუ მთელი რიცხვი უფრო მეტი აღმოჩნდება, ვიდრე ამას ამ რიცხვისათვის არჩეული ტიპი ითვალისწინებს?

გაარჩიეთ პროგრამა:

```
#include <iostream>
#include <climits>
using namespace std;
int main(){
```

```
    int x =INT_MAX, y =INT_MIN;
    cout<<"x = "<<x<<'\t'
        <<"x + 1 = " << x + 1 << '\t'
        <<"x + 2 = " << x + 2 << endl;
    cout<<"y = " << y <<'\t'
        <<"y - 1 = " << y – 1 << '\t'
        <<"y - 2 = " << y – 2 << endl;
    //system("pause");
    return 0;
}
```

დავალება:

ა) შეასრულეთ მოცემული პროგრამა. გააანალიზეთ მიღებული შედეგები.

ბ) შეასწორეთ პროგრამა და შეასრულეთ იგივე ოპერაციები უნიშნო მთელი ტიპის რიცხვებისთვის (ტიპი **unsigned int**). გაითვალისწინეთ, რომ ამ შემთხვევაში x =UINT_MAX, y =0.

პრაქტიკული მეცადინეობა № 3

პრაქტიკული მეცადინეობის თემები:

- **if** შეტყობინება, **if-else** შეტყობინება
- ოპერატორი **? :**

მასალა დამოუკიდებლად გასაცნობისათვის:

1. **if, if-else** შეტყობინებებით განხორციელებული განშტოების რამდენიმე ნიმუში:

№	პროგრამის ტრანსმენტი	შედეგი
1	cout<<"1111\n"; if(5 > 3) cout<<"2222\n"; cout<<"3333\n";	1111 2222 3333
2	cout<<"1111\n"; if(15 <= 3) cout<<"2222\n"; cout<<"3333\n";	1111 3333

3	cout<<"1111\n";
	if(15 <= 3)
	  cout<<"2222\n";
	else
	  cout<<"3333\n";
	cout<<"4444\n";
		1111
		3333
		4444

4	int a =11;
	cout<<"1111\n";
	if(a >= 5)
	  cout<<"2222\n";
	else
	  cout<<"3333\n";
	cout<<"4444\n";
		1111
		2222
		4444

6	if(true) // იგივეა რაც if(1)
	  cout<<"yes";
	else
	  cout<<"no";
		yes

7	if(0) // იგივეა რაც if(false)
	  cout<<"yes";
	else
	  cout<<"no";
		no

9	int k = 7;
	if(k > 0)
	  cout<<k<<" > 0"<<endl;
	else
	  cout<<k<<" <= 0"<<endl;
		7 > 0

10	int a =10,b =111;
	if(a > b)
	  cout<<a<<" > "<<b<<endl;
	else
	  cout<<a<<" <= "<<b<<endl;
		10 <= 111

11	int x =8;
	if(x % 2 == 0)
	  cout<<x<<" is even\n";
	else
	  cout<<x<<" is odd\n";
		8 is even

12	`int x =9;`
	`if(x % 2 == 0)`
	`  cout<<x<<" is even\n";`
	`else`
	`  cout<<x<<" is odd\n";`
		`9 is odd`

13	`int x =12;`
	`if(x % 3 != 0)`
	`  cout<<x<<" ar aris 3-is jeradi\n";`
	`else`
	`  cout<<x<<" 3-is jeradia\n";`
		`12 3-is jeradia`

14	`int x =14;`
	`if(x % 3 != 0)`
	`  cout<<x<<" ar aris 3-is jeradi\n";`
	`else`
	`  cout<<x<<" 3-is jeradia\n";`
		`14 ar aris 3-is jeradi`

2. ჩადგმული **if-else** შეტყობინების გამოყენების რამდენიმე ნიმუში (გავითვალისწინოთ, რომ **else** შეტყობინება ეკუთვნის მის წინ მდგარ პირველივე **if** შეტყობინებას):

№	პროგრამის ფრაგმენტი	შედეგი

1	`if(5 >= 0)`
	`  if(3 > 2) cout<<"yes";`
	`  else cout<<"no";`
		`yes`

2	`if(5 >= 0)`
	`  if(2 > 3) cout<<"yes";`
	`  else cout<<"no";`
		`no`

3	`if(5 < 0)`
	`  if(2 > 3) cout<<"yes";`
	`  else cout<<"no";`	

4	`if(5 < 0)`
	`  if(2 > 3) cout<<"yes";`
	`  else cout<<"no";`
	`else cout<<"this";`
		`this`

3. if-else შეტყობინებაში ლოგიკური ოპერატორების გამოყენების რამდენიმე ნიმუში:

№	პროგრამის ფრაგმენტი	შედეგი

1	`if( 5 > 3  && 123 <= 3736 )`
	`// ეს სტრიქონი კომპილერისთვის იგივეა რაც if( 1 && 1 )`
	`  cout<<"1111\n";`
	`else`
	`  cout<<"2222\n";`
		`1111`

2	`if( 5 > 333  && 123 <= 3736 )`
	`// ეს სტრიქონი კომპილერისთვის იგივეა რაც if( 0 && 1 )`
	`  cout<<"1111\n";`
	`else`
	`  cout<<"2222\n";`
		`2222`

3	`if( 5 > 333  || 123 <= 3736 )`
	`// ეს სტრიქონი კომპილერისთვის იგივეა რაც if( 0 || 1 )`
	`  cout<<"1111\n";`
	`else`
	`  cout<<"2222\n";`
		`1111`

4	`if( 5 > 333  || 123 > 3736 )`
	`// ეს სტრიქონი კომპილერისთვის იგივეა რაც if( 0 || 0 )`
	`   cout<<"1111\n";`
	`else`
	`   cout<<"2222\n";`
		2222

სააუდიტორიო სამუშაო:

1. დაადგინეთ და დაბეჭდეთ

ა) ორ მთელ რიცხვს შორის უდიდესი

```
#include <iostream>
using namespace std;
int main(){
  int a =21, b =15, max;
  if(a > b) max =a;
  else max =b;
  cout<<"udidesi = "<<max<<endl;
  system("pause");
  return 0;
}
```

პასუხი :

```
udidesi = 21
Press any key to continue . . .
```

ბ) ორ სიმბოლოს შორის უდიდესის ამორჩევა **?** **:** ოპერატორის გამოყენებით

```
#include <iostream>
using namespace std;
int main(){
   char p ='A', c ='M', max;
   max = p > c ? p : c;
   cout<<"udidesi aris "<<max<<endl;
   system("pause");
   return 0;
}
```

პასუხი:

```
udidesi aris M
Press any key to continue . . .
```

2. იპოვეთ და დაბეჭდეთ ორ სტრიქონს შორის უმცირესი

```
#include <iostream>
#include <string>
using namespace std;
int main(){
   string S ="netbook", P ="ipod", min;
   min =S;
   if(P < min) min =P;
   cout<<"umciresi striqonia \""
       <<min<<"\""<<endl;
   system("pause");
   return 0;
}
```

პასუხი:

```
umciresi striqonia "ipod"
Press any key to continue . . .
```

3. იპოვეთ და დაბეჭდეთ სამ ნამდვილ ცვლადს შორის უმცირესი

```
#include <iostream>
using namespace std;
int main(){
   double x =1.5,y =2.0,z =-1.2, min;
   min =x;
```

```
    if(y < min) min =y;
    if(z < min) min =z;
    cout<<"min = "<<min<<endl;
    system("pause");
    return 0;
}
```

პასუხი:

```
min = -1.2
Press any key to continue . . .
```

4. იპოვეთ და დაბეჭდეთ y ცვლადის მნიშვნელობა:

ა) $y=\begin{cases} 1-x^2, & \text{თუ } x>-2 \\ 5x+7, & \text{თუ } x\le -2 \end{cases}$

```
#include <iostream>
using namespace std;
int main(){
    int x =2, y;
    if(x >- 2) y =1-x*x;
    else  y =5*x+7;
    cout<<"y = "<<y<<endl;
    system("pause");
    return 0;
}
```

პასუხი:

```
y = -3
Press any key to continue . . .
```

ბ) $y=\begin{cases} x, & \text{თუ } -5\le x\le 5 \\ 2*x-2, & \text{წინააღმდეგ შემთხვევაში} \end{cases}$

```
#include <iostream>
using namespace std;
int main(){
    int x =10, y;
    if(x >= -5 && x <= 5) y =x;
    else  y =2*x-2;
    cout<<"y = "<<y<<endl;
    system("pause");
    return 0;
}
```

პასუხი:

```
y = 18
Press any key to continue . . .
```

გ) $y=\begin{cases} \dfrac{x}{x-5}, & \text{თუ } x\ge 0\ (x\ne 5) \\ 5, & \text{თუ } x=5 \\ 4x^2, & \text{თუ } x<0 \end{cases}$

```
#include <iostream>
using namespace std;
int main(){
    double x =7, y;
    if(x >= 0)
       if(x == 5)y =5;
       else y =x/(x-5);
    else y =4*x*x;
    cout<<"y = "<<y<<endl;
    system("pause");
```

```
    return 0;
}
```

პასუხი:

```
y = 3.5
Press any key to continue . . .
```

დამოუკიდებელი სამუშაო:

1. მიუთითეთ შეკდომები, რომლებიც დაშვებულია პროგრამის შემდეგ ფრაგმენტებში:

ა)
```
int x =1,y =2;
if(x < 2) x =x+1; y =0;
else x =0; y =y+1;
cout<<"x = "<<x<<" y = "<<y<<endl;
```

ბ)
```
int x =-1,y =3;
if x>0 x =x+1;
else {x =0; y =y+1;}
cout<<"x = "<<x<<" y = "<<y<<endl;
```

გ)
```
#include <iostream>
using namespace std;
int main(){
  int x =1,y =2;
   if(x == 1)
      if(y == 1)
         cout<<"x = "<<x<<" y = "<<y<<endl;
   else cout<<"x != 1\n";
  system("pause");
   return 0;
}
```

პროგრამის შესრულების შედეგია :

```
x != 1
Press any key to continue . . .
```

რაც არასწორია. შეეცადეთ პროგრამა შეასწოროთ.

2. რა იქნება პროგრამის შემდეგი ფრაგმენტის შესრულების შედეგი? (შეავსეთ ცხრილი)

№	პროგრამის ფრაგმენტი	შედეგი

1	`int x, y =-3;`
	`if(y != 0) x =10;`
	`cout<<"x = "<<x<<endl;`	

2	`int x, y =-3;`
	`if(y == 0) x =10;`
	`else x =5;`
	`cout<<"x = "<<x<<endl;`	

3	`int  x =1,y =1,z;`
	`if(x > 0 && y <= 1)`
	`  z =x+y;`
	`else`
	`  z =y-x;`
	`cout<<"z = "<<z<<endl;`	

4	`int  x =1,y =1,z;`
	`if(x < 0 && y <= 1)`
	`  z =x+y;`
	`else`
	`  z =y-x;`
	`cout<<"z = "<<z<<endl;`	

3. იპოვეთ და დაბეჭდეთ y ცვლადის მნიშვნელობა. პროგრამა შეასრულეთ `int x` –ის სხვადასხვა მნიშვნელობებისთვის:

ა) $y = \begin{cases} x+1, & \text{თუ } x \geq 5 \\ x-1, & \text{თუ } x < 5 \end{cases}$

ბ) $y = \begin{cases} x^2+5, & \text{თუ } x > 2 \\ -2-x, & \text{თუ } x \leq 2 \end{cases}$

გ) $y = \begin{cases} 1, & \text{თუ } x < -1 \text{ ან } x > 1 \\ 2x-1, & \text{წინააღმდეგ შემთხვევაში} \end{cases}$

დ) $y = \begin{cases} \dfrac{1}{x+5}, & \text{თუ } x \geq 0 \\ 5x^2, & \text{თუ } x < 0 \ (x \neq -5) \\ 0, & \text{თუ } x = -5 \end{cases}$

4. შეადარეთ მოცემული ორი ნამდვილი რიცხვი და დაბეჭდეთ მათ შორის უმცირესი. თუ რიცხვები ტოლია დაბეჭდეთ შესაბამისი გზავნილი.

5. მოცემული მთელი ტიპის ერთი რიცხვი გაყავით მეორეზე და შედეგი დაბეჭდეთ. თუ გამყოფი ნულის ტოლია დაბეჭდეთ შესაბამისი გზავნილი.

6. მოცემულია მთელი რიცხვი `k`. გავზარდოთ მისი მნიშვნელობა 5–ით, თუ იგი ლუწია, ხოლო თუ კენტია – გავზარდოთ 3–ჯერ და დავბეჭდოთ.

7. ქვემოთ მოცემული პროგრამა პოულობს `x` რიცხვის მოდულს:

```
#include <iostream>
using namespace std;
int main(){
 int x =-7, modulx;
   modulx = x >= 0 ? x : -x;
   cout<<x<<" -is moduli udris "<<modulx
       <<" -s"<<endl;
   system("pause");
   return 0;
}
```

პასუხი:

```
-7 -is moduli udris 7 -s
Press any key to continue . . .
```

ა) შეასრულეთ იგი `x`–ის სხვადასხვა მნიშვნელობებისათვის და დარწმუნდით მის სისწორეში;

ბ) დაწერეთ პროგრამა, რომელიც ამოხსნის იგივე ამოცანას **`if-else`** შეტყობინების გამოყენებით.

8. დაწერეთ პროგრამა, რომელიც სამ მთელ რიცხვს შორის აირჩევს უმცირესს და დაბეჭდავს მას.

9*. `k` ცვლადს მიანიჭოთ იმ მეოთხედის ნომერი, რომელშიც ძევს წერტილი `A(x,y)` `(xy!=0)`. დაწერეთ პროგრამა და განიხილეთ შემთხვევები: ა) `A(-1,3);` ბ) `A(2,-5);` გ) `A(-4,-7);` დ) `A(7,4)`.

10*. ცნობილია, რომ მოცემული ოთხი `a, b, c, d` რიცხვიდან სამი ტოლია, ერთი კი განსხვავებული. `n` ცვლადს მიანიჭოთ განსხვავებული რიცხვის სიდიდე. დაწერეთ პროგრამა და ჩაატარეთ მისი ტესტირება სხვადასხვა ოთხეულებისთვის.

ლაბორატორიული მეცადინეობა № 3

ლაბორატორიული სამუშაოს აღწერა:

- **char** ტიპი
- **if-else** შეტყობინება
- ASCII კოდების შესახებ

ლაბორატორიული სამუშაო:

ამოცანა 1. გაუშვით შემდეგი პროგრამა შესრულებაზე:

```
//////////////////////////////////////////
//  ავტორი:
//  პროგრამა: ascii.cpp, რომელიც გვიჩვენებს
//            სიმბოლოების კოდებს
//////////////////////////////////////////
#include <iostream>
using namespace std;
int main()
{
   char x1 ='a', x2 ='b', x3 ='c';
   cout << "simbolo " << x1 << "  kodi " << (int)x1 << endl;
   cout << "simbolo " << x2 << "  kodi " << (int)x2 << endl;
   cout << "simbolo " << x3 << "  kodi " << (int)x3 << endl;
   system("pause");
   return 0;
}
```

დავალება:

1. ჩაატარეთ პროგრამის შედეგის ანალიზი.
2. განსხვადი ცვლადებზე შეცვალეთ შემდეგით:
   ა) **char** x1='0', x2='1', x3='2';
   ბ) **char** x1='K', x2='L', x3='M';

   ორივე შემთხვევაში შეასრულეთ პროგრამა და დარწმუნდით თქვენს დასკვნებში.

ამოცანა 2. გაუშვით შემდეგი პროგრამა შესრულებაზე:

```
//////////////////////////////////////////
//  ავტორი:
//  პროგრამა: სიმბოლური და მთელი ტიპის
//            ცვლადების შეკრება
//////////////////////////////////////////
#include <iostream>
using namespace std;
int main()
{
    char x ='a'; int y =2, k;
    k = x+y;
    cout<<"k = "<<k<<endl
        <<"k = "<<char(k)<<endl;
    system("pause");
    return 0;
}
```

დავალება:

ა). გააანალიზეთ პროგრამის შედეგი.

ბ). შეცვალეთ x –სა და y –ის მნიშვნელობები, კვლავ შეასრულეთ პროგრამა და დარწმუნდით თქვენს დასკვნებში.

ამოცანა 3. ვიპოვოთ ორ სიმბოლოს შორის უდიდესი. ამოვბეჭდოთ მისი მნიშვნელობა და კოდი (**შენიშვნა:** ორ სიმბოლოს შორის უდიდესია ის, რომლის კოდიც მეტია).

```
//////////////////////////////////////////
//  ავტორი:
//  პროგრამა: სიმბოლოებს შორის უდიდესის
//              პოვნა
//////////////////////////////////////////
#include <iostream>
using namespace std;
int main()
{
   char x ='a', y ='k', m;
   // ორ სიმბოლოს შორის უდიდესის კოდი
   if( x > y) m =x;
   else m =y;
   cout<<(int)m<<endl;
   // ორ სიმბოლოს შორის უდიდესი
   cout<<m<<endl;
   system("pause");
   return 0;
}
```

დავალება:

შეცვალეთ პროგრამა ისე, რომ x –სა და y –ის მნიშვნელობების შეტანა ხდებოდეს კლავიატურიდან, კვლავ შეასრულეთ პროგრამა და დარწმუნდით მის სისწორეში.

დამოუკიდებელი სამუშაო:

1. დაადგინეთ 'd' და 'z' სიმბოლოთა კოდები.
2. დაწერეთ პროგრამა, რომელიც გაარკვევს კოდების დიაპაზონს:
   ა)'A'–დან 'Z' სიმბოლომდე. (**მითითება:** საკმარისია დაბეჭდოთ 'A' და 'Z'-ის კოდები).
   ბ)'0'–დან '9' სიმბოლომდე.
3. დაადგინეთ 120-ის ტოლი კოდის მქონე სიმბოლო.
4. იპოვეთ ორ სიმბოლოს შორის უმცირესი. დაბეჭდეთ მისი მნიშვნელობა და კოდი.
5. იპოვეთ სამ სიმბოლოს შორის უდიდესი. დაბეჭდეთ მისი მნიშვნელობა და კოდი.

პრაქტიკული მეცადინეობა № 4

პრაქტიკული მეცადინეობის თემები:

- მათემატიკური ფუნქციების გამოყენება
- ოპერატორი **sizeof**
- ამორჩევის **switch** შეტყობინება. **break** შეტყობინება

მასალა დამოუკიდებლად გასაცნობისათვის:

1. ზოგიერთი მათემატიკური ფუნქცია:
   1) `abs(x)` ფუნქციაა, რომელიც გამოითვლის **მთელი** x რიცხვის მოდულს – **მთელ** რიცხვს. `fabs(x)` ფუნქცია გამოითვლის **ნამდვილი** x რიცხვის მოდულს – **ნამდვილ** რიცხვს. ამ ფუნქციებით სარგებლობისათვის საჭიროა `#include<cmath>` დირექტივის ჩართვა.

№	პროგრამის ფრაგმენტი	შედეგი
1	`cout<<abs(-21)<<endl;`	21
2	`cout<<abs(345)<<endl;`	345
3	`cout<<abs(0)<<endl;`	0
4	`cout<<fabs(-32.25)<<endl;`	32.25
5	`cout<<fabs(98.00457)<<endl;`	98.0046
6	`cout<<fabs(-1)<<endl;`	1

   2) ფუნქცია `sqrt(x)` გამოითვლის **არაუარყოფითი ნამდვილი** x რიცხვიდან კვადრატულ ფესვს – **ნამდვილ** რიცხვს. ფუნქცია აღწერილია `cmath` ფაილში.

№	პროგრამის ფრაგმენტი	შედეგი
1	`cout<<sqrt(2.25)<<endl;`	1.5
2	`cout<<sqrt(-2.25)<<endl;`	-1.#IND
3	`cout<<sqrt(36.)<<endl;`	6
4	`cout<<sqrt(-16.0)<<endl;`	-1.#IND
5	`cout<<sqrt(0.)<<endl;`	0

   3) ფუნქცია `pow(x,y)` გამოითვლის x^y მნიშვნელობას – **ნამდვილ** რიცხვს, სადაც x და y – **ნამდვილი** რიცხვებია. საჭიროებს ფუნქცია აღწერილია `cmath` ფაილში.

№	პროგრამის ფრაგმენტი	შედეგი
1	`cout<<pow(1.44, 0.5)<<endl;`	1.2
2	`cout<<pow(2.0, -3)<<endl;`	0.125
3	`cout<<pow(3.0, 3)<<endl;`	27
4	`cout<<pow(-0.7, 2)<<endl;`	0.49
5	`cout<<pow(-5, 4.0)<<endl;`	625
6	`cout<<pow(-0.2, -3)<<endl;`	-125
7	`cout<<pow(-16,0.5)<<endl;`	-1.#IND

2. ოპერატორი **sizeof**:
   მოცემული კომპიუტერის მეხსიერებაში მოცემული ტიპის ობიექტის ზომის გარკვევა შეიძლება **sizeof** ოპერატორის საშუალებით. პროგრამაში შეგვიძლია ვისარგებლოთ **int** ტიპის მინიმალური `INT_MIN` და მაქსიმალური `INT_MAX` მუდმივების მნიშვნელობებით, **unsigned int** (უნიშნო მთელი) ტიპის მაქსიმალური `UINT_MAX` მუდმივის მნიშვნელობით, **char** ტიპის მინიმალური `CHAR_MIN` და მაქსიმალური `CHAR_MAX` მუდმივების მნიშვნელობებით. ამ მუდმივების მნიშვნელობებით სარგებლობისთვის ჩავრთოთ `#include<climits>` დირექტივა.

float ტიპის მუდმივების მნიშვნელობებია – მინიმალური FLT_MIN და მაქსიმალური FLT_MAX, **double** ტიპის მუდმივების მნიშვნელობებია – მინიმალური DBL_MIN და მაქსიმალური DBL_MAX. ამ მუდმივების მნიშვნელობებით სარგებლობისთვის ჩავრთოთ #include<cfloat> დირექტივა.

№	პროგრამის ფრაგმენტი	შედეგი

1	`// int ტიპის ზომა`
	`cout<<sizeof(int)<<" baiti\n";`
	`// int ტიპის რიცხვების დიაპაზონი`
	`cout<<INT_MIN<<" : "<<INT_MAX<<endl;`
		4 baiti
		-2147483648 : 2147483647

2	`// unsigned int ტიპის ზომა`
	`cout<<sizeof(unsigned int)<<" baiti\n";`
	`// unsigned int ტიპის რიცხვების დიაპაზონი`
	`cout<<0<<" : "<<UINT_MAX<<endl;`
		4 baiti
		0 : 4294967295

3	`// char ტიპის ზომა`
	`cout<<sizeof(char)<<" baiti\n";`
	`// char ტიპის ცვლილების დიაპაზონი`
	`cout<<CHAR_MIN<<" : "<<CHAR_MAX<<endl;`
		1 baiti
		-128 : 127

4	`// unsigned char ტიპის ზომა`
	`cout<<sizeof(unsigned char)<<" baiti\n";`
	`// char ტიპის ცვლილების დიაპაზონი`
	`cout<<0<<" : "<<UCHAR_MAX<<endl;`
		1 baiti
		0 : 255

5	`// float ტიპის ზომა`
	`cout<<sizeof(float)<<" baiti\n";`
	`// float ტიპის რიცხვების დიაპაზონი`
	`cout<<FLT_MIN<<" : "<<FLT_MAX<<endl;`
		4 baiti
		1.17549e-038 : 3.40282e+038

6	`// double ტიპის ზომა`
	`cout<<sizeof(double)<<" baiti\n";`
	`// double ტიპის რიცხვების დიაპაზონი`
	`cout<<DBL_MIN<<" : "<<DBL_MAX<<endl;`
		8 baiti
		2.22507e-308 : 1.79769e+308

3. მაგალითები **switch** შეტყობინების გამოყენებაზე

№	პროგრამის ფრაგმენტი	შედეგი

1	`int a =5;`
	`switch(a){`
	`  case 3: cout<<"a aris 3\n"; break;`
	`  case 5: cout<<"a aris 5\n"; break;`
	`  case 8: cout<<"a aris 8\n"; break;`
	`}`
		`/* დაიბეჭდება */`
		a aris 5

2	`int x =7;`
	`switch(x){`
	`  case 1: cout<<"x = 1\n"; break;`
	`  case 3: cout<<"x = 2\n"; break;`
	`  default: cout<<"x != 1, x != 3\n"; break;`
	`}`
		`/* დაიბეჭდება */`
		x != 1, x != 3

3	`int n =3;` `switch(n){`	`/* დაიბეჭდება */`

	`  case 1: cout<<"Anna\n"; break;`
	`  case 3: cout<<"Irakli\n"; break;`
	`  default: cout<<"Giorgi\n"; break;`
	`}`
		`Irakli`

4	`int n =3;`
	`switch(n){`
	`  case 1: cout<<"Anna\n"; break;`
	`  case 3: cout<<"Irakli\n";`
	`  default: cout<<"Giorgi\n"; break;`
	`}`
		`/* დაიბეჭდება */`
		
		`Irakli`
		`Giorgi`

5	`int n =1, m;`
	`switch(n){`
	`  case 1: m =n+1; cout<<"m = "<<m<<endl;`
	`  case 2: m =3*n; cout<<"m = "<<m; break;`
	`  default: m =n; cout<<"m = "<<m; break;`
	`}`
		`/* დაიბეჭდება */`
		
		`m = 2`
		`m = 3`

6	`int n =1, m;`
	`switch(n){`
	`  case 1: m =n+1; cout<<"m = "<<m<<endl;`
	`  case 2: m =3*n; cout<<"m = "<<m<<endl;`
	`  default: m =n; cout<<"m = "<<m<<endl;break;`
	`}`
		`/* დაიბეჭდება */`
		`m = 2`
		`m = 3`
		`m = 1`

10	`char c ='A';`
	`switch(c){`
	`  case 'A': cout<<"Aleko\n";  break;`
	`  case 'K': cout<<"Kote\n";  break;`
	`  case 'M': cout<<"Mari\n";  break;`
	`  default: cout<<"Gio\n";`
	`}`
		`/* დაიბეჭდება */`
		
		`Aleko`

12	`char c ='K';`
	`switch(c){`
	`  case 'A': cout<<"Aleko\n";  break;`
	`  case 'K': cout<<"Kote\n";`
	`  case 'M': cout<<"Mari\n";`
	`  default: cout<<"Gio\n";`
	`}`
		`/* დაიბეჭდება */`
		
		`Kote`
		`Mari`
		`Gio`

13	`char p;`
	`cin>>p;`
	`switch(p){`
	`  case 'A': case 'O':`
	`  case 'I': case 'E':`
	`  case 'U': cout<<"Xmovani asoa\n"; break;`
	`  default: cout<<"Ar aris xmovani aso\n";`
	`}`
		`/* შეტანა */`
		
		`B`
		`/* გამოტანა */`
		
		`Ar aris xmovani aso`

14	`int a, b =2, c;`
	` cin>>a;`
	` switch( b+a ){`
	`  case 6: c =3;  break;`
	`  case 12: c =2*a-1;  break;`
	`  default: c =a-b;`
	` }`
	` cout<<"c = "<<c<<'\n';`
		`//შეტანა  //გამოტანა`
		`4  c =3`
		
		`10  c =19`
		
		`120  c =118`

საუდიტორიო სამუშაო:

1. ჰერონის ფორმულის მეშვეობით გამოთვალეთ სამკუთხედის ფართობი, თუ მოცემულია სამკუთხედის a, b და c გვერდები. ჰერონის ფორმულის თანახმად,

$$S=\sqrt{p(p-a)(p-b)(p-c)},\quad \text{სადაც}\quad p=(a+b+c)/2.$$

```
#include <iostream>
#include <cmath>
using namespace std;
int main(){
   double a, b, c, p, s;
   cout<<"SemoitaneT a, b, c gverdebis sidideebi :\n";
   cin>>a>>b>>c;
   //თუ ეს გვერდები ქმნიან სამკუთხედს, მაშინ ვისარგებლოთ ჰერონის ფორმულით
   if( a< b+c && b< a+c && c< a+b )
   {
      p =(a+b+c)/2;
      s =sqrt(p*(p-a)*(p-b)*(p-c));
      cout<<"partobi = "<<s<<endl;
   }
   else
   cout<<"aseTi gverdebis mqone samkuTxedi ar arsebobs\n";
   system("pause");
   return 0;
}
```

პროგრამის შესრულების შედეგია:

```
SemoitaneT a, b, c gverdebis sidideebi :
3 4 5
partobi = 6
Press any key to continue . . .
```

2. იპოვეთ და დაბეჭდეთ u ცვლადის მნიშვნელობა, თუ x და y – მთელი რიცხვებია და:

$$u=\begin{cases} x+y, & \text{თუ } |x-y|=3;\\ 2(x+y), & \text{თუ } |x-y|=6;\\ y-3x, & \text{თუ } |x-y|=7;\\ 10, & \text{წინააღმდეგ შემთხვევაში} \end{cases}$$

```
#include <iostream>
using namespace std;
int main(){
  int u, x, y;
  cout<<"x = ";
  cin>>x;
  cout<<"y = ";
  cin>>y;
  if(abs(x - y) == 3) u =x + y;
  else if(abs(x - y) == 6) u = 2*(x + y);
       else if(abs(x - y) == 7) u = y - 3*x;
            else u = 10;
  cout<<"u = "<<u<<endl;
  system("pause");
```

```
  return 0;
}
```

პროგრამის შესრულების შედეგია:

```
x = 3
y = 5
u = 10
Press any key to continue . . .
```

3. იგივე ამოცანა გავაკეთოთ **switch** შეტყობინების გამოყენებით:

```
#include <iostream>
using namespace std;
int main(){
  int u, x, y;
  cout<<"x = ";
  cin>>x;
  cout<<"y = ";
  cin>>y;
  switch( abs(x - y) ){
     case 3: u =x + y; break;
     case 6: u = 2*(x + y); break;
     case 7: u = y - 3*x; break;
     default: u = 10;
  }
  cout<<"u = "<<u<<endl;
  system("pause");
  return 0;
}
```

პროგრამის შესრულების შედეგია:

```
x = -7
y = -4
u = -11
Press any key to continue . . .
```

4. დაწერეთ პროგრამა, რომელიც კლავიატურიდან შეტანილი სიმბოლო-ციფრისთვის გაარკვევს და დაბეჭდავს წარმოადგენს იგი კენტ ციფრს, თუ ლუწს.

```
#include <iostream>
using namespace std;
int main(){
  char cifri;
  cout<<"shemoitaneT simbolo - cifri\n";
  cin>>cifri;
  switch(cifri){
     case '0': case '2':
     case '4': case '6':
     case '8':
          cout<<cifri<<" aris luwi\n"; break;
     case '1': case '3':
     case '5': case '7':
     case '9':
          cout<<cifri<<" aris kenti\n"; break;
     default:
          cout<<cifri<<" ar aris cifri.\n"
                      "Scade Tavidan!\n";
  }
```

```
    system("pause");
    return 0;
}
```

პროგრამის შესრულების შედეგია:

```
shemoitaneT simbolo - cifri
7
7 aris kenti
Press any key to continue . . .
```

5. კლავიატურიდან შეიტანეთ სტუდენტის შეფასება, რომელიც მან მიიღო გამოცდაზე (A, B, C, D, E). დაბეჭდეთ ქულების შესაბამისი დიაპაზონი.

```
#include <iostream>
using namespace std;
int main()
{
    char shefaseba;
    cin>>shefaseba;
    switch(shefaseba)
    {   case 'A': cout<<"91 - 100\n"; break;
        case 'B': cout<<"81 - 90\n"; break;
        case 'C': cout<<"71 - 80\n"; break;
        case 'D': cout<<"61 - 70\n"; break;
        case 'E': cout<<"51 - 60\n"; break;
        default: cout<<"ver chaabareT\n";
    }
   system("pause");
   return 0;
}
```

პროგრამა შესრულების შედეგებია:

```
A
91 - 100
Press any key to continue . . .
```

```
F
ver chaabareT
Press any key to continue . . .
```

დამოუკიდებელი სამუშაო:

1. შეადგინეთ პროგრამა, რომელიც გამოითვლის და დაბეჭდავს მანძილს 0X ღერძის ორ A და B წერტილს შორის.

2. შეადგინეთ პროგრამა, რომელიც გამოითვლის და დაბეჭდავს მანძილს სიბრტყის ორ A(-1.5,2.0) და B(1,-2.5) წერტილს შორის. $A(x_1,y_1)$ და $B(x_2,y_2)$ წერტილებს შორის მანძილი განისაზღვრება ფორმულით

$$d=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}\ .$$

3. შეადგინეთ პროგრამა, რომელიც გამოითვლის და დაბეჭდავს მანძილს სივრცის ორ — A(-1.5,2.0,1.2) და B(1,-2.5,0.5) წერტილს შორის.

$$d=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}\ .$$

4*. ჰერონის ფორმულის მეშვეობით გამოთვალეთ სამკუთხედის ფართობი, თუ მოცემულია სამკუთხედის წვეროების კოორდინატები.

5. გამოთვალეთ $2x^5+3x^3-7$-ის მნიშვნელობა, თუ $x=2.1$.

6. დაწერეთ პროგრამა, რომელიც შეამოწმებს, არის თუ არა მოცემული n წელი ნაკიანი. წელიწადს ეწოდება ნაკიანი, თუ მისი გამომსახველი რიცხვი 4-ის ჯერადია, მაგრამ 100-ის

ჯერადი წლებიდან ნაკიანია მხოლოდ 400-ის ჯერადები. მაგალითად, 1700, 1800 და 1900 – არაა ნაკიანი წლები, ხოლო 2000 – ნაკიანი წელია. n წელიწადის ნაკიანობას შეგამოწმებთ შემდეგი პირობით:

```
n%4 == 0 && n%100 != 0 || n%400 == 0
```

7. რას დაბეჭდავს პროგრამის ფრაგმენტი,

```
int n, m;
cin>>n;
switch(n){
   case 1: m =n+1; cout<<"m = "<<m; break;
   case 2: m =3*n; cout<<"m = "<<m; break;
   default: m =n; cout<<"m = "<<m; break;
}
```

თუ შეიტანთ n -ის შემდეგ მნიშვნელობებს: ა) 1; ბ) 2; გ) 3.

8*. დაწერეთ პროგრამა, რომელიც დაითვლის და დაბეჭდავს კორექტულ შედეგს, თუ შეიტანთ ორ მთელ რიცხვს და შემდეგ არითმეტიკული მოქმედების ბინარულ ოპერატორს (`'+'`, `'-'`, `'*'` ან `'/'`).

ლაბორატორიული მეცადინეობა № 4

ლაბორატორიული სამუშაოს ამოცანა:

- ჩადგმული **if - else**
- **switch** ოპერატორის გამოყენება

ლაბორატორიული სამუშაო:

ამოცანა 1. კვადრატული განტოლების ამოხსნა.

```
//////////////////////////////////////////
//  ავტორი:
//  პროგრამა: კვადრატული განტოლების
//            ამოხსნა
//////////////////////////////////////////
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
   double a,b,c,d,x1,x2;
   cout<<"SeitaneT gantolebis a, b da c koeficientebi\n";
   cin>>a>>b>>c;
   d =b*b-4*a*c;
   if(d < 0)
     cout<<"\ngantolebas namdvili amonaxsni ar gaaCnia\n";
   else if(d == 0)
        {
          x1 =-b/(2*a);
          cout<<"\ngantolebis amonaxsnebia:  "
                "x1 = x2 = "<<x1<<'\n';
        }
        else
        {
          x1 =(-b-sqrt(d))/(2*a);
          x2 =(-b+sqrt(d))/(2*a);
          cout<<"\ngantolebis amonaxsnebia:  "
              "x1 = "<<x1<<'\t'<<"x2 = "<<x2<<'\n';
        }
     system("pause");
     return 0;
}
```

დავალება:

ა) შეასრულეთ პროგრამა შემდეგი მონაცემებისთვის: 1) a =1, b =4, c =4; 2) a =1, b =5, c =6; 3) a =1, b =2, c =9.

ბ) შეიძლება თუ არა ამ ამოცანაში გამოვიყენოთ **switch** ოპერატორი?

გ) პროგრამა იმუშავებს სწორად, თუ შეიტანთ ნულისაგან განსხვავებულ a –ს. შეასრულეთ პროგრამა მონაცემებისთვის a =0, b =2, c =3; და ახსენით მიღებული შედეგი. როგორ გადაწერეთ პროგრამა, რომ მან კორექტულად იმუშავოს a =0 შემთხვევაშიც?

ამოცანა 2. დაწერეთ პროგრამა, რომელიც კლავიატურიდან შეტანილ პატარა ასოსთვის გაარკვევს ხმოვანია იგი თუ თანხმოვანი და დაბეჭდავს შესაბამის დასკვნას.

```
//////////////////////////////////////
//  ავტორი:
//  პროგრამა: ხმოვანი თუ თანხმოვანი?
//////////////////////////////////////
```

```
#include <iostream>
using namespace std;
int main(){
  char aso;
  cout<<"shemoitaneT patara aso\n";
  cin>>aso;
  switch(aso){
     case 'a': case 'o':
     case 'i': case 'e':
     case 'u': cout<<aso<<" aris xmovani\n"; break;
     default: cout<<aso<<" aris Tanxmovani\n";
  }
  system("pause");
  return 0;
}
```

დავალება:

ა) შეასრულეთ პროგრამა სხვადასხვა პატარა ასოებისთვის და დარწმუნდით მის სისწორეში

ბ) რა მოხდება, თუ კლავიატურიდან პატარა ასოს ნაცვლად შეიტანთ სხვა სიმბოლოს? როგორ გავასწოროთ პროგრამის კოდი, რომ ამ შემთხვევაში პროგრამამ დაგვიბეჭდოს სათანადო გზავნილი? **მითითება:** პროგრამის ტექსტში **switch** ოპერატორამდე ჩაამატეთ შემდეგი ფრაგმენტი

```
if(aso < 'a' || aso > 'z')
   cout<<aso<<" ar aris patara aso.\n"
             "Try again!\n";
else
```

კვლავ შეასრულეთ პროგრამა ნებისმიერი სიმბოლოსთვის და დარწმუნდით, რომ იგი ყოველთვის იმუშავებს კორექტულად.

დამოუკიდებელი სამუშაო:

1. გაარჩიეთ ვალუტის გაცვლის პროგრამა, რომელსაც ლარებში შემოტანილი თანხის სიდიდე გადაჰყავს მოთხოვნის შესაბამისად დოლარში (D), ევროში (E) ან ინგლისურ ფუნტში (P).

```
#include <iostream>
using namespace std;
int main(){
  double amount, convert;
  char letter;
  cout<<"Enter amount in GEL : ";
  cin>>amount;
  cout<<"Enter the first letter of currency\n"
        "you wish to convert to (D, E or P): ";
  cin>>letter;
  switch(letter){
     case 'D': convert = amount*0.588408;
               cout<<amount<<" GEL == "<<convert<<" USD\n";
               break;
     case 'E': convert = amount*0.413449;
               cout<<amount<<" GEL == "<<convert<<" EUR\n";
               break;
     case 'P': convert = amount*0.365252;
               cout<<amount<<" GEL == "<<convert<<" GBP\n";
               break;
  }
  system("pause");
  return 0;
```

```
}
```

2. გაარჩიეთ პროგრამა, რომელსაც მოთხოვნის შესაბამისად გადაჰყავს სანტიმეტრები (c) დიუმებში (i) ან პირიქით.

```
#include <iostream>
using namespace std;
int main()
{
   // number of centimeters in an inch
   const double cm_per_inch = 2.54;
   int length; // length in inches or centimeters
   char unit;
   cout<<"Please enter a length followed by a letter (c or i):\n";
   cin>>length>>unit;
   switch (unit) {
     case 'i' :
         cout<<length<<" in == "<<cm_per_inch*length<<" cm\n";
         break;
     case 'c' :
         cout<<length<<" cm == "<<length/cm_per_inch<<" in\n";
         break;
     default:
         cout<<"Sorry, I don't know a unit called '"<<unit<<"'\n";
         break;
   }
   system("pause");
   return 0;
}
```

3. გაარჩიეთ პროგრამა, რომელიც "ხსნის" გადასასვლელზე ქცევის წესებს შუქნიშნის ჩართული ფერის მიხედვით (შემოიტანეთ Yellow, Green ან Red).

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
  string Traffic_light_color;
  cout<<"What color of a traffic light is included? ";
  cin>>Traffic_light_color;
  switch (Traffic_light_color[0]){
       case 'Y': cout<<"Be ready !\n"; break;
       case 'G': cout<<"Go !\n"; break;
       case 'R': cout<<"Stop !\n";
  }
  system("pause");
  return 0;
}
```

4. გაარჩიეთ პროგრამა, რომელიც თვის ნომრის მიხედვით ბეჭდავს თუ წელიწადის რომელ დროს ეკუთვნის მოცემული თვე.

```
#include <iostream>
using namespace std;
int main()
{
  int month_number;
  cout<<"Enter month's number ";
```

```
  cin>>month_number;
  switch(month_number){
    case 12:
    case 1:
    case 2: cout<<"Winter\n"; break;
    case 3:
    case 4:
    case 5: cout<<"Spring\n"; break;
    case 6:
    case 7:
    case 8: cout<<"Summer\n"; break;
    default: cout<<"Autumn\n";
  }
  system("PAUSE");
  return 0;
}
```

5. **switch** შეტყობინების გამოყენებით გამოთვალეთ მთელი d ცვლადის მნიშვნელობა, თუ

$$d=\begin{cases} a\%b+10, & a-c=3;\\ a+b-b/4, & a-c=6;\\ a-b+2c, & a-c=9. \end{cases}$$

a, b და c მთელი რიცხვები შეიტანეთ კლავიატურიდან. შეამოწმეთ თქვენი პროგრამის მუშაობა შემდეგი სანწყისი მონაცემებისათვის:
1). a =2, b =3, c =5; 2). a =2, b =2, c =3; 3). a =2, b =1, c =4.

6. **switch** შეტყობინების გამოყენებით გამოთვალეთ u ცვლადის მნიშვნელობა, თუ

$$u=\begin{cases} 3x-y, & x+2y=1;\\ y-x, & x+2y=2;\\ x+y, & x+2y=3. \end{cases}$$

თუ კი $x+2y$ არ უდრის არც 1-ს, არც 2-ს და არც 3-ს, მაშინ u-ს მივანიჭოთ $x-y$-ის მნიშვნელობა. x და y რიცხვები შეიტანეთ კლავიატურიდან.

პრაქტიკული მეცადინეობა № 5

პრაქტიკული მეცადინეობის თემები:

- **while** შეტყობინება
- ვექტორის (vector) template-სპეციალიზაციები სხვადასხვა ტიპისთვის

მოსალა დამოუკიდებლად გასაკნობისათვის:

1. მაგალითები **while** შეტყობინების გამოყენებაზე:

№	პროგრამის ფრაგმენტი	შედეგი

1	`int counter =0;`
	`while (counter < 3){`
	`  cout<<"Hello, Students!\n";`
	`  counter++;`
	`}`
		`Hello, Students!`
		`Hello, Students!`
		`Hello, Students!`

2	`int number =1;`
	`while (number <= 3){`
	`  cout<<"number = "<<number<<endl;`
	`  number++;`
	`}`
		`number =1`
		`number =2`
		`number =3`

3	`char aso ='F';`
	`while (aso >= 'A'){`
	`  cout<<aso<<' ';`
	`  aso--;`
	`}`
		`F E D C B A`

4	`while ( true )` `  cout<<"usasrulo ganmeoreba\n";`	`//პირობა ყოველთვის ჭეშმარიტია:` `//პროცესი არ დასრულდება`

5	`int n =7;`
	`while ( true ){`
	`  cout<<"n = "<<n<<endl;`
	`  n--;`
	`  if(n == 4) break;`
	`}`
		`n =7`
		`n =6`
		`n =5`

6	`int n =2;`
	`while (n <= 6){`
	`  cout<<"n = "<<n<<endl;`
	`  n +=2;`
	`}`
		`n =2`
		`n =4`
		`n =6`

7	`int n =2;`
	`while ( true ){`
	`  cout<<"n = "<<n<<endl;`
	`  if(n == 6) break;`
	`  n +=2;`
	`}`
		`n =2`
		`n =4`
		`n =6`

8	`int counter =1;`
	`double x =1.2;`
	`while ( true ){`
	`  cout<<x<<endl;`
	`  x +=0.1;`
	`  if(counter == 5) break;`
	`  counter ++;`
	`}`
		`1.2`
		`1.3`
		`1.4`
		`1.5`
		`1.6`

9	`while ( false )`
	`  cout<<"Sesruldeba?\n";`
	`cout<<"ar Sesrulda\n";`
		`ar Sesrulda`
		`// პირობა თავიდანვე მცდარია:`
		`//შეტყობინება არ შესრულდება`

2. მაგალითები **vector** –ის სხვადასხვა სპეციალიზაციის გამოყენებაზე:

№	პროგრამის ტრაგმენტი	შედეგი
1	cout<<"test #1 of vector"<<endl; vector<int> vect1; cout<<"size of vect1 : "<<vect1.size()<<endl;	//ცარიელი ვექტორი test #1 of vector size of vect1 : 0

2	cout<<"test #2 of vector"<<endl;
	vector<int> vector1;
	//ორ ელემენტს ვამატებთ ვექტორში
	vector1.push_back(7777);
	vector1.push_back(2345);
	cout<<"first element is :"<<vector1[0]<<endl;
	cout<<"second element is :"<<vector1[1]<<endl;
		test #2 of vector
		first element :7777
		second element :2345

3	cout<<"test #3 of vector"<<endl;
	vector<int> vector1(3);
	cout<<"vector1.size()="<<vector1.size()<<endl;
	cout<<"first element is:"<<vector1[0] <<endl;
	cout<<"second element is:"<<vector1[1]<<endl;
	cout<<"3-th element is:"<<vector1[2]<<endl;
	cout<<"4-rd element is:"<<vector1[3]<<endl;
		//სამი ელემენტის ადგილი
		//იკავშნება და იწყება
		// ნულებით
		test #3 of vector
		vector1.size()=3
		first element is:0
		second element is:0
		3-th element is:0
		//Debug Assetion
		// Failed!
		//Expression: vector
		// subscript out of
		// range

4	cout<<"test #4 of vector"<<endl;
	vector<double> vector1(2);
	vector1[0] = 11.11;
	vector1[1] = 22.22;
	cout<<"first element:"<<vector1[0] <<endl;
	cout<<"second element:"<<vector1[1]<<endl;
		test #4 of vector
		first element:11.11
		second element:22.22

5	cout<<"test #5 of vector"<<endl;
	vector<double> vector1(2);
	vector1[0] = 11.1;
	vector1[1] = 22.2;
	vector1.push_back(77.77);
	vector1.push_back(23.45);
	cout<<"vector1[0]="<<vector1[0] <<endl;
	cout<<"vector1[1]="<<vector1[1]<<endl;
	cout<<"vector1[2]="<<vector1[2]<<endl;
	cout<<"vector1[3]="<<vector1[3]<<endl;
		test #5 of vector
		vector1[0]=11.1
		vector1[1]=22.2
		vector1[2]=77.77
		vector1[3]=23.45

6	cout<<"test #6 of vector"<<endl;
	string str;
	vector<string> a(2);
	a[0] = "Good ";
	a[1] = "By";
	str = a[0]+' '+a[1];
	cout<<str<<endl;
		test #6 of vector
		Good By

7	cout<<"test #6 of vector"<<endl;
	string str;
	vector<string> vctr;
	vctr.push_back("Good");
	vctr.push_back("By!");
	str = vctr[0]+' '+vctr[1];
	cout<<str<<endl;
		test #6 of vector
		Good By!

საუდიტორიო სამუშაო:

1. კლავიატურიდან შეიტანეთ 5 მთელი რიცხვი და ჩაწერეთ ვექტორში. შემდეგ დაბეჭდეთ ვექტორის ელემენტები ინდექსების ჩვენებით:

```
#include <iostream>
#include <vector>
using namespace std;
int main(){
  vector<int> a;
  int number, index = 0;
  while(index < 5){
     cin>> number;
     a.push_back(number);
     index++;
  }
  index = 0;
  while(index < a.size()){
     cout<<"a[ "<<index<<" ] = "<<a[index]<<endl;
     index++;
  }
  //system("PAUSE");
  return 0;
}
```

პროგრამის შესრულების შედეგია:

```
1 2 3 4 5
a[ 0 ] = 1
a[ 1 ] = 2
a[ 2 ] = 3
a[ 3 ] = 4
a[ 4 ] = 5
Press any key to continue . . .
```

2. კლავიატურიდან შეიტანეთ 10 ნამდვილი რიცხვი (შეტანა დაამთავრეთ სიმბოლოს აკრეფით) და ჩაწერეთ ვექტორში. შემდეგ იპოვეთ ამ რიცხვების ჯამი და დაბეჭდეთ.

```
#include <iostream>
#include <vector>
using namespace std;
int main(){
  vector<double> v;
  double x;
  while(cin>>x)
     v.push_back(x);
  double sum =0;
  int i = 0;
  while(i < v.size()){
     sum += v[i];
     i++;
  }
  cout<<"jami = "<<sum<<endl;
  //system("PAUSE");
  return 0;
}
```

პროგრამის შესრულების შედეგია:

```
1.2 0.3 12.25 -11.25 3.0 20.5 -10 -5.5 0 2.125 |
jami = 12.625
Press any key to continue . . .
```

3. დაწერეთ პროგრამა, რომელიც კლავიატურიდან შემოტანილ მთელ რიცხვებს შორის იპოვის და დაბეჭდავს უდიდესს.

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
int main(){
  vector<int> v;
  int number;
  while(cin>>number)
     v.push_back(number);
  sort(v.begin(), v.end());
  cout<<"udidesi = "<<v[v.size()-1]<<endl;
  //system("PAUSE");
  return 0;
}
```

პროგრამის შესრულების შედეგია:

```
9 14 0 -3 27 -15 8 9 25 *
udidesi = 27
Press any key to continue . . .
```

4. კლავიატურიდან შემოიტანეთ წინადადება, რომელშიც სიტყვები გამოყოფილია ჰარით. შეტანა დაამთავრეთ ახალ სტრიქონზე Ctrl+z კლავიშების კომბინაციის აკრეფით. დაბეჭდეთ სიტყვების რაოდენობა და ყველა განსხვავებული სიტყვა.

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>
using namespace std;
int main(){
  vector<string> sentence;
  string word;
  while(cin>>word)
     sentence.push_back(word);
  cout<<"Number of words = "<<sentence.size()<<endl;
  sort(sentence.begin(), sentence.end());
  cout<<sentence[0]<<endl;
  int i =1;
  while(i < sentence.size()){
     if(sentence[i-1] != sentence[i])
       cout<<sentence[i]<<endl;
     ++i;
  }
  //system("PAUSE");
  return 0;
}
```

პროგრამის შესრულების შედეგია:

```
a plan a map a dog
^Z
Number of words = 6
a
dog
map
plan
Press any key to continue . . .
```

დამოუკიდებელი სამუშაო:

1. რას დაბეჭდავს შემდეგი ფრაგმენტი? ახსენით და შემდეგ შეამოწმეთ კომპიუტერზე:

№	პროგრამის ფრაგმენტი

1	`int k =1; char p ='A';`
	`  while( k < 4 ){`
	`  cout<<p++<<endl;`
	`  ++k;`
	`  }`

2	`int x =5;`
	`  while ( -25 || 0 ){`
	`  cout<<"ar dasruldeba\n";`
	`  x++;`
	`  }`

3	`int x =5;`
	`  while ( 10-10 && x > 7 ) {`
	`  cout<<"ar shesruldeba\n";`
	`  x++;`
	`  }`

4	`int n =7, s =30;`
	`  while ( true ){`
	`  s -=n;`
	`  n--;`
	`  if(n == 4) break;`
	`  }`
	`cout<<"s = "<<s<<endl;`

5	`int n =2, p =1;`
	`  while( n <= 7 ){`
	`  ++n;`
	`  if( n%2 == 0) continue;`
	`  p *=n;`
	`  }`
	`  cout<<"p ="<<p<<endl;`

2. გაარჩიეთ პროგრამა, რომელიც "temperature.in" ფაილიდან შემოტანილ საშუალო თვიური ტემპერატურის 12 მაჩვენებლის (თვეების) მიხედვით ბეჭდავს წელიწადის საშუალო ტემპერატურას.

```
#include <iostream>
#include <fstream>
using namespace std;
int main(){
  double temps, S =0;
  ifstream fin ("temperature.in");
  while( fin >> temps )
     S += temps;
  cout << "sashualo temperatura = "<< S/12 <<endl;
  //system("PAUSE");
  return 0;
}
```

პროგრამის შესრულების შედეგია:

temperature.in ფაილი	გამოსატანი ეკრანი
12.5 15 14.5 16 18 25.5 35 32.5 34.5 28 20 17	sashualo temperatura = 22.375 Press any key to continue . . .

3. გაარჩიეთ პროგრამა, რომელიც კლავიატურიდან შემოტანილ სტრიქონში ითვლის ასო 'a'-ს რაოდენობას.

```
#include <iostream>
#include <string>
using namespace std;
int main(){
  string Str;
  int counter =0, i =0;
  getline(cin, Str);
  while(i < Str.size()){
     if( Str[i] == 'a' )
       counter++;
     ++i;
  }
  cout<< "striqonSi aris " << counter
      << " aso 'a' "<< endl;
  //system("PAUSE");
  return 0;
}
```

პროგრამის შესრულების შედეგია:

```
a man a cat a dog
striqonSi aris 5 aso 'a'
Press any key to continue . . .
```

დავალება:

ა). შეასრულეთ პროგრამა სხვადასხვა სტრიქონისთვის და დარწმუნდით მიღებული შედეგების სისწორეში;

ბ). პროგრამის `getline(cin, Str);` შეტყობინება შეცვალეთ `cin >> Str;` – ით და ისევ შეასრულეთ პროგრამა. შეგექმნათ პრობლემა?

დაიმახსოვრეთ: თუ სტრიქონი შედგება რამოდენიმე სიტყვისაგან (ანუ შეიცავს სიტყვების გამყოფებს – ჰარის ან/და ტაბულაციის – სიმბოლოებს) კლავიატურიდან მის შესატანად გამოიყენება ფუნქცია `getline` ფორმატით `getline(cin, <სტრიქონის სახელი>)`, რადგან `cin >> <სტრიქონის სახელი>` -ის შემთხვევაში წაიკითხება მხოლოდ პირველი სიტყვა (პირველივე გამყოფ სიმბოლომდე).

4. კლავიატურიდან შეიტანეთ მთელი რიცხვები, ჩაწერეთ სათანადო ვექტორში და შემდეგ დაბეჭდეთ მათი ჯამის მეოთხედი.

5. კლავიატურიდან შეიტანეთ მთელი რიცხვები, ჩაწერეთ სათანადო ვექტორში და შემდეგ დაბეჭდეთ მათი საშუალო არითმეტიკული.

6. კლავიატურიდან შეიტანეთ სიმბოლოთა მიმდევრობა, ჩაწერეთ სათანადო ვექტორში და შემდეგ დაბეჭდეთ შეტანილი სიმბოლოების რაოდენობა.

7. კლავიატურიდან შეიტანეთ 12 მთელი რიცხვი და დაბეჭდეთ მათი საშუალო არითმეტიკული.

8*. კლავიატურიდან შეიტანეთ ტექსტი – წერტილით დამთავრებული რამოდენიმე წინადადება, პირველ წინადადებაში დაადგინეთ: 'a' სიმბოლო გვხვდება მეტად თუ 'b'.

9. კლავიატურიდან შეიტანეთ რიცხვების რაოდენობა – m, შემდეგ m ცალი მთელი რიცხვი და დაბეჭდეთ მათი ნამრავლი.

10. კლავიატურიდან შეიტანეთ მთელი რიცხვი m, შემდეგ დაბეჭდეთ m რიცხვის ფაქტორიალი. რა სახის პრობლემა შეიძლება შეგექმნათ?

ლაბორატორიული მეცადინეობა № 5

ლაბორატორიული სამუშაოს ამოწერა:

- ფაილიდან რიცხვების, სტრიქონების და სიმბოლოების წაკითხვა
- პროგრამის შედეგის ფაილში გამოტანა

ლაბორატორიული სამუშაო:

ამოცანა 1. `data.txt` ფაილი შეიცავს 3 სიმბოლოს. დაბეჭდეთ ეს სიმბოლოები `output.txt` ფაილში შებრუნებული რიგით.

მითითება:

- შექმენით პროექტი სახელით `lp 5.1`, დაამატეთ მასში `5-1.cpp` ფაილი, რომელშიც აკრიფეთ პროგრამა.
- დაამატეთ პროექტში ტექსტური ფაილი `data.txt` შემდეგი გზით: `Project -> Add New Item -> Text File(.txt);` `Name` ველში ჩაწერეთ ფაილის სახელი `data` (გაფართოების გარეშე) და დააჭირეთ `Add` ღილაკს (ან `Enter`-ს).
- ჩაწერეთ `data.txt` ფაილში 3 სიმბოლო.

შესაბამის პროგრამას აქვს სახე:

```
//////////////////////////////////////////////////
//  ავტორი:
//  პროგრამა: ფაილიდან სიმბოლოების წაკითხვა და
//            სხვა ფაილში შებრუნებული რიგით ჩაწერა
//////////////////////////////////////////////////
#include <iostream>
#include <fstream>
using namespace std;
int main(){
   char a, b, c;
   ifstream ifs("data.txt");
   ifs>>a>>b>>c;
   ofstream ofs("output.txt");
   ofs<<c<<' '<<b<<' '<<a;
   return 0;
}
```

პროგრამის გაშვების შემდეგ `lp 5.1` პროექტის შიგა `lp 5.1` საქაღალდეში ავტომატურად შეიქმნება `output.txt` ფაილი. გახსენით იგი (მაგალითად, `File -> Open -> File` და გახსენით `output.txt` ფაილი), გააანალიზეთ პროგრამის მუშაობის შედეგი. მაგალითად,

ფაილი `data.txt`	ფაილი `output.txt`
XYO	O Y X

მითითება: ერთდროულად გახსნილი რამოდენიმე ფაილის შემთხვევაში `CTRL+F6` კლავიშების კომბინაცია ახორციელებს გადართვას მათ შორის.

ამოცანა 2. `monacemebi.in` ფაილში ჩაწერილია 3 მთელი რიცხვი. `pasuxi.out` ფაილში დაბეჭდეთ ამ რიცხვების საშუალო არითმეტიკული.

შესაბამის პროგრამას აქვს სახე:

```
//////////////////////////////////////////////////
//  ავტორი:
//  პროგრამა: ფაილიდან წაკითხული მთელი რიცხვების
//        საშუალო არითმეტიკულის სხვა ფაილში ბეჭდვა
//////////////////////////////////////////////////
#include <iostream>
```

```
#include <fstream>
using namespace std;
int main(){
   int a, b, c;
   double average;
   ifstream fin( "monacemebi.in" );
   fin>>a>>b>>c;
   average =(a+b+c)/3.;
   ofstream fout( "pasuxi.out" );
   fout<<a<<' '<<b<<' '<<c
       <<" ricxvebis sashualo ariTmetikuli = "
       <<average;
   return 0;
}
```

შეასრულეთ პროგრამა რამდენიმეჯერ რიცხვთა სხვადასხვა სამეულებისათვის. შეამოწმეთ pasuxi.out ფაილში მიღებული შედეგები.

ამოცანა 3. input.in ფაილის პირველ სტრიქონში ჩაწერილია სტუდენტის გვარი, მეორეში კი – მისი სახელი. output.out ფაილის პირველ სტრიქონში ჩაწერეთ ამ სტუდენტის სახელი, მეორეში კი გვარი.

შესაბამის პროგრამას აქვს სახე:

```
//////////////////////////////////////////////
// ავტორი:
// პროგრამა: ფაილიდან წაკითხული სტრიქონების
//        შებრუნებული რიგით სხვა ფაილში ბეჭდვა
//////////////////////////////////////////////
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main(){
   string saxeli, gvari;
   ifstream fin("input.in");
   ofstream fout("output.out");
   fin>> gvari >> saxeli;
   fout<< saxeli << endl << gvari;
   fin.close();
   fout.close();
   return 0;
}
```

დავალება: გაარკვიეთ რა მოხდება, თუ input.in ფაილში გვარი და სახელი ჩაწერილია ერთ სტრიქონში, გამოყოფილია ჰარით და გვინდა დავბეჭდოთ გვარი და სახელი output.out ფაილში შებრუნებული რიგით: ა). ერთ სტრიქონად; ბ) ორ სტრიქონად.

ამოცანა 4. ricxvebi.txt ფაილში ჩაწერილია 3 მთელი რიცხვი. shedegi.txt ფაილში გადაიტანეთ მხოლოდ ის რიცხვები, რომლებიც ეკუთვნის [10,30] შუალედს.

შესაბამის პროგრამას აქვს სახე:

```
//////////////////////////////////////////////////////////////////
//  ავტორი:
//  პროგრამა: ფაილის სტრიქონებში მოცემული რიცხვებიდან
//       [10,30] შუალედში მოთავსებული რიცხვების სხვა ფაილში ბეჭდვა
//////////////////////////////////////////////////////////////////
#include <iostream>
#include <fstream>
using namespace std;
```

```
int main(){
   int ricxvi;
   ifstream fin( "ricxvebi.txt" );
   ofstream fout( "shedegi.txt" );
   fin >>ricxvi;
   if(ricxvi >=10 && ricxvi <=30) fout<<ricxvi<<endl;
   fin >>ricxvi;
   if(ricxvi >=10 && ricxvi <=30) fout<<ricxvi<<endl;
   fin >>ricxvi;
   if(ricxvi >=10 && ricxvi <=30) fout<<ricxvi<<endl;
   fin.close();
   fout.close();
   return 0;
}
```

დავალება:

1. შეასრულეთ პროგრამა და გააანალიზეთ shedegi.txt ფაილი.
2. გადაწერეთ პროგრამა განმეორების შეტყობინების გამოყენებით, შეასრულეთ იგი და გააანალიზეთ შედეგი. **მითითება:** პროგრამაში 3 –ჯერ გამეორებული შეტყობინებები

```
fin >>ricxvi;
if(ricxvi >=10 && ricxvi <=30) fout<<ricxvi<<endl;
```

შეცვალეთ შემდეგი ფრაგმენტით:

```
int mTvleli = 1;
while( mTvleli <= 3 ){
   fin >>ricxvi;
   if(ricxvi >=10 && ricxvi <=30) fout<<ricxvi<<endl;
   mTvleli++;
 }
```

დამოუკიდებელი სამუშაო:

1. მოცემული ფაილი შეიცავს 3 სიმბოლოს. დაადგინეთ სიმბოლოთა შორის უმცირესი და დაბეჭდეთ ეს სიმბოლო და მისი კოდი სხვა ფაილში.
2. მოცემული ფაილი შედგება 2 სტრიქონისგან. თითოეულში ჩაწერილია მთელი რიცხვი. მეორე ფაილში ჩაწერეთ ამ რიცხვების
   ა) ჯამი; ბ) ნამრავლი;
   გ) საშუალო გეომეტრიული.
   მითითება: $a_1, a_2, \ldots, a_n$ რიცხვების საშუალო გეომეტრიული გამოითვლება შემდეგი ფორმულით: საშუალო გეომეტრიული $= \sqrt[n]{a_1 \cdot a_2 \cdot \ldots \cdot a_n}$.
3. მოცემული ფაილი შეიცავს სტრიქონს. მეორე ფაილში ჩაწერეთ ამ სტრიქონის სიგრძის მნიშვნელობა.
4. ერთ ფაილში, სახელად weight.in წერია სამი სახის პროდუქტის წონა. მეორე ფაილში, სახელად price.in წერია ამ პროდუქტების ფასები. მესამე ფაილში გამოიტანეთ მოცემული რაოდენობის პროდუქტების შეძენაზე დახარჯული თანხის ოდენობა.

გაარჩიეთ და გააანალიზეთ შესაბამისი C++–პროგრამა:

```
#include <iostream>
#include <fstream>
using namespace std;
int main(){
  double a, b, c =0;
  ifstream ifs1("weight.in");
  ifstream ifs2("price.in");
```

```
  while( ifs1 >> a, ifs2 >> b )
     c += a*b;
  ofstream ofs("answer.out");
  ofs<<c;
  return 0;
}
```

პროგრამის მუშაობის შედეგია:

ფაილი weight.in	ფაილი price.in	ფაილი answer.out
100 50.5 200	3.5 10 7.5	2355

პრაქტიკული მეცადინეობა № 6

პრაქტიკული მეცადინეობის თემები:

- **for** შეტყობინება
- რამდენიმე მარტივი ფუნქცია
- შემთხვევითი რიცხვების გენერირება
- სხვადასხვა ტიპის ვექტორთან მუშაობა

მასალა დამოუკიდებლად გასაცნობისათვის:

1. მაგალითები **for** შეტყობინების გამოყენებაზე:

№	პროგრამის ფრაგმენტი	შედეგი
1	`for(int i=1; i<4; i++)` `  cout << "i = " << i << endl;`	i =1 i =2 i =3
2	`for(int i=3; i>=2; i--)` `  puts("iswavle C++");`	iswavle C++ iswavle C++
3	`for(char c='K'; c>'D'; --c)` `  cout << c << ' ';`	K J I H G F E
4	`for(int n=5; n<=15; n+=5)` `  cout << "n = " << n << endl;`	n =5 n =10 n =15
5	`for(char p='A'; p<'F'; ++p)` `  cout << p << ' ';`	A B C D E

6	`for(int a=5; a>=2; a--){`
	`  if(a%3 == 0) break;`
	`  cout << "a = " << a << endl;`
	`}`
		a =5
		a =4

7	`for(int a=5; a>=2; a--){`
	`  if(a%3 == 0) continue;`
	`  cout << "a = " << a << endl;`
	`}`
		a =5
		a =4
		a =2

8	`for(int i=1; i<50; i++){`
	`  if (i%7 != 0) continue;`
	`  cout << I << ' ';`
	`}`
		7 14 21 28 35 42 49

9	`for(int i=7; i<50; i+=7)` `  cout << I << ' ';`	7 14 21 28 35 42 49
10	`for(int x=10; x>5; x++)` `  puts("usasrulo ganmeoreba");`	პირობა ყოველთვის ჭეშმარიტია: პროცესი არ დასრულდება

11	`for(int k=5; true; k++){`
	`  if(k%2 == 0) continue;`
	`  cout << "k = " << k << endl;`
	`  if(k == 9) break;`
	`}`
		k =5
		k =7
		k =9

12	`for(int n=5; n<3; n++)` `  cout << "n = " << n << endl;`	პირობა თავიდანვე მცდარია: შეტყობინება არ შესრულდება

2. `time(x)` ფუნქციით სარგებლობისათვის საჭიროა `#include <ctime>` ბრძანების პროგრამაში ჩართვა. `time(NULL)` წარმოადგენს სისტემის მიმდინარე კალენდარულ დროს, გაზომილს წამებში. დროის ათვლა მიმდინარეობს გარკვეული თარიღიდან.

#	პროგრამის ფრაგმენტი	შედეგი	გაშვების დრო
1	`cout<<time(NULL)<<endl;`	1319571546	11:39 PM
2	`cout<<time(NULL)<<endl;`	1319571606	11:40 PM
3	`cout<<time(NULL)<<endl;`	1319571906	11:45 PM

3. ფუნქცია `rand()` ახდენს მთელი ფსევდოშემთხვევითი რიცხვის გენერირებას. ყოველი მიმართვისას ფუნქცია აბრუნებს **მთელ** რიცხვს `[0,RAND_MAX]` შუალედიდან. `RAND_MAX`, ჩვეულებრივ, არის `SHRT_MAX –`ის ტოლი.

#	პროგრამის ფრაგმენტი	შედეგი
1	`cout<<rand()<<endl;` `cout<<rand()<<endl;` `cout<<rand()<<endl;`	41 18467 6334

2	`/* იგივე ფრაგმენტი შესრულდა 2 საათის შემდეგ */`
	`cout<<rand()<<endl;`
	`cout<<rand()<<endl;`
	`cout<<rand()<<endl;`
		
		41
		18467
		6334

4. სხვადასხვა ტიპის ვექტორთან მუშაობის მაგალითები:

№	პროგრამის ფრაგმენტი	შედეგი

1	`/* ვექტორში ჩავწეროთ სამი რიცხვი და შემდეგ დავბეჭდოთ ვექტორის ელემენტები */`
	`vector<int> v;`
	`v.push_back(23);`
	`v.push_back(-70);`
	`v.push_back(101);`
	`for(int i=0; i<v.size(); i++ )`
	`  cout << v[i]<< endl;`
		23
		-70
		101

2	`/* კლავიატურიდან შევიტანოთ ორი რიცხვი და ჩავწეროთ ვექტორში */`
	`vector<int> a;`
	`int number;`
	`for(int m=1; m<=2; m++ ){`
	`  cin>>number;`
	`  a.push_back(number);`
	`}`
	`cout<<"first element is "<< a[0] <<endl;`
	`cout<<"second element is "<< a[1]<<endl;`
	`cout<<"vector contains "<< a.size()`
	`  <<" elements\n";`
		`// შეტანა`
		10 -100
		
		`// გამოსატანი ეკრანი`
		first element is 10
		second element is -100
		vector contains 2 elements

3	`/* ვექტორში ჩავწეროთ სამი სიტყვა და შემდეგ დავბეჭდოთ ვექტორის ელემენტები შებრუნებული რიგით*/`
	`vector<string> words;`
	`words.push_back("Mariam");`
	`words.push_back("Giorgi");`
	`words.push_back("Elene");`
	`for(int i=words.size()-1; i>=0; i-- )`
	`  cout << words[i] << endl;`
		Elene
		Giorgi
		Mariam

4	/* ვექტორში ჩავწეროთ სამი ნამდვილი რიცხვი, შემდეგ დავითვალოთ და დავბეჭდოთ მათი ჯამი */
	vector<double> reals;
	reals.push_back(2.25);
	reals.push_back(10.75);
	reals.push_back(-5);
	double sum =0;
	for(int k=0; k<reals.size(); ++k)
	sum += reals[k];
	cout << "jami = " << sum << endl;
		jami = 8

5	/* ვექტორში ჩავწეროთ 5 შემთხვევითი რიცხვი და შემდეგ დავბეჭდოთ ვექტორის ელემენტები*/
	vector<int> vec;
	int number;
	for(int count=1; count<=5; count++){
	number =rand();
	vec.push_back(number);
	}
	for(int i=0; i<vec.size(); i++)
	cout << vec[i] << endl;
		41
		18467
		6334
		26500
		19169

საავტორიო სამუშაო:

1. ვექტორში ჩავწეროთ პირველი 10 ნატურალური რიცხვის კვადრატი და შემდეგ დავბეჭდოთ ეს ვექტორი.

```
#include <iostream>
#include <vector>
using namespace std;
int main(){
  vector<int> vec;
  for(int n=1; n<11; n++)
     vec.push_back(n * n);
  for(int i=0; i<vec.size(); i++)
     cout << vec[i] << '\t';
  cout<<endl;
  system("PAUSE");
  return 0;
}
```

პროგრამის შესრულების შედეგია:

```
1       4       9       16      25      36      49      64      81      100
Press any key to continue . . .
```

2. შექმენით ფუნქცია, რომლის პროტოტიპია `int square(int k);` და რომელიც გამოითვლის ნატურალური რიცხვის კვადრატს. გამოიყენეთ ფუნქცია წინა ამოცანის ამოხსნისთვის.

```
#include <iostream>
#include <vector>
using namespace std;
int square (int );
int main(){
  vector<int> vec;
  for(int n=1; n<11; n++)
     vec.push_back( square (n) );
  for(int i=0; i< vec.size(); i++)
```

```
      cout << vec[i] << '\t';
  cout<<endl;
  system("PAUSE");
  return 0;
}
int square (int k){
   return k * k;
}
```

3. შექმენით კიდევ ერთი ფუნქცია პროტოტიპით `void printVector(vector<int> );` რომელიც დაბეჭდავს ვექტორის ელემენტებს. შემდეგ გამოიყენეთ ეს ფუნქცია წინა ამოცანაში.

```
#include <iostream>
#include <vector>
using namespace std;
int square (int );
void printVector(vector<int> );
int main(){
  vector<int> vec;
  for(int n=1; n<11; n++)
     vec.push_back( square (n) );
  printVector( vec );
  system("PAUSE");
  return 0;
}
int square (int k){
   return k * k;
}
void printVector(vector<int> x){
   for(int i=0; i< x.size(); i++)
     cout << x[i] << '\t';
   cout << endl;
}
```

4. ჩაწერეთ ვექტორში N შემთხვევითი მთელი რიცხვი და დაბეჭდეთ ვექტორი. შემდეგ დაალაგეთ ვექტორის ელემენტები ზრდადობით და ისევ დაბეჭდეთ ვექტორი. პროგრამაში გამოიყენეთ ადრე შექმნილი ფუნქცია `printVector`.

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
void printVector(vector<int> );
int main(){
  vector<int> vec;
  int N;
  cout<<"ShemoitaneT ricxvebis raodenoba : ";
  cin>>N;
  for(int count=1; count<=N; count++)
     vec.push_back(rand());
  printVector( vec );
  sort( vec.begin(), vec.end() );
  printVector( vec );
  system("PAUSE");
  return 0;
}
void printVector(vector<int> x){
```

```
    for(int i=0; i<x.size(); i++)
      cout<<x[i]<<"  ";
    cout<<endl;
}
```

პროგრამის შესრულების შედეგია:

```
ShemoitaneT ricxvebis raodenoba : 5
41  18467  6334  26500  19169
41  6334  18467  19169  26500
Press any key to continue . . .
```

დამოუკიდებელი სამუშაო:

1. გაარჩიეთ პროგრამა, რომელიც ითვლის 310 –იდან 452 –მდე 7–ის ჯერადი რიცხვების ჯამს.

```
#include <iostream>
using namespace std;
int main(){
    int a;          // მთელი რიცხვი
    int s =0;       // 7–ის ჯერადი რიცხვების ჯამი
    //რიცხვებისთვის 310-დან 452-ის ჩათვლით, ბიჯით 1, სრულდება:
    for(a = 310; a <= 452; a++)
       if( a%7 == 0 ) s += a;  //თუ რიცხვი 7-ის ჯერადია დავამატოთ იგი ჯამს
    cout<<"s = "<<s<<endl;
    system("pause");
    return 0;
}
```

პროგრამის შესრულების შედეგია:

```
s =7630
Press any key to continue . . .
```

2. ამოცანის პირობაა: შეკრიბეთ შემთხვევითი რიცხვები, ვიდრე ჯამი არ გახდება 10 –ის ჯერადი, ოღონდ ჯამში არ გაითვალისწინოთ 13–ზე დამთავრებული რიცხვები. გაარჩიეთ შესაბამისი C++ –ის პროგრამა.

```
#include <iostream>
using namespace std;
int main (){
    int a;          // შემთხვევითი რიცხვი
    int s =0;       // რიცხვების ჯამი
    while ( true ){
       a =rand();
       if(a%100 == 13)
          continue;
       s +=a;
       if(s%10 == 0)
          break;
    }
    cout<< "s = " << s <<'\n';
    system("PAUSE");
    return 0;
}
```

პროგრამის შესრულების შედეგია:

```
s =554810
Press any key to continue . . .
```

3. ჩაწერეთ სათანადო ვექტორში კლავიატურიდან შემოტანილი 5 სიტყვა და შემდეგ დაბეჭდეთ ვექტორის ელემენტები. ვექტორის ელემენტების დასაბეჭდათ შექმენით და გამოიყენეთ ფუნქცია პროტოტიპით **void** printWords(vector<string>);

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;
void printWords(vector<string> );
int main(){
  vector<string> W;
  string word;
  for(int count=1; count<=5; count++){
     cin>>word;
     W.push_back(word);
  }
  printWords(W);
  system("PAUSE");
  return 0;
}
void printWords(vector<string> a){
   for(int i=0; i<a.size(); i++ )
      cout<<a[i]<<endl;
}
```

პროგრამის შესრულების შედეგია:

```
Tbilisi Batumi KuTaisi Gori Telavi
Tbilisi
Batumi
KuTaisi
Gori
Telavi
Press any key to continue . . .
```

4. რას დაბეჭდავს შემდეგი ფრაგმენტი? ახსენით და შემდეგ შეამოწმეთ კომპიუტერზე:

№	პროგრამის ფრაგმენტი

1	`int k; char p;`
	`for(k=1, p='A'; k<9; ++k)`
	`  cout << p++ << endl;`

2	`int x;`
	`for(x=10; -10 || 0; x++)`
	`  puts("ar dasruldeba");`

3	`int x;`
	`for(x=10; x>3 && 5-5; x++)`
	`  puts("ar shesruldeba");`

4	`int s =30;`
	`for(int n =7; true; --n){`
	`  s -=n;`
	`  if(n == 4) break;`
	`}`
	`cout << "s = " << s << endl;`

5	`int n =5, p =1;`
	`for(  ; n<=11;  n++){`
	`  if( n%2 == 0) continue;`
	`  p *=n;`
	`}`
	`cout << "p = " << p << endl;`

5. კლავიატურიდან შევიტანოთ მთელი რიცხვები, ჩავწეროთ სათანადო ტიპის ვექტორში და შემდეგ დავბეჭდოთ მათი ჯამის მეოთხედი.

6. კლავიატურიდან შევიტანოთ მთელი რიცხვები, ჩავწეროთ სათანადო ტიპის ვექტორში და შემდეგ დავბეჭდოთ მათი საშუალო არითმეტიკული.

7. კლავიატურიდან შეიტანეთ სიმბოლოთა მიმდევრობა, ჩაწერეთ სათანადო ტიპის ვექტორში და შემდეგ დაბეჭდეთ შეტანილი სიმბოლოების რაოდენობა.

8. იპოვეთ და დაბეჭდეთ 7–დან 32–მდე რიცხვების ჯამი.

9. იპოვეთ 14 –იდან 102 –მდე კენტი რიცხვების ჯამი.

10. იპოვეთ 4 –იდან 123 –მდე ლუწი რიცხვების ჯამი.

11. იპოვეთ 5 –იდან 103 –მდე 5–ის ჯერადი რიცხვების ჯამის მეოთხედი.

12. იპოვეთ m-იდან n-მდე ნატურალურ რიცხვთა ჯამი. m და n შემოიტანეთ კლავიატურიდან.

13. a და b – მთელი რიცხვებია. c ცვლადს მიანიჭეთ a–დან b–მდე რიცხვების საშუალო არითმეტიკული, და შემდეგ დაბეჭდეთ. a და b შეიტანეთ კლავიატურიდან.

14. კლავიატურიდან შევიტანოთ 12 მთელი რიცხვი და დავბეჭდოთ მათი საშუალო არითმეტიკული.

15. იპოვეთ შემთხვევითი რიცხვების, გარდა ლუწი რიცხვებისა, საშუალო არითმეტიკული ვიდრე მათი ჯამი არ გახდება 21 –ის ჯერადი.

16. იპოვეთ შემთხვევითი რიცხვების, გარდა 5–ის ჯერადი რიცხვებისა, საშუალო არითმეტიკული ვიდრე მათი ჯამი არ გახდება 19 –ის ჯერადი.

17. რამდენჯერმე დაბეჭდეთ `cout << (1.0*rand())/RAND_MAX << endl;` რა ტიპის შემთვევით რიცხვებს ქმნის `(1.0*rand())/RAND_MAX` ? რომელი დიაპაზონიდან?

18*. წინა სავარჯიშოს საფუძველზე, როგორ შევქმნათ შემთხვევითი ნამდვილი რიცხვები დიაპაზონიდან `[a.b]`?

ლაბორატორიული მეცადინეობა № 6

ლაბორატორიული სამუშაოს ამოწერა:

- `rand()` და `time()` ფუნქციების გამოყენება
- შემთხვევითი რიცხვების შექმნა და ბეჭდვა

ლაბორატორიული სამუშაო:

ამოცანა 1. დაბეჭდეთ სამი მთელი შემთხვევითი რიცხვი `[0;100]` შუალედიდან.

შესაბამის პროგრამას აქვს სახე:

```
//////////////////////////////////////////
// ავტორი:
// პროგრამა: სამი შემთხვევითი რიცხვის [0;100]
//                შუალედიდან ბეჭდვა
//////////////////////////////////////////
#include <iostream>
using namespace std;
int main(){
  cout<<"random numbers:\n";
  cout<<rand()%101<<' '<<rand()%101<<' '
      <<rand()%101<<endl;
  system("PAUSE");
  return 0;
}
```

დავალება:

გადაწერეთ პროგრამა ისე, რომ შესაძლებელი გახდეს N ცალი შემთხვევითი რიცხვის `[0;100]` შუალედიდან დაბეჭდვა, თუ N–ის მნიშვნელობას შევიტანთ კლავიატურიდან.

ამოცანა 2. დაწერეთ პროგრამა, რომელიც ბეჭდავს მიმდინარე დროს წამებში და წუთებში..

შესაბამის პროგრამას აქვს სახე:

```
//////////////////////////////////////
// ავტორი:
// პროგრამა: მიმდინარე დროის წამებში და
//               წუთებში ბეჭდვა
//////////////////////////////////////
#include <iostream>
#include <ctime>
using namespace std;
int main(){
  cout<<"mimdinare dro (second): "<<time(NULL)<<endl;
  cout<<"mimdinare dro (minute): "<<time(NULL)/60<<endl;
  system("PAUSE");
  return 0;
}
```

დავალება:

დაამატეთ პროგრამაში შეტყობინება, რომელიც დაბეჭდავს მიმდინარე გროს საათებში.

ამოცანა 3. შექმენით 10 შემთხვევითი მთელი რიცხვი `[65,90]` დიაპაზონიდან – დიდი ინგლისური ასოების კოდები. დაბეჭდეთ ამ კოდების შესაბამისი სიმბოლოები. დასაბეჭდად გამოიყენეთ თქვენ მიერ განსაზღვრული ფუნქცია. შემდეგ დაბეჭდეთ სიმბოლოების მორის უმცირესი და მისი კოდი.

შესაბამის პროგრამას აქვს სახე:

```
////////////////////////////////////////////
// ავტორი:
// პროგრამა: 10 შემთხვევითი რიცხვის შექმნა და მათი
//              შესაბამისი სიმბოლოების ბეჭდვა
////////////////////////////////////////////
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
void printChar(vector<int> );
int main(){
  vector<int> vec;
  int code;
  for(int count=1; count<=10; count++){
     code =rand()%(91 -65) + 65;
     vec.push_back(code);
  }
  printChar(vec);
  sort( vec.begin(), vec.end() );
  printChar(vec);
  cout<<"umciresia "<<(char)vec[0]
      <<" kodiT "<<vec[0]<<endl;
  system("PAUSE");
  return 0;
}
void printChar(vector<int> m){
   for(int i=0; i<m.size(); i++ )
      cout<<(char)m[i]<<' ';
   cout<<endl;
}
```

დავალება:

შეცვალეთ პროგრამა ისე, რომ მან დაბეჭდოს შემთხვევითი კოდების მქონე სიმბოლოები დალაგებული კლებადობით. **მითითება:**

```
sort( vec.begin(), vec.end() );
```

ალგორითმის ნაცვლად გამოიყენეთ მისი სხვა სახე:

```
sort( vec.rbegin(), vec.rend() );
```

შესაბამისად შეცვალეთ უმცირები სიმბოლოს და მისი კოდის ბეჭდვის შეტყობინებაც.

ამოცანა 4. დაწერეთ პროგრამა, რომელიც შექმნის 25 შემთხვევით პატარა ასოს (['a','z'] შუალედიდან) და შემდეგ დაბეჭდავს უდიდესის კოდს. **მითითება:** ისარგებლეთ ფორმულით:

```
rand()%('z'+1- 'a') + 'a';
```

დამოუკიდებელი სამუშაო:

1. მოცემული ფაილი შეიცავს 3 სიმბოლოს. სიმბოლოთა შორის უმცირესი და უდიდესი მათ კოდებთან ერთად დაბეჭდეთ სხვა ფაილში. ამოცანა გაანალიზეთ ორი ხერხით: 1) ვექტორის და დალაგების ალგორითმის გამოყენების გარეშე; 2) სათანადო ტიპის ვექტორის და დალაგების ალგორითმის გამოყენებით.
2. 75 შემთხვევითი რიცხვი დიაპაზონში [-15,80] დაბეჭდეთ კლებადობით.
3. კლავიატურიდან შემოიტანეთ N რაოდენობის ნამდვილი რიცხვი. დაბეჭდეთ დადებით რიცხვების შორის უმცირესი და მისი რიგითი ნომერი.

პრაქტიკული მეცადინეობა № 7

პრაქტიკული მეცადინეობის თემები:

- ფუნქციის პარამეტრები, დასაბრუნებელი მნიშვნელობა
- ფუნქციის პროტოტიპი, განსაზღვრა
- ფუნქციის გამოძახება

მასალა დამოუკიდებლად გასაცნობისათვის:

1. მარტივი ფუნქციების და ძირითად პროგრამაში მათი გამოძახების ნიმუშები.
   მითითება: იმისათვის რომ ვნახოთ შექმნილი ფუნქციის მუშაობის შედეგი, მოყვანილი ფრაგმენტები უნდა შევავსოთ სრულ პროგრამამდე. მაგალითად, პირველი ფრაგმენტისთვის უნდა დავწეროთ:

```
#include <iostream>
using namespace std;
int my_abs(int );
int main(){
      int a, n;
      cin>>a;
      n = my_abs(a);
      cout<<n<<endl;
   system("pause");
   return 0;
}
   int my_abs(int x){
       if(x >= 0) return x;
       return -x;
   }
```

ხოლო სხვა ნიმუშებში შევცვალოთ ფუნქციის სათაური (პროტოტიპი), main()-ის ფრაგმენტი და ფუნქციის განსაზღვრა შესაბამისად.

№	ფუნქცია	ფუნქციის გამოძახება – main()-ის ფრაგმენტი

1	/*ფუნქცია პოულობს მთელი რიცხვის მოდულს*/
	int my_abs(int x){
	if(x >= 0) return x;
	return -x;
	}
		int main(){
		int a, n;
		cin>>a;
		n = my_abs(a);
		cout<<n<<endl;
		. . .
		}

2	/* ფუნქცია პოულობს ნამდვილი რიცხვის მოდულს */
	float my_fabs(float x){
	if(x < 0) return -x;
	return x;
	}
		int main(){
		float y, k;
		cin>>y;
		k = my_fabs(y);
		cout<<k<<endl;
		. . .
		}

3	/* ფუნქცია ადგენს რიცხვის ნიშანს */
	int my_sign(float x){
	if(x > 0) return 1;
	else if(x == 0) return 0;
	return -1;
	}
		int main(){
		float n; int s;
		cin>>n;
		s = my_sign(n);
		cout<<s<<endl;
		. . .
		}

4	/* ფუნქცია პოულობს 2 მთელი რიცხვიდან	int main(){ int m, n, max;

	უდიდესს */
	int my_max2(int a, int b){
	  if(a > b) return a;
	  return b;
	}
		cin>>m>>n;
		  max = my_max2(m,n);
		  cout<<max<<endl;
		  . . .
		}

5	/* ფუნქცია პოულობს 2 ნამდვილი რიცხვიდან უმცირესს */
	float my_fmin2(float x, float y){
	  return (x < y) ? x : y;
	}
		int main(){
		  float a, b, min;
		  cin>>a>>b;
		  min = my_fmin2(a,b);
		  cout<<min<<endl;
		  . . .
		}

6	/* ფუნქცია პოულობს 3 მთელი რიცხვიდან უმცირესს */
	int my_min3(int a, int b, int c){
	  int m = a;
	  if(b < m) m = b;
	  if(c < m) m = c;
	  return m;
	}
		int main(){
		  int k, n, m, min;
		  cin>>k>>n>>m;
		  min = my_min3(k,n,m);
		  cout<<min<<endl;
		  . . .
		}

7	/* ფუნქცია პოულობს 3 ნამდვილი რიცხვიდან უდიდესს */
	float my_fmax3(float a,float b,float c){
	  float m = a;
	  if(b > m) m = b;
	  if(c > m) m = c;
	  return m;
	}
		int main(){
		  float x, y, z, max;
		  cin>>x>>y>>z;
		  max = my_fmax3(x,y,z);
		  cout<<max<<endl;
		  . . .
		}

8	/* ფუნქცია ითვლის მთელი რიცხვის კვადრატს */
	int square(int a){
	  return a * a;
	}
		int main(){
		  int k;
		  cin>>k;
		  cout<<square(k)<<endl;
		  . . .
		}

9	/* ფუნქცია ითვლის ორი მთელი რიცხვის საშუალოს */
	double average(int n, int m){
	  return (n + m)/2.;
	}
		int main (){
		  int a, b; double rez;
		  cin>>a>>b;
		  rez = average(a,b);
		  cout<<rez<<endl;
		  . . .
		}

11	/* ფუნქცია პოულობს შემთხვევით რიცხვს [A,B] შუალედიდან */
	
	int my_rand(int A, int B){
	  return rand()%(B + 1 - A) + A;
	}
		int main(){
		  int A, B, number;
		  cin>>A>>B;
		  number = my_rand(A, B);
		  cout<<number<<endl;
		  . . .
		}

12	/* ფუნქცია ითვლის x რიცხვის მთელ n ხარისხს, $x \neq 0$, $n \geq 0$ */
	float my_pow(float x, int n){
	// precondition: n >= 0, x != 0
	  float d = 1; int i;
	  if(n == 0) return 1;
	  else{
		int main(){
		  float x, degree;
		  int n;
		  cin>>x>>n;
		  degree = my_pow(x, n);
		  cout<<degree<<endl;
		  . . .

	``
	    for(i = 1; i <= n; i++)
	       d *= x;
	    return d;
	  }
	}
	``
		``
		}
		``

2. მარტივი bool ტიპის ფუნქციები (ფუნქცია–პრედიკატები) და ძირითად პროგრამაში მათი გამოძახების მაგალითები

№	ფუნქცია	ფუნქციის გამოძახება

1	```
	/* ფუნქცია ადგენს არის თუ არა მთელი N
	      რიცხვი 3 –ის ჯერადი */
	int Multiple_3(int N){
	   if(N%3 == 0) return 1;
	   return 0;
	}
	``
		``
		int main(){
		  int number, k;
		  cin>>number;
		  k = Multiple_3(number);
		  if(k == 1)
		    cout<<number<<" aris 3-is"
		          " jeradi\n";
		  else cout<<number<<" ar aris"
		          " 3-is jeradi\n";
		  . . .
		}
		``

2	```
	/* ფუნქცია ადგენს არის თუ არა მთელი A
	          რიცხვი ლუწი */
	int IsEven(int N){
	   return N%2 == 0;
	}
	``
		``
		int main(){
		  int num;
		  cin>>num;
		  if(IsEven(num))
		      cout<<num<<" is even\n";
		  else cout<<num<<" is't"
		         " even\n";
		  . . .
		}
		``

3	```
	/* ფუნქცია ადგენს არის თუ არა ნამდვილი
	      D რიცხვი უარყოფითი */
	bool IsNegative(float D){
	   if(D < 0)
	      return true;
	   return false;
	}
	``
		``
		int main(){
		  float num;
		  cin>>num;
		  if(IsNegative(num))
		   cout<<num<<" is negative\n";
		  else cout<<num<<" is't"
		         " negative\n";
		  . . .
		}
		``

4	```
	/* ფუნქცია ადგენს არის თუ არა მთელი N
	  რიცხვი დადებითი და 5 –ის ჯერადი */
	bool func(int N){
	   if(N > 0 && N%5 == 0)
	     return true;
	   return false;
	}
	``
		``
		int main(){
		  int number;
		  bool answer;
		  cin>>number;
		  answer = func(number);
		  if(answer)
		     puts("Yes");
		  else puts("No");
		  . . .
		}
		``

5	```
	  /* ფუნქცია ადგენს არის თუ არა მისი
	     პარამეტრი სიმბოლო–ციფრი */
	bool IsDigit(char p){
	   if(p >= '0' && p <= '9')
	      return true;
	   return false;
	}
	``
		``
		int main(){
		  char ch;
		  cin>>ch;
		  if(IsDigit(ch))
		    cout<<ch<<" aris "
		         "simbolo-cifri\n";
		  else cout<<ch<<" ar aris "
		         "simbolo-cifri\n";
		``

		. . . }

6	/* ფუნქცია ადგენს არის თუ არა მისი პარამეტრი დიდი ასო */
	bool IsUpperLetter(char p){
	if(p >= 'A' && p <= 'Z')
	return true;
	return false;
	}
		int main(){
		char sym;
		cin>>sym;
		if(IsUpperLetter(sym))
		cout<<sym<<" aris"
		" didi aso\n";
		else cout<<sym<<" ar aris"
		" didi aso\n";
		. . .
		}

საუდიტორიო სამუშაო:

1. დაწერეთ ფუნქცია, რომელიც გამოითვლის გამოსახულებას $2x^3+3x^2-5x-10$, სადაც x მთელი რიცხვია. გამოიძახეთ ფუნქცია ძირითად პროგრამაში კლავიატურიდან შემოტანილი მთელი რიცხვისთვის და შედეგი დაბეჭდეთ.

```
#include <iostream>
using namespace std;
int func(int x);   // ფუნქციის პროტოტიპი (ფუნქციაზე განაცხადი)
int main(){
  int a, rezult;
  cout<<"ShemoitaneT mTeli ricxvi  ";
  cin>>a;
  rezult = func(a);   // ფუნქციის გამოძახება
  cout<<"Gamosaxuleba = "<<rezult<<endl;
  //system("PAUSE");
  return 0;
}
// ფუნქციის განსაზღვრა
int func(int x){
  return 2*x*x*x+3*x*x-5*x-10;
}
```

პროგრამა დაბეჭდავს:

```
ShemoitaneT mTeli ricxvi  5
Gamosaxuleba = 290
Press any key to continue . . .
```

2. შექმენით ფუნქცია პროტოტიპით `bool IsLetter(char p)`, რომელიც დაადგენს არის თუ არა მისი p პარამეტრი ასო. გამოიძახეთ ფუნქცია ძირითად პროგრამაში კლავიატურიდან შემოტანილი სიმბოლოსთვის და შედეგი დაბეჭდეთ.

```
#include <iostream>
using namespace std;
bool IsLetter(char p);
int main(){
  char sym;
  cout<<"ShemoitaneT simbolo : ";
  cin>>sym;
  if( IsLetter(sym) )
     cout<<sym<<" aris aso\n";
  else cout<<sym<<" ar aris aso\n";
  //system("PAUSE");
```

```
  return 0;
}
bool IsLetter(char p){
  if(p >= 'a' && p <= 'z'
       || p >= 'A' && p <= 'Z')
    return true;
  return false;
}
```

პროგრამა დაბეჭდავს:

```
ShemoitaneT simbolo : R
R aris aso
Press any key to continue . . .
```

3. numbers.dat ფაილში ჩაწერილია მთელი რიცხვები. დაითვალეთ, რამდენი კხრიანია ფაილში.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
using namespace std;
int main(){
  int number;
  vector<int> N;
  ifstream fin("numbers.dat");
  while(fin>>number)
     N.push_back(number);
  int quantity;
  quantity = count( N.begin(), N.end(), 9);
  cout<<quantity<<endl;
  //system("PAUSE");
  return 0;
}
```

პროგრამის შესრულების შედეგია:

numbers.dat ფაილი	გამოსატანი ეკრანი
1 2 3 4 9 6 7 8 9 10 11 12 13 9 15	3 Press any key to continue . . .

4. numbers.dat ფაილში ჩაწერილია მთელი რიცხვები. დაითვალეთ, მათგან რამდენია კენტი.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
using namespace std;
bool isOdd(int x)       // ფუნქცია-პრედიკატზე განაცხადი
{                       // და მისი განსაზღვრა
   return x % 2 != 0;
}
int main(){
  int number;
  vector<int> N;
  ifstream fin("number.dat");
  while(fin>>number)
```

```
    N.push_back(number);
  int quantity;
  // აქ გამოიყენება ფუნქცია isOdd
  quantity = count_if( N.begin(), N.end(), isOdd);
  cout<<quantity<<endl;
  //system("PAUSE");
  return 0;
}
```

პროგრამის შესრულების შედეგია:

numbers.dat ფაილი	გამოსატანი ეკრანი
1 2 3 4 9 6 7 8 9 10 11 12 13 9 15	9 Press any key to continue . . .

დამოუკიდებელი სამუშაო:

1. დაწერეთ ფუნქცია, რომელიც გამოითვლის გამოსახულებას:
   ა) $x^3 - 4x^2 + 5x - 7$;
   ბ) $0.5x^2 + 2.5x + 12$.
   x მთელი რიცხვია. გამოიძახეთ ფუნქცია ძირითად პროგრამაში კლავიატურიდან შემოტანილი მთელი რიცხვისთვის და შედეგი დაბეჭდეთ.

2. მოცემულია პირობა $f(x) = \begin{cases} 2x + 34, & \text{თუ } x \geq 10, \\ x^2 - x - 1, & \text{წინააღმდეგ შემთხვევაში.} \end{cases}$

   დაწერეთ ფუნქცია, რომელიც გამოითვლის $f(x)$-ის მნიშვნელობას მთელი x რიცხვისთვის. გამოიძახეთ ფუნქცია ძირითად პროგრამაში კლავიატურიდან შემოტანილი მთელი რიცხვისთვის და შედეგი დაბეჭდეთ.

   გაარჩიეთ ამ ამოცანის შესაბამისი პროგრამა:

```
#include <iostream>
using namespace std;
int function(int x);
int main(){
  int a, rezult;
  cout<<"ShemoitaneT mTeli ricxvi  ";
  cin>>a;
  rezult =function(a);
  cout << "Gamosaxuleba = " << rezult << '\n';
  //system("PAUSE");
  return 0;
}
int function(int x){
  if(x >= 10) return 2*x+34;
  return x*x-x-1;
}
```

პროგრამის შესრულების შედეგია:

```
ShemoitaneT mTeli ricxvi  7
Gamosaxuleba = 41
Press any key to continue . . .
```

დავალება:

ა) შეასრულეთ პროგრამა კომპიუტერზე სხვადასხვა მთელი a რიცხვისთვის;
ბ) ახსენით function ფუნქციის ალგორითმი.

3. მოცემულია პირობა:

ა) $f(x)=\begin{cases} 5x^3-12, & \text{თუ } x>3, \\ x^2+2x+4, & \text{წინააღმდეგ შემთხვევაში.} \end{cases}$;

ბ) $f(x)=\begin{cases} 0.2x+3.45, & \text{თუ } x\geq 10, \\ x^2-x-1, & \text{წინააღმდეგ შემთხვევაში.} \end{cases}$;

გ) $f(x)=\begin{cases} 7x^2+3, & \text{თუ } x\geq 1, \\ x-25, & \text{თუ } x=0, \\ x^3+6x+8, & \text{თუ } x\leq -1. \end{cases}$;

დ) $f(x)=\begin{cases} 11.4x-0.3, & \text{თუ } x>5, \\ x^2-2.5x+9, & \text{თუ } x=5, \\ x^3-8, & \text{თუ } x<5. \end{cases}$.

დაწერეთ ფუნქცია, რომელიც გამოითვლის $f(x)$-ის მნიშვნელობას მთელი x რიცხვისთვის. გამოიძახეთ ფუნქცია ძირითად პროგრამაში კლავიატურიდან შემოტანილი მთელი რიცხვისთვის და შედეგი დაბეჭდეთ.

4. მოცემულია პირობა:

ა) $f(x)=\begin{cases} 100.5x^3+50, & \text{თუ } x\leq 20.5, \\ 20x-45.5, & \text{წინააღმდეგ შემთხვევაში.} \end{cases}$;

ბ) $f(x)=\begin{cases} 15x^2+30x+20, & \text{თუ } x\geq 10.5, \\ x^3-40x+50, & \text{თუ } x=10.5, \\ x+25.5, & \text{თუ } x\leq 10.5. \end{cases}$.

დაწერეთ ფუნქცია, რომელიც გამოითვლის $f(x)$-ის მნიშვნელობას ნამდვილი x რიცხვისთვის. გამოიძახეთ ფუნქცია ძირითად პროგრამაში კლავიატურიდან შემოტანილი ნამდვილი რიცხვისთვის და შედეგი დაბეჭდეთ.

5. დაწერეთ ფუნქცია, რომელიც დაადგენს:
   ა) არის თუ არა მთელი N რიცხვი უარყოფითი. N ფუნქციის პარამეტრია;
   ბ) არის თუ არა მთელი K რიცხვი 5-ის ან 7-ის ჯერადი. K ფუნქციის პარამეტრია;
   გ) არის თუ არა ნამდვილი M რიცხვი 10-ზე ნაკლები. M ფუნქციის პარამეტრია;
   დ) ბოლოვდება თუ არა მთელი A რიცხვი 3-ით ან 9-ით. A ფუნქციის პარამეტრია;
   ე) წარმოადგენს თუ არა მისი პარამეტრი – სიმბოლო B – პატარა ასოს;
   ვ) არის თუ არა მთელი M რიცხვი სამნიშნა. M ფუნქციის პარამეტრია.

6. დაწერეთ ფუნქცია, რომელიც იპოვის:
   ა) N-დან M-მდე ჩათვლით კენტი რიცხვების საშუალო არითმეტიკულს. N და M ფუნქციის პარამეტრებია;
   ბ) C-დან D-მდე ჩათვლით ლუწი რიცხვების რაოდენობას. C და D ფუნქციის პარამეტრებია;
   გ) a-დან b-მდე ჩათვლით 7-ის ჯერადი რიცხვების ჯამს. a და b ფუნქციის პარამეტრებია;
   დ) M-დან K-მდე ჩათვლით 3-ის ჯერადი რიცხვების საშუალო არითმეტიკულს. M და K ფუნქციის პარამეტრებია.

7. შექმენით ფუნქცია, რომელიც:
   ა) გამოითვლის სამი მთელი რიცხვის საშუალო არითმეტიკულს. გამოიყენეთ ფუნქცია ძირითად პროგრამაში და დაბეჭდეთ მიღებული შედეგი;

ბ) დააბრუნებს მთელი რიცხვის კუბს. ეს რიცხვი ფუნქციის პარამეტრს წარმოადგენს. გამოიყენეთ ფუნქცია main-ში და დაბეჭდეთ 1-დან 100-მდე რიცხვთა კუბები;

გ) გამოითვლის მართკუთხედის ფართობს (მართკუთხედის სიგრძე და სიგანე – ფუნქციის პარამეტრებია). გამოიყენეთ ფუნქცია ძირითად პროგრამაში და შეადარეთ 2 მართკუთხედის ფართობი;

დ) დააბრუნებს 2 მთელ რიცხვს შორის უმცირესს. რიცხვები ფუნქციის პარამეტრებს წარმოადგენენ. გამოიყენეთ ფუნქცია ძირითად პროგრამაში და იპოვეთ 4 მთელ რიცხვს შორის უმცირესი;

ე) გამოითვლის სამკუთხედის ფართობს 3 გვერდის მიხედვით. გამოიყენეთ ფუნქცია ძირითად პროგრამაში და შეადარეთ 2 სამკუთხედის ფართობი;

ვ) დააბრუნებს 1-ს, თუ მისი პარამეტრი-სიმბოლო არის ხმოვანი ასო, წინააღმდეგ შემთხვევაში დააბრუნებს 0-ს. ძირითად პროგრამაში გამოიძახეთ ფუნქცია სხვადასხვა სიმბოლოებისათვის;

ზ) პარამეტრად მიიღებს სიმბოლოს, და თუ ეს სიმბოლო-ციფრია – დააბრუნებს მის კოდს, წინააღმდეგ შემთხვევაში კი დააბრუნებს –1-ს. გამოიყენეთ ფუნქცია ფაილში ჩაწერილი სიმბოლოების მიმდევრობისათვის.

8. გაარჩიეთ პროგრამა, რომელიც ხსნის შემდეგ ამოცანას:

უმცირესი მთელი რიცხვი, რომელიც მეტია $\log_2 N$ -ზე, წარმოადგენს ბიტების რაოდენობას მთელი N რიცხვის ორობით ჩანაწერში. დაწერეთ ფუნქცია, რომელიც გამოითვლის $\log_2 N$ -ზე მეტ უმცირეს მთელ რიცხვს. გამოიყენეთ ფუნქცია მთელი N რიცხვის ორობით წარმოდგენაში ბიტების რაოდენობის განსაზღვრისთვის.

```
#include <iostream>
using namespace std;
int my_log2(int N);
int main(){
  int N, degree;
  cin >> N;
  degree = my_log2(N);
  cout << degree << endl;
  system("PAUSE");
  return 0;
}
int my_log2(int N){
   int d;
   if(N < 0) N = -N;
   for(d =0; N > 0; d++, N /= 2);
   return d;
}
```

პროგრამის შესრულების შედეგია:

```
9
4
Press any key to continue . . .
```

დავალება:

ა) შეასრულეთ პროგრამა კომპიუტერზე სხვადასხვა მთელი N რიცხვისთვის;

ბ) ახსენით my_log2 ფუნქციის ალგორითმი.

9. დაწერეთ ფუნქცია, რომელიც ითვლის:

ა) $1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{20}$ ჯამს; ბ) $1+\frac{1}{3}+\frac{1}{5}+\cdots+\frac{1}{n}$ ჯამს, სადაც კენტი n რიცხვი ფუნქციის პარამეტრია.

ლაბორატორიული მეცადინეობა № 7

ლაბორატორიული სამუშაოს ამოცანა:

- ორ ცვლადს შორის უდიდესის დამდგენს ფუნქციის შექმნა
- ფუნქციის შექმნა, რომელიც დაადგენს, წარმოადგენს თუ არა სტრიქონი პალინდრომს

ლაბორატორიული სამუშაო:

ამოცანა 1: შექმენით ფუნქცია, რომელიც იპოვის 2 მთელი რიცხვიდან უდიდესს. ეს რიცხვები ფუნქციის პარამეტრებია. გამოიყენეთ ფუნქცია main–ში კლავიატურიდან შემოტანილი 2 მთელი რიცხვისთვის და დაბეჭდეთ მათ შორის უდიდესი.

პროგრამის სახე შემდეგია:

```
//////////////////////////////////////////
// პროგრამა: 2 მთელ რიცხვს შორის უდიდესის
//                პოვნა
//////////////////////////////////////////
#include <iostream>
using namespace std;
int max2Int(int , int );
int main(){
  int a, b;
  cout << "SemoitaneT 2 mTeli ricxvi : ";
  cin >> a >> b;
  int max = max2Int(a, b);
  cout << "udidesia " << max << endl;
  system("PAUSE");
  return 0;
}
int max2Int(int x, int y){
  return x > y ? x : y;
}
```

დავალება: შეასრულეთ პროგრამა სხვადასხვა a და b რიცხვებისთვის.

ამოცანა 2: შექმენით ფუნქცია, რომელიც იპოვის 2 სტრიქონიდან უდიდესს. ეს სტრიქონები ფუნქციის პარამეტრებია. გამოიყენეთ ფუნქცია main–ში კლავიატურიდან შემოტანილი 2 სტრიქონისთვის და დაბეჭდეთ მათ შორის უდიდესი.

პროგრამას აქვს შემდეგი სახე:

```
//////////////////////////////////////////
// პროგრამა: 2 სტრიქონს შორის უდიდესის
//                პოვნა
//////////////////////////////////////////
#include <iostream>
#include <string>
using namespace std;
string max2String(string , string );
int main(){
  string a, b;
  cout << "SemoitaneT 2 striqoni : ";
  cin >> a >> b;
  string max = max2String(a, b);
  cout << "udidesia " << max << endl;
  system("PAUSE");
  return 0;
}
string max2String(string x, string y){
```

```
  return x > y ? x : y;
}
```

დავალება: შეასრულეთ პროგრამა სხვადასხვა a და b სტრიქონებისთვის.

ამოცანა 3: დაწერეთ ფუნქცია, რომლის პარამეტრი არის სტრიქონი. ფუნქციამ უნდა დაადგინოს, წარმოადგენს თუ არა იგი პალინდრომს. სტრიქონი არის პალინდრომი, თუ ის ერთნაირად იკითხება როგორც მარჯვნიდან მარცხნივ, ისე მარცხნიდან მარჯვნივ. მაგალითად, ANNA – სტრიქონი-პალინდრომია. გამოიყენეთ ფუნქცია კლავიატურიდან შეტანილი სტრიქონისთვის და დაბეჭდეთ შედეგი.

პროგრამის სახე შემდეგია:

```
//////////////////////////////////////////
// პროგრამა: სტრიქონსთვის დადგენა, არის თუ
//              არა იგი პალინდრომი
//////////////////////////////////////////
#include <iostream>
#include <string>
using namespace std;
bool Palindrom(string );
int main(){
  string S; bool p;
  cout << "Enter string\n";
  getline(cin, S);
  p = Palindrom(S);
  if( p ) cout << "Is Palindrom" << endl;
  else  cout << "Is't Palindrom" << endl;
  system("pause");
  return 0;
}
bool Palindrom(string K){
   int i, len;
   len = K.size();
   for(i=0; i<len/2; i++)
     if(K[i] != K[len-1-i])
        return false;
   return true;
}
```

დავალება:

ა) შეასრულეთ პროგრამა სხვადასხვა სტრიქონებისათვის, მაგალითად, ai ia, level, Ai Ia.

ბ) ახსენით Palindrom ფუნქციის ალგორითმი.

ამოცანა 4: ქვემოთ მოყვანილ პროგრამაში S სტრიქონის შებრუნებისთვის გამოიყენება C++ -ის სტანდარტული string კლასის ალგორითმი reverse:

```
//////////////////////////////////////////
// პროგრამა: სტრიქონის შებრუნება
//////////////////////////////////////////
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;
int main(){
  string S ="Bjarne Stroustrup";
  reverse(S.begin(), S.end());
```

```
  cout << S << endl;
  //system("pause");
  return 0;
}
```

დავალება:

ა) გაეცანით კოდს, შეასრულეთ პროგრამა S სტრიქონის სხვადასხვა მნიშვნელობისთვის;

ბ) გამოიყენეთ ალგორითმი reverse იმის დასადგენად, წარმოადგენს თუ არა კლავიატურიდან შეტანილი S სტრიქონი პალინდრომს.

დამოუკიდებელი სამუშაო:

1. შექმენით ფუნქცია, რომელიც ითვის 2 სიმბოლოდან უდიდესს. ეს სიმბოლოები ფუნქციის პარამეტრებია. გამოიყენეთ ფუნქცია main-ში კლავიატურიდან შემოტანილი 2 სიმბოლოსთვის და დაბეჭდეთ მათ შორის უდიდესი.

2. დაწერეთ ფუნქცია, რომელიც ითვის 2 ნამდვილი რიცხვიდან უდიდესს. ეს რიცხვები ფუნქციის პარამეტრებია. გამოიყენეთ ფუნქცია main-ში კლავიატურიდან შემოტანილი 2 ნამდვილი რიცხვისთვის და დაბეჭდეთ მათ შორის უდიდესი.

3. დაწერეთ ფუნქცია, რომელიც გამოითვლის ნატურალური N რიცხვის ციფრთა ჯამს. გამოიძახეთ ფუნქცია main-ში კლავიატურიდან შეტანილი N რიცხვის ციფრთა ჯამის დასადგენად. მიღებული შედეგი დაბეჭდეთ.

გაარჩიეთ შესაბამისი პროგრამა:

```
#include <iostream>
using namespace std;
int DigitsSum(int );
int main(){
   int number, sum;
   cin>>number;
   sum = DigitsSum(number);
   cout << "Digits sum of " << number
        << " = " << sum << endl;
   system("pause");
   return 0;
}
int DigitsSum(int N){
   int s =0;
   while(N != 0){
     s += N%10;
     N /= 10;
   }
   return s;
}
```

დავალება:

ა) შეასრულეთ პროგრამა სხვადასხვა ნატურალური რიცხვისთვის.

ბ) ახსენით DigitsSum ფუნქციის ალგორითმი

გ) შეცვალეთ ფუნქციაში while(N != 0) სტრიქონი while(N) -ით, კვლავ შეასრულეთ პროგრამა და ახსენით შედეგი.

4*. დაწერეთ ფუნქცია, რომელიც დაადგენს, წარმოადგენს თუ არა ნატურალური m რიცხვი პალინდრომს. რიცხვი არის პალინდრომი, თუ ის ერთნაირად იკითხება როგორც მარცხნიდან მარჯვნივ ისე მარჯვნიდან მარცხნივ. მაგალითად, 12321 – პალინდრომია.

5.* დაწერეთ ფუნქცია, რომელიც დაადგენს არის თუ არა ნატურალური N რიცხვი ავტომორფული. რიცხვი არის ავტომორფული, თუ მისი კვადრატის ბოლოვდება ამავე რიცხვით. მაგალითად, 5, 6, 25 და ა.შ. – ავტომორფული რიცხვებია.

6*. დაწერეთ ფუნქცია, რომელიც დაადგენს არის თუ არა ნატურალური m რიცხვი სრულყოფილი. რიცხვი არის სრულყოფილი, თუ ის უდრის ყველა თავისი გამყოფის ჯამს, გარდა თავის თავისა. მაგალითად, რიცხვი 6 – სრულყოფილია, ვინაიდან 6=1+2+3.

7*. დაწერეთ ფუნქცია, რომელიც დაადგენს არის თუ არა ნატურალური N რიცხვი მარტივი.

8. ფიბონაჩის რიცხვები განისაზღვრება შემდეგი წესით: $F_0 = F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$, თუ $n \geq 2$. დაწერეთ ფუნქცია, რომელიც ჩაწერს ვექტორში პირველ K ფიბონაჩის რიცხვს და შემდეგ დაბეჭდავს ვექტორს.

გაარჩიეთ შესაბამისი C++– პროგრამა:

```
#include <iostream>
#include <vector>
using namespace std;
void Fibonacci_numbers(int );
int main(){
   int K;
   cout << "ShemoitaneT ricxvebis raodenoba : ";
   cin >> K;
   cout << "\nFibonachis pirveli " << K << " ricxvia\n\n";
   Fibonacci_numbers(K);
   system("pause");
   return 0;
}
void Fibonacci_numbers(int K){
  vector<int> F(K);
  F[0] = F[1] = 1;
  for(int i=2; i<K; i++)
     F[i] = F[i-1] + F[i-2];
  for(int i=0; i<K; i++)
     cout << F[i] << ' ';
  cout << endl;
}
```

დავალება:

ა) გაუშვით პროგრამა შესრულებაზე და შეიტანეთ ფიბონაჩის რიცხვების რაოდენობად 15. დარწმუნდით, რომ პროგრამამ დაბეჭდა:

```
Fibonachis pirveli 15 ricxvia
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
```

ბ) შეასრულეთ პროგრამა K –ს სხვადასხვა მნიშვნელობებისთვის.

9. დაწერეთ ფუნქცია, რომელიც:
   ა) იპოვის ფიბონაჩის მე–20 რიცხვს;
   ბ) იპოვის ფიბონაჩის N –ურ რიცხვს. N ფუნქციის პარამეტრია;
   გ) გამოითვლის ისეთი ფიბონაჩის რიცხვების ჯამს, რომლებიც არ აღემატება M –ს. M ფუნქციის პარამეტრია.

10. შექმენით ფუნქცია `void draw(int M);` რომელიც ერთ სტრიქონად გამოიტანს ეკრანზე M ცალ:
    ა) '*'–ს; ბ) '+'–ს; გ) '-'–ს; დ) სლეშს.

პრაქტიკული მეცადინეობა № 8

პრაქტიკული მეცადინეობის თემები:

- ფუნქციების გადატვირთვა
- ფუნქციები `insert` და `erase`
- ცვლადის მისამართის ,,აღება'', მისამართით ცვლადის მნიშვნელობის ,,აღდგენა''.

მასალა დამოუკიდებლად გასაცნობისათვის:

1. გადატვირთული ფუნქციების ნიმუშები

№	ფუნქცია	ფუნქციის გამოძახება – main()-ის ფრაგმენტი

1	/*ფუნქცია ადგენს არის თუ არა რიცხვი დადებითი */
	bool IsPositive(int K){
	return K > 0;
	}
	bool IsPositive(double K){
	return K > 0;
	}
		int main(){
		int a =10, b =-30;
		double c =-2.5, d =0.0956;
		cout<<boolalpha
		<<a<<" is positive? "<<IsPositive(a)
		<<endl<<b<<" is positive? "<<IsPositive(b)
		<<endl<<c<<" is positive? "<<IsPositive(c)
		<<endl<<d<<" is positive? "<<IsPositive(d)
		<<endl;
		. . .
		}

2	/* ფუნქცია პოულობს რიცხვის მოდულს */
	int my_abs(int x){
	return x < 0 ? -x : x;
	}
	double my_abs(double x){
	return x < 0 ? -x : x;
	}
		int main(){
		int a =-17; double d =20;
		cout<<a<<" -is moduli aris "
		<<my_abs(a)<<endl
		<<d<<" -is moduli aris "
		<<my_abs(d)<<endl;
		. . .
		}

3	/* ფუნქცია პოულობს ორი რიცხვიდან უმცირესს */
	int myMin(int x, int y){
	return x < y ? x : y;
	}
	double myMin(double x,
	double y){
	return x < y ? x : y;
	}
		int main(){
		int a =123, b =39;
		double c =-1.75, d =0.48;
		cout<<a<<" da "<<b
		<<" -s Soris umciresia "
		<<myMin(a, b)<<endl
		<<c<<" da "<<d
		<<" -s Soris umciresia "
		<<myMin(c, d)<<endl;
		. . .
		}

4	/* ფუნქცია ითვლის რიცხვის კვადრატს */
	int square(int a){
	return a * a;
	}
	double square(double a){
	return a * a;
	}
		int main(){
		int x =11;
		double y =-0.7;
		cout<<x<<" -is kvadrati udris "
		<<square(x)<<endl
		<<y<<" -is kvadrati udris "
		<<square(y)<<endl;
		. . .
		}

5	/* ფუნქცია ითვლის ორი რიცხვის საშუალოს */
	double average(int n,int m){
	return (n + m)/2.;
	}
		int main(){
		int a =23, b =12;
		double c =5.5, d =2.25;
		cout<<a<<" da "<<b
		<<" risxvebis saSualo = "
		<<average(a, b)<<endl

```
double average(double n,
               double m){
   return (n + m)/2;
}
```

```
        <<c<<" da "<<d
        <<" risxvebis saSualo = "
        <<average(c, d)<<endl;
  . . .
}
```

2. ზოგიერთი სტანდარტული ფუნქციის გამოყენება ვექტორთან

№	ფუნქციის გამოძახება	ფუნქციის მუშაობის შედეგი

1	`int main(){`
	`  vector<int> v;`
	`  . . .`
	`//ვექტორის პირველ ელემენტად 7 –ის ჩასმა`
	`  v.insert(v.begin(),7);`
	`  . . .`
	`}`
		`//თავდაპირველი ვექტორი`
		`56 -37 302 47 0 300 -1 89`
		`//insert ფუნქციის მუშაობის შემდეგ`
		`7 56 -37 302 47 0 300 -1 89`

2	`int main(){`
	`  vector<int> V;`
	`  . . .`
	`/*ვექტორის მეოთხე (ინდექსით 3) ელემენტად 100–ის ჩასმა */`
	`  V.insert(V.begin()+3,100);`
	`  . . .`
	`}`
		`//თავდაპირველი ვექტორი`
		`56 -37 302 47 0 300 -1 89`
		`//insert ფუნქციის მუშაობის შემდეგ`
		`56 -37 302 100 47 0 300 -1 89`

3	`int main(){`
	`  vector<int> A;`
	`  . . .`
	`//ვექტორის შუა ელემენტად 200–ის ჩასმა`
	`A.insert(A.begin()+A.size()/2,200);`
	`  . . .`
	`}`
		`//თავდაპირველი ვექტორი`
		`56 -37 30 47 0 35 -1 89`
		`//insert ფუნქციის მუშაობის შემდეგ`
		`56 -37 30 47 200 0 35 -1 89`

4	`int main(){`
	`  vector<int> v;`
	`  . . .`
	`//ვექტორის პირველი ელემენტის წაშლა`
	`  v.erase(v.begin());`
	`  . . .`
	`}`
		`//თავდაპირველი ვექტორი`
		`56 -37 30 47 0 35 -1 89`
		`// erase ფუნქციის მუშაობის შემდეგ`
		`-37 30 47 0 35 -1 89`

5	`int main(){`
	`  vector<int> v;`
	`  . . .`
	`/*ვექტორის მეოთხე (ინდექსით 3) ელემენტის წაშლა */`
	`  v.erase(v.begin()+3);`
	`  . . .`
	`}`
		`//თავდაპირველი ვექტორი`
		`56 -37 30 47 0 35 -1 89`
		`// erase ფუნქციის მუშაობის შემდეგ`
		`56 -37 30 0 35 -1 89`

6	`int main(){`
	`  vector<int> v;`
	`  . . .`
	`//ვექტორის შუა ელემენტის წაშლა`
	`v.erase (v.begin()+v.size()/2);`
	`  . . .`
	`}`
		`//თავდაპირველი ვექტორი`
		`7 -14 30 15 -9 35 -1 56 0`
		`// erase ფუნქციის მუშაობის შემდეგ`
		`7 -14 30 15 35 -1 56 0`

საუდიტორიო სამუშაო:

1. გადატვირთეთ ფუნქცია Max, რომელიც დააბრუნებს რამდენიმე ცვლადს შორის უდიდესს. გამოიყენეთ ფუნქცია main-ში 2 მთელი რიცხვისთვის, 3 ნამდვილი რიცხვისთვის, 2 სტრიქონისთვის და 3 სიმბოლოსთვის და დაბეჭდეთ მათ შორის უდიდესი.

```
#include <iostream>
#include <string>
using namespace std;
int Max(int , int );
double Max(double , double , double );
string Max(string, string);
char Max(char , char , char );
int main(){
  int n =45, m =-7;
  double a =3.65, b =75.123, c = -3.143;
  char p ='F', q ='K', u = ',';
  string s1 ="C++ ena", s2 ="C ena";
  cout<<n<<" da "<<m<<" -s Soris udidesia: "
      << Max(n, m) << endl
      << a <<  ", " << b << " da "<< c <<" -s Soris udidesia: "
      << Max(a, b, c) << endl
      << p << ", " << q << " da "<< u << " -s Soris udidesia: "
      <<Max(p, q, u)<<endl
      << s1 << " da "<< s2 << " -s Soris udidesia: "
      << Max(s1, s2)<<endl;
  return 0;
}
int Max(int x, int y){
  return x > y ? x : y;
}
double Max(double x, double y, double z){
  double max = x;
  if(max < y)  max = y;
  if(max < z)  max = z;
  return max;
}
string Max(string x, string y){
  return x > y ? x : y;
}
char Max(char x, char y, char z){
  double max = x;
  if(max < y)  max = y;
  if(max < z)  max = z;
  return max;
}
```

პროგრამა დაბეჭდავს:

```
45 da -7 -s Soris udidesia: 45
3.65, 75.123 da -3.143 -s Soris udidesia:
75.123
F, K da , -s Soris udidesia: K
C++ ena da C ena -s Soris udidesia: C++ ena
Press any key to continue . . .
```

2. data.txt ფაილი შეიცავს მთელ რიცხვებს. ჩაწერეთ ისინი სათანადო სპეციფიკაციის ვექტორში და შემდეგ ჩასვით ვექტორის პირველ ელემენტად რიცხვი 49.

```
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
void printVector(vector<int> );
```

```
int main(){
  vector<int> V;
  int number;
  ifstream fin("data.txt");
  while(fin>>number)
     V.push_back(number);
  printVector(V);
  V.insert(V.begin(),49);
  printVector(V);
  return 0;
}
void printVector(vector<int> x){
   for(int i=0; i<x.size(); i++)
       cout<<x[i]<<' ';
   cout<<endl;
}
```

პროგრამის შესრულების შედეგია:

data.txt ფაილი	გამოსატანი ეკრანი
56 -37 202 0 300 -1 89	56 -37 202 0 300 -1 89 49 56 -37 202 0 300 -1 89 Press any key to continue . . .

3. ქვემოთ მოყვანილი პროგრამები გვიბეჭდავენ სხვადასხვა ტიპის ცვლადების მისამართებს და ამ მისამართებზე განთავსებულ მნიშვნელობებს.

ა)

```
#include <iostream>
using namespace std;
int main(){
  int N =38;
  cout<<"Address of N (in hexodecimal) = "<<&N<<endl
      <<"Address of N (in decimal) = "<<(int)&N<<endl
      <<"Value of N = "<<N<<endl
      <<"Value of N = "<<*(&N)<<endl<<endl;
  int K =123;
  cout<<"Address of K = "<<(int)&K<<endl
      <<"Previous address = "<<(int)(&K - 1)<<endl
      <<"Next address = "<<(int)(&K + 1)<<endl
      <<"Next next address = "<<(int)(&K + 2)<<endl;
  cout<<"int tipi ikavebs "<<sizeof(int)<<" baits\n";
  cout<<"int tipi ikavebs "<<sizeof(N)<<" baits\n";
  system("PAUSE");
  return 0;
}
```

პროგრამა დაბეჭდავს:

```
Address of N (in hexodecimal) = 0012FF60
Address of N (in decimal) = 1245024
Value of N = 38
Value of N = 38

Address of K = 1245012
Previous address = 1245008
Next address = 1245016
Next next address = 1245020
int tipi ikavebs 4 baits
int tipi ikavebs 4 baits
Press any key to continue . . .
```

ბ)

```
#include <iostream>
using namespace std;
int main(){
  double D = 0.00748;
  cout<<"Address of D = "<<(int)&D<<endl
      <<"Value of D = "<<D<<endl
      <<"Value of D = "<<*(&D)<<endl;
  double Q =1.00987;
  cout<<"\nAddress of Q = "<<(int)&Q<<endl
      <<"Previous address = "<<(int)(&Q - 1)<<endl
      <<"Next address = "<<(int)(&Q + 1)<<endl
      <<"Next next address = "<<(int)(&Q + 2)<<endl;
  cout<<"double tipi ikavebs "<<sizeof(double)<<" baits\n";
  return 0;
}
```

პროგრამის შესრულების შედეგია:

```
Address of D = 2293616
Value of D = 0.00748
Value of D = 0.00748

Address of Q = 2293608
Previous address = 2293600
Next address = 2293616
Next next address = 2293624
double tipi ikavebs 8 baits
Press any key to continue . . .
```

გ)

```
#include <iostream>
using namespace std;
int main(){
  char C ='M';
  cout<<"Address of C = "<<(int)&C<<endl
      <<"Value of C = "<<C<<endl
      <<"Value of C = "<<*(&C)<<endl;
  char P ='J';
  cout<<"\nAddress of P = "<<(int)&P<<endl
      <<"Previous address = "<<(int)(&P - 1)<<endl
      <<"Next address = "<<(int)(&P + 1)<<endl
      <<"Next next address = "<<(int)(&P + 2)<<endl;
  cout<<"char tipi ikavebs "<<sizeof(C)<<" baits\n";
  return 0;
}
```

პროგრამის შესრულების შედეგია:

```
Address of C = 2293623
Value of C = M
Value of C = M

Address of P = 2293622
Previous address = 2293621
Next address = 2293623
Next address = 2293624
char tipi ikavebs 1 baits
Press any key to continue . . .
```

დამოუკიდებელი სამუშაო:

1. გადატვირთეთ ფუნქცია MIN, რომელიც დაადგენს 2 ცვლადიდან უმცირესს. გამოიყენეთ ფუნქცია main–ში 2 მთელი რიცხვისთვის, 2 ნამდვილი რიცხვისთვის, 2 სტრიქონისთვის და 2 სიმბოლოსთვის და დაბეჭდეთ შესაბამის ცვლადებს შორის უმცირესი.

2. გადატვირთეთ ფუნქცია min, რომელიც დაადგენს 3 ცვლადიდან უმცირესს. გამოიყენეთ ფუნქცია main–ში 3 მთელი რიცხვისთვის, 3 ნამდვილი რიცხვისთვის, 3 სტრიქონისთვის და 3 სიმბოლოსთვის და დაბეჭდეთ შესაბამის ცვლადებს შორის უმცირესი.

3. გადატვირთეთ ფუნქცია MAX, რომელიც დაადგენს 3 ცვლადიდან უდიდესს. გამოიყენეთ ფუნქცია main–ში 3 მთელი რიცხვისთვის, 3 ნამდვილი რიცხვისთვის, 3 სტრიქონისთვის და 3 სიმბოლოსთვის და დაბეჭდეთ შესაბამის ცვლადებს შორის უდიდესი.

4. გადატვირთეთ ფუნქცია neg, რომელიც დაბრუნებს რიცხვის მოპირდაპირე მნიშვნელობას. გამოიყენეთ ფუნქცია main–ში მთელი რიცხვისთვის, ნამდვილი რიცხვისთვის.

5. ყურადღებით გაარჩიეთ ამოცანა შემდეგი პირობით:
გადატვირთეთ ფუნქცია Mult, ანუ შექმენით მისი 3 ვერსია:
ა) ფუნქცია ბეჭდავს 2 ნამდვილი რიცხვის ნამრავლს;
ბ) ფუნქცია ითვლის 3 მთელი რიცხვის ნამრავლს;
გ) ფუნქცია ითვლის სტრიქონის 2–ჯერ გაზრდილ სიგრძეს.
გამოიყენეთ ფუნქციები ძირითად პროგრამაში სათანადო მონაცემებისთვის.

```
#include <iostream>
#include <string>
using namespace std;
void Mult(double ,double );
int Mult(int ,int ,int );
int Mult(string );
int main(){
   int a, b, c;
   double f, t;
   string s;
   cout<<"Enter string:\n";
   getline(cin, s);
   cout<<"The doubled length of a string = "
       <<Mult(s)<<endl<<endl;
   cout<<"Enter 3 integers:\n";
   cin>>a>>b>>c;
   cout<<"Product of 3 integers = "
       <<Mult(a, b, c)<<endl;
   cout<<"\nEnter 2 reals:\n";
   cin>>f>>t;
   cout<<"Product of 2 reals = ";
   Mult(f, t);
   return 0;
}
void Mult(double a, double b){
   cout<<a*b<<endl;
}
int Mult(int a, int b, int c){
   return a*b*c;
}
int Mult(string k){
   return 2*k.size();
}
```

პროგრამის შესრულების შედეგია:

```
Enter string:
I'm string
The doubled length of a string = 20

Enter 3 integers:
12 5 3
```

```
Product of 3 integers = 180

Enter 2 reals:
1.2 4.0
Product of 2 reals = 4.8

Press any key to continue . . .
```

დავალება: შეასრულეთ პროგრამა კომპიუტერზე კლავიატურიდან შემავალი სხვადასხვა მონაცემებისთვის და დარწმუნდით მის სისწორეში. ***დაიმახსოვრეთ:*** C++ –ში შესაძლებელია ისეთი ფუნქციების გადატვირთვა, რომელთა ალგორითმები განსხვავდებიან.

6. Reals.txt ფაილი შეიცავს ნამდვილ რიცხვებს. ჩაწერეთ ისინი სათანადო სპეციფიკაციის ვექტორში და შემდეგ ჩასვით ვექტორის მეორე (ინდექსით 1) ელემენტად რიცხვი -0.72.

7. ამოცანის პირობაა: numbers.txt ფაილი შეიცავს მთელ რიცხვებს. ჩაწერეთ ისინი სათანადო სპეციფიკაციის ვექტორში და შემდეგ ჩასვით ვექტორის ბოლოსწინა ელემენტად რიცხვი 120.
გაარჩიეთ შესაბამისი C++ –ის პროგრამა:

```
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
void printVector(vector<int> );
int main(){
  vector<int> V;
  int number;
  ifstream fin("data.txt");
  while(fin>>number)
     V.push_back(number);
  printVector(V);
  V.insert(V.end()-1,120);
  printVector(V);
  system("PAUSE");
  return 0;
}
void printVector(vector<int> x){
   for(int i=0; i<x.size(); i++)
       cout<<x[i]<<' ';
   cout<<endl;
}
```

პროგრამის შესრულების შედეგია:

numbers.txt ფაილი	გამოსატანი ეკრანი
7 -14 30 15 -9 35 -1 56 0	7 -14 30 15 -9 35 -1 56 0 7 -14 30 15 -9 35 -1 56 120 0 Press any key to continue . . .

8. ამოცანის პირობაა: sorted.txt ფაილი შეიცავს ზრდადობით დალაგებულ მთელ რიცხვებს. ჩაწერეთ ეს რიცხვები სათანადო სპეციფიკაციის ვექტორში და შემდეგ ჩასვით ვექტორში კლავიატურიდან შემოტანილი მთელი რიცხვი ისე, რომ დალაგება არ დაირღვეს.
გაარჩიეთ შესაბამისი C++ –ის პროგრამა:

```
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
void printVector(vector<int> );
```

```
int main(){
  vector<int> v;
  int num;
  ifstream ifs("sorted.txt");
   while(ifs>>num)
     v.push_back(num);
  printVector(v);
  int ins;
  cout<<"Enter insert number : ";
  cin>>ins;
  if(ins > v[v.size()-1])v.push_back(ins);
  else
  {
    int i =0;
    while(ins > v[i]) i++;
    v.insert( v.begin()+ i, ins );
  }
  printVector(v);
  return 0;
}
void printVector(vector<int> x){
   for(int i=0; i<x.size(); i++)
       cout<<x[i]<<' ';
   cout<<endl;
}
```

პროგრამის შესრულების ერთ-ერთი შედეგია:

sorted.txt ფაილი	გამოსატანი ეკრანი
-7 -5 1 3 5 7 9	-7 -5 1 3 5 7 9 Enter insert number : -10 -10 -7 -5 1 3 5 7 9 Press any key to continue . . .

დავალება:

ა) შეასრულეთ პროგრამა კომპიუტერზე სხვადასხვა ჩასასმელი მთელი რიცხვისთვის და დარწმუნდით პროგრამის სისწორეში;

ბ) ახსენით ელემენტის ჩასმის ალგორითმი.

9. ამოცანის პირობაა: ვექტორში ნამდვილი რიცხვებია. წაშალეთ ვექტორის ბოლო ელემენტი. რიცხვები ვექტორში შეიტანეთ `reals.info` ფაილიდან.

გაარჩიეთ შესაბამისი C++ –ის პროგრამა:

```
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
void printVector(vector<double> );
int main(){
  vector<double> D;
  double number;
  ifstream ifs("reals.info");
   while(ifs>>number)
     D.push_back(number);
  printVector(D);
  D.erase(D.end()-1);
  printVector(D);
  system("PAUSE");
  return 0;
}
```

```
void printVector(vector<double> x){
   for(int i=0; i<x.size(); i++)
       cout<<x[i]<<' ';
   cout<<endl;
}
```

პროგრამის შესრულების შედეგია:

reals.info ფაილი	გამოსატანი ეკრანი
0.05 -3.24 53.75 1 -0.94	0.05 -3.24 53.75 1 -0.94 0.05 -3.24 53.75 1 Press any key to continue . . .

დავალება:

ა) შეასრულეთ პროგრამა კომპიუტერზე და ახსენით შეტყობინება

```
D.erase(D.end()-1);
```

ბ) შეკვალეთ იგი შემდეგით:

```
D.pop_back();
```

და ისევ შეასრულეთ პროგრამა კომპიუტერზე. გააანალიზეთ მიღებული შედეგი.

10. შემოიტანეთ კლავიატურიდან 20 მთელი რიცხვი და ჩაწერეთ სათანადო სპეციფიკაციის ვექტორში. შემდეგ წაშალეთ ვექტორის რიგით მე-15 ელემენტი და დაბეჭდეთ ვექტორი.

11. ამოცანის პირობაა: reals.dat ფაილი შეიცავს ნამდვილ რიცხვებს. ჩაწერეთ ისინი სათანადო სპეციფიკაციის ვექტორში და შემდეგ წაშალეთ კლავიატურიდან შემოტანილი ნამდვილი რიცხვის ტოლი ვექტორის ელემენტი. თუ ასეთი ელემენტი ვექტორში არ არის, დაბეჭდეთ შესაბამისი გზავნილი.

გაარჩიეთ შესაბამისი C++ –ის პროგრამა:

```
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
void printVector(vector<double> );
int main(){
  vector<double> v;
  double num;
  ifstream ifs("reals.dat");
   while(ifs>>num)
     v.push_back(num);
  printVector(v);
  double del;
  cout<<"Enter number to delete: ";
  cin>>del;
  int i;
  for(i=0; i<v.size(); i++)
     if(v[i] == del) break;
  if(i == v.size())
    cout<<"vector not contains this number\n";
  else{
    v.erase( v.begin()+ i );
    printVector(v);
  }
  return 0;
}
void printVector(vector<double> x){
   for(int i=0; i<x.size(); i++)
       cout<<x[i]<<' ';
```

```
    cout<<endl;
}
```

პროგრამის შესრულების ერთ-ერთი შედეგია:

reals.dat ფაილი	გამოსატანი ეკრანი
1.2 -0.03 2.5 0.056 12.45	1.2 -0.03 2.5 0.056 12.45 Enter deleted number : 1.2 -0.03 2.5 0.056 12.45 Press any key to continue . . .

დავალება:

ა) შეასრულეთ პროგრამა კომპიუტერზე სხვადასხვა წასაშლელი ნამდვილი რიცხვისთვის და დარწმუნდით პროგრამის სისწორეში;

ბ) ახსენით ელემენტის წაშლის ალგორითმი.

ლაბორატორიული მეცადინეობა № 8

ლაბორატორიული სამუშაოს ამოწერა:

- ვექტორის გარკვეული თვისების ელემენტებს შორის უდიდესის და უმცირესის დადგენა

ლაბორატორიული სამუშაო:

ამოცანა 1: ვექტორში მთელი რიცხვებია, მათ შორის 3-ის ჯერადებიც. შექმენით ფუნქცია, რომელიც იპოვეთ ვექტორის 3-ის ჯერად ელემენტებს შორის უდიდესის ინდექსს. გამოიყენეთ ფუნქცია main-ში კლავიატურიდან შევსებული ვექტორისთვის და დაბეჭდეთ ნაპოვნი ინდექსი და შესაბამისი რიცხვი.

პროგრამის სახე შემდეგია:

```
//////////////////////////////////////////
// პროგრამა: ვექტორის 3-ის ჯერად ელემენტებს
//           შორის უდიდესის ინდექსის პოვნა
//////////////////////////////////////////
#include <iostream>
#include <vector>
using namespace std;
int maxIndex(vector<int> );
int main(){
  vector<int> A;
  int number;
  while(cin>>number)
     A.push_back(number);
  int index = maxIndex(A);
  cout<<index<<'\t'<<A[index]<<endl;
  return 0;
}
int maxIndex(vector<int> x){
  int i, index;
  for(i=0; i<x.size(); ++i)
    if(x[i]%3 == 0) break;
  index = i;
  for(i++; i<x.size(); ++i)
    if(x[i]%3 == 0 && x[i] > x[index])
       index = i;
  return index;
}
```

დავალება:

ა) შეასრულეთ პროგრამა რამდენჯერმე სხვადასხვა მთელ რიცხვთა მიმდევრობისთვის;

ბ) ახსენით maxIndex ფუნქციის ალგორითმი.

ამოცანა 2: ვექტორში ნამდვილი რიცხვებია, მათ შორის 200-ზე მეტი რიცხვებიც. შექმენით ფუნქცია, რომელიც იპოვეთ ვექტორის 200-ზე მეტ ელემენტებს შორის უმცირესის ინდექსს. გამოიყენეთ ფუნქცია main-ში კლავიატურიდან შევსებული ვექტორისთვის და დაბეჭდეთ ნაპოვნი ინდექსი და შესაბამისი რიცხვი.

პროგრამის სახე შემდეგია:

```
//////////////////////////////////////////
// პროგრამა: ვექტორის 200-ზე მეტ ელემენტებს
//           შორის უმცირესის ინდექსის პოვნა
//////////////////////////////////////////
#include <iostream>
```

```
#include <vector>
using namespace std;
bool property(double );
int minIndex(vector<double> );
int main(){
  vector<double> A;
  double number;
  while(cin>>number)
     A.push_back(number);
  int index = minIndex(A);
  cout<<index<<'\t'<<A[index]<<endl;
  return 0;
}
bool property(double d){
   return d > 200;
}
int minIndex(vector<double> x){
  int i, index;
  for(i=0; i<x.size(); ++i)
    if( property(x[i]) ) break;
  index = i;
  for(i++; i<x.size(); ++i)
    if(x[i] > 200 && x[i] < x[index])
       index = i;
  return index;
}
```

დავალება:

ა) შეასრულეთ პროგრამა რამდენჯერმე სხვადასხვა ნამდვილ რიცხვთა მიმდევრობისთვის;

ბ) ახსენით minIndex ფუნქციის ალგორითმი.

ამოცანა 3: sentence.dat ფაილი შეიცავს სტუდენტების სახელებს. ინფორმაცია ფაილიდან ჩაწერეთ სათანადო სპეციფიკაციის ვექტორში. დაწერეთ ფუნქცია, რომელიც დაადგენს, არის თუ ვექტორში სახელი Anna. გამოიყენეთ ფუნქცია main–ში და დაბეჭდეთ შესაბამისი გზავნილი.

პროგრამის სახე შემდეგია:

```
//////////////////////////////////////////
// პროგრამა: სტრიქონების ვექტორში მოცემული
//                 სიტყვის ძებნა
//////////////////////////////////////////
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
using namespace std;
void printVector(vector<string> );
bool contains(vector<string> );
int main(){
  vector<string> S;
  string word;
  ifstream ifs("sentence.dat");
  while(ifs>>word)
     S.push_back(word);
  printVector(S);
  if ( contains(S) )
    cout<<"Name \"Anna\" is in file\n";
  else
    cout<<"File not contains name \"Anna\"\n";
```

```
  return 0;
}
void printVector(vector<string> x){
   for(int i=0; i<x.size(); i++)
       cout<<x[i]<<' ';
   cout<<endl;
}
bool contains(vector<string> x){
  for(int i=0; i<x.size(); ++i)
     if(x[i] == "Anna") return true;
  return false;
}
```

დავალება:

ა) შექმენით ფაილი `sentence.dat` და ჩაწერეთ მასში სახელები;
ბ) შეასრულეთ პროგრამა, როდესაც ფაილში სახელი `Anna` არის და როდესაც არ არის და დააკვირდით შედეგებს;
გ) ახსენით `contains` ფუნქციის ალგორითმი.

დამოუკიდებელი სამუშაო:

1. ვექტორში მთელი რიცხვებია, მათ შორის ლუწებიც. შექმენით ფუნქცია, რომელიც იპოვეთ ვექტორის ლუწ ელემენტებს შორის უდიდესის ინდექსს. გამოიყენეთ ფუნქცია `main`-ში კლავიატურიდან შეყვანილი ვექტორისთვის და დაბეჭდეთ ნაპოვნი ინდექსი და შესაბამისი რიცხვი.

2. ვექტორში მთელი რიცხვებია, მათ შორის 7-ის ჯერადებიც. შექმენით ფუნქცია, რომელიც იპოვეთ ვექტორის 7-ის ჯერად ელემენტებს შორის უმცირესის ინდექსს. გამოიყენეთ ფუნქცია `main`-ში კლავიატურიდან შეყვანილი ვექტორისთვის და დაბეჭდეთ ნაპოვნი ინდექსი და შესაბამისი რიცხვი.

3. ვექტორში მთელი რიცხვებია, მათ შორის კენტებიც. შექმენით ფუნქცია, რომელიც იპოვეთ ვექტორის კენტ ელემენტებს შორის უმცირესის ინდექსს. რიცხვები ვექტორში ჩაწერეთ `ints.dat` ფაილიდან. გამოიყენეთ ფუნქცია და დაბეჭდეთ ნაპოვნი ინდექსი და შესაბამისი რიცხვი.

4. ვექტორში ნამდვილი რიცხვებია, მათ შორის 25-ზე მეტი რიცხვებიც. შექმენით ფუნქცია, რომელიც იპოვეთ ვექტორის 25-ზე მეტ ელემენტებს შორის უმცირესის ინდექსს. რიცხვები ვექტორში ჩაწერეთ `data.in` ფაილიდან. გამოიყენეთ ფუნქცია და დაბეჭდეთ ნაპოვნი ინდექსი და შესაბამისი რიცხვი.

5. `words.txt` ფაილი შეიცავს სიტყვებს. ინფორმაცია ფაილიდან ჩაწერეთ სათანადო სპეციფიკაციის ვექტორში. დაწერეთ ფუნქცია, რომელიც დაადგენს, არის თუ არა ვექტორში სიტყვა `student`. გამოიყენეთ ფუნქცია `main`-ში და დაბეჭდეთ შესაბამისი გზავნილი.

6. `words.in` ფაილი შეიცავს სიტყვებს. ინფორმაცია ფაილიდან ჩაწერეთ სათანადო სპეციფიკაციის ვექტორში. დაწერეთ ფუნქცია, რომელიც წაშლის ვექტორიდან სიტყვას `war`. თუ ასეთი სიტყვა ვექტორში არ არის, დაბეჭდავს შესაბამის გზავნილს. გამოიყენეთ ფუნქცია `main`-ში და დაბეჭდეთ ვექტორი.

7. `Words.info` ფაილი შეიცავს ზრდადობით დალაგებულ სიტყვებს. ინფორმაცია ფაილიდან ჩაწერეთ სათანადო სპეციფიკაციის ვექტორში. დაწერეთ ფუნქცია, რომელიც ჩასვამს ვექტორში სიტყვას `Peace` ისე, რომ სიტყვების დალაგება არ დაირღვეს. გამოიყენეთ ფუნქცია `main`-ში და დაბეჭდეთ ვექტორი.

8. ვექტორში მთელი რიცხვებია. შექმენით ფუნქცია, რომელიც იპოვეთ ვექტორის 3-ის ჯერად ელემენტებს შორის უდიდესის ინდექსს. თუ 3-ის ჯერადი რიცხვები ვაილში არ არის, ფუნქცია დააბრუნებს -1 -ს. გამოიყენეთ ფუნქცია main-ში კლავიატურიდან შეყვანილი ვექტორისთვის და დაბეჭდეთ ნაპოვნი ინდექსი და შესაბამისი რიცხვი, ან გზავნილი ვაილში 3-ის ჯერადი რიცხვების არარსებობის შესახებ.

გაარჩიეთ შათანადო C++ -პროგრამა:

```
#include <iostream>
#include <vector>
using namespace std;
int specialMax(vector<int> );
int main(){
  vector<int> A;
  int number;
  while(cin>>number)
     A.push_back(number);
  int index = specialMax(A);
  if(index != -1)
     cout<<index<<'\t'<<A[index]<<endl;
  else cout<<"VeqtorSi 3 -is jeradi ricxvebi"
             " ar aris\n";
  system("PAUSE");
  return 0;
}
int specialMax(vector<int> x){
  int i, index;
  for(i=0; i<x.size(); ++i)
    if(x[i]%3 == 0) break;
  if(i == x.size()) return -1;
  index = i;
  for(i++; i<x.size(); ++i)
    if(x[i]%3 == 0 && x[i] > x[index])
       index = i;
  return index;
}
```

9. ვექტორში ნამდვილი რიცხვებია. შექმენით ფუნქცია, რომელიც იპოვეთ ვექტორის 180-ზე ნაკლებ ელემენტებს შორის უდიდესის ინდექსს. თუ 180-ზე ნაკლები რიცხვები ვაილში არ არის, ფუნქცია დააბრუნებს -1 -ს. გამოიყენეთ ფუნქცია main-ში კლავიატურიდან შეყვანილი ვექტორისთვის და დაბეჭდეთ ნაპოვნი ინდექსი და შესაბამისი რიცხვი, ან სათანადო გზავნილი.

პრაქტიკული მეცადინეობა № 9

პრაქტიკული მეცადინეობის თემები:

- პოინტერი. მოქმედებები პოინტერზე
- ფუნქციის პარამეტრი/არგუმენტი – პოინტერი
- ფუნქცია srand

მასალა დამოუკიდებლად გასაცნობისათვის:

1. პოინტერის გამოყენების მარტივი მაგალითები

№	პროგრამის ფრაგმენტი	შედეგი

1	`int n =179;`
	`// განაცხადი მთელი ტიპის p პოინტერზე`
	`int *p;  // იგივეა, რაც int* p;`
	`p = &n;`
	`cout<<*p<<endl;`
		179

2	`double a =12.5407;`
	`double *k;`
	`k = &a;`
	`cout<<*k <<endl;`
		12.5407

3	`char sym = '?';`
	`char* ptr = &sym;`
	`cout<<*ptr<<endl;`
		?

4	`int n = -48;`
	`/* განაცხადი მთელი ტიპის t პოინტერზე და მისი ინიციალიზება n ცვლადის მისამართით */`
	`int* t = &n;`
	`/* სინტაქსურად ვგრეთვე სწორია`
	`int *t = &n; */`
	`cout<<*t<<endl;`
		-48

5	`float y = 4.25;`
	`float* p = &y;`
	`*p = 0.78;`
	`cout<<y<<endl;`
		0.78

6	`char s = 65;`
	`char* ptr = &s;`
	`cout<<*ptr<<' '<<(int)*ptr<<endl;`
		A 65

7	`int c = 100;`
	`int *k;`
	`k = &c;`
	`cout<<(int)k<<' '<<k`
	`<<' '<<*k<<endl;`
		`/* c -ს მისამართი დაიბეჭდა როგორც მთელი რიცხვი და როგორც თექვსმეტობითი რიცხვი, შემდეგ დაიბეჭდა c -ს მნიშვნელობა*/`
		1310560 0013FF60 100

8	`int b = 50;`
	`int* q = &b;`
	`cout<<*q<<' '<<*q+5<<endl;`
		50 55

9	`float t = 35.9;`
	`float *m;`
	`m = &t;`
	`cout<<(int)m<<endl<<*m-1<<endl;`
		2293620
		34.9

10	`char ch = 'E';`
	`char* k = &ch;`
	`cout<<(int)k<<' '<<*k<<endl;`
		2293625 E

11	`double d = 0.75E7;`
	`double* q = &d;`
	`cout<<*q/25<<endl<<(int)q<<endl;`
		300000
		2293616

12	`int k = -37;`
	`int* t = &k;`
	`cout<<*t+1<<' '<<int(t)`
	`<<' '<<int(t+1)<<endl;`
		/* დაიბეჭდა k-ს 1-თით გაზრდილი მნიშვნელობა, k-ს მისამართი და მისი მომდევნო მისამართი */
		-36 2293620 2293624

2. ფუნქცია srand(x) განსაზღვრავს rand()-ის მიერ გენერირებულ პირველ ფსევდოშემთხვევით რიცხვს. x არგუმენტი არის **unsigned int** ტიპის. ხშირად ეს ფუნქცია გამოიყენება იმისთვის, რომ სხვადასხვა ჯერზე გაშვების დროს პროგრამამ შეძლოს ფსევდოშემთხვევითი რიცხვების **სხვადასხვა** მიმდევრობის მიღება. ამისათვის ამ მიმდევრობის გენერაციამდე უნდა გამოვიძახოთ srand(x) ფუნქცია პარამეტრის სხვადასხვა მნიშვნელობით. ხშირად პარამეტრად გამოიყენება სისტემის კალენდარული დროის მიმდინარე მნიშვნელობა time(NULL). srand ფუნქციაზე განაცხადი მოცემულია stdlib.h ფაილში.

№	პროგრამის ფრაგმენტი	შედეგი

1	`srand(0);`
	`cout<<rand()<<endl;`
	`cout<<rand()<<endl;`
	`cout<<rand()<<endl;`
		38
		7719
		21238

2	`srand(1);`
	`cout<<rand()<<endl;`
	`cout<<rand()<<endl;`
	`cout<<rand()<<endl;`
		41
		18467
		6334

3	`srand(157);`
	`cout<<rand()<<endl;`
	`cout<<rand()<<endl;`
	`cout<<rand()<<endl;`
		551
		24051
		7865

4	`srand(time(NULL));`
	`cout<<rand()<<endl;`
	`cout<<rand()<<endl;`
	`cout<<rand()<<endl;`
		30956
		13975
		5745

5	`// იგივე ფრაგმენტი შესრულდა 10 წუთის შემდეგ`
	`srand(time(NULL));`
	`cout<<rand()<<endl;`
	`cout<<rand()<<endl;`
	`cout<<rand()<<endl;`
		31263
		8517
		13809

საუდიტორიო სამუშაო:

1. რა შეცდომას შეიცავს შემდეგი ფრაგმენტი?

```
int a, b =25;
int *p = &a;
*p = &b;
cout<<*p;
```

ფრაგმენტის მესამე სტრიქონი

```
*p = &b;
```

გამოიწვევს კომპილაციის შეცდომას: cannot convert from 'int *' to 'int'
გამოსახულება *p = &b კორექტულია მხოლოდ პოინტერის ინიციალიზების მომენტში, ანუ თუ დავწერთ

```
int *p = &b;
```

სხვა შემთხვევაში *p ადნიშნავს მეხსიერებაში p ცვლადში ჩაწერილ მისამართზე მითავსებულ რიცხვს, ჩვენ შემთხვევაში 25 –ს.
ამიტომ, ინიციალიზების დროს სტილისტურად გამართლებულია ვარსკვლავის მითავსება int –ის გვერდით, ასე

```
int* p = &b;
```

თუმცა, სინტაქსურად ასევე სწორია ჩანაწერი int *p = &b;

2. double *x, y, *z; განაცხადში რა ტიპისაა x, y და z ცვლადები?

x და z წარმოადგენენ double ტიპის პოინტერებს, ხოლო y – double ტიპის რიცხვს. ასეთ შემთხვევებში სტილისტურად გამართლებულია * –ის მიწერა ცვლადის სახელთან, რაც x –ის და z –ის ირიბ მნიშვნელობებზე მიუთითებს, ანუ double ტიპის იმ რიცხვებზე, რომელთა მისამართებიც ჩაიწერება x –ის და z –ის მეხსიერებაში.

სწორია თუ არა შემდეგი მინიჭებები?

y = 0.09;	სწორია, y –ის მეხსიერებაში ჩაიწერება ნამდვილი რიცხვი	
*x = &y;	შეცდომაა, უნდა იყოს	x = &y;
z = y;	შეცდომაა, უნდა იყოს	z = &y;
z = x;	სწორია, პოინტერების მინიჭება ნებადართულია.	

3. დაწერეთ ფუნქცია, რომელიც გაუცვლის მნიშვნელობებს 2 მთელ რიცხვს. გამოიყენეთ ფუნქცია ძირითად პროგრამაში და შედეგი დაბეჭდეთ.

```
#include <iostream>
using namespace std;
void swap (int *, int *);
int main(){
  int a, b;
  cout<<"ShemoitaneT 2 mTeli ricxvi :  ";
  cin>>a>>b;
  cout<<"swap funqciis shesrulebamde :\n";
  cout<<"a = "<<a<<' '<<"b = "<<b<<endl;
  swap(&a, &b);
  cout<<"swap funqciis shesrulebis shemdeg :\n";
  cout<<"a = "<<a<<' '<<"b = "<<b<<endl;
  return 0;
}
void swap (int * x, int * y){
   int dam;
   dam = *x;
   *x = *y;
   *y = dam;
}
```

პროგრამის შესრულების შედეგია:

```
ShemoitaneT 2 mTeli ricxvi :  5 7
swap funqciis shesrulebamde :
a = 5 b = 7
swap funqciis shesrulebis shemdeg :
a = 7 b = 5
Press any key to continue . . .
```

4. ა) დაწერეთ ფუნქცია, რომელიც გაუცვლის მნიშვნელობებს 2 ნამდვილ რიცხვს. გამოიყენეთ ფუნქცია ძირითად პროგრამაში და დაალაგეთ ზრდადობით 3 მოცემული ნამდვილი რიცხვი. შედეგი დაბეჭდეთ.

```
#include <iostream>
```

```
using namespace std;
void swap (double* , double* );
int main(){
    double a, b, c;
    cout<<"ShemoitaneT 3 namdvili ricxvi :\n";
    cin>>a>>b>>c;
    if(a>b) swap(&a, &b);
    if(b>c) swap(&b, &c);
    if(a>b) swap(&a, &b);
    cout<<"ZrdadobiT dalagebuli 3 ricxvi :\n";
    cout<<a<<' '<<b<<' '<<c<<endl;
    return 0;
}
void swap (double* x, double* y){
   double t;
   t = *x; *x = *y; *y = t;
}
```

პროგრამის შესრულების შედეგია:

```
ShemoitaneT 3 namdvili ricxvi :
10.5 -3.7 22.25
ZrdadobiT dalagebuli 3 ricxvi :
-3.7 10.5 22.25
Press any key to continue . . .
```

ბ) მოიყვანეთ ისეთი სამი რიცხვის მაგალითი, რომ პროგრამის მესამე

```
if(a>b) swap(&a, &b);
```

შეტყობინების შესრულება აუცილებელი იყოს რიცხვების დალაგებისათვის.

დამოუკიდებელი სამუშაო:

1. შეასრულეთ და გააანალიზეთ შემდეგი პროგრამები

ა)
```
#include <iostream>
using namespace std;
int main(){
  int x = 15, y = 34;
    int *k, *p;
    k = &y;
    p = k;
    x = *p;
    cout<< x <<endl;
   return 0;
}
```

ბ)
```
#include <iostream>
using namespace std;
int main(){
  double x = 7.86, y = 3.37, *k, *p;
   k = &x;
   p = &y;
   y = *k;
   cout<<*p<<endl;
  return 0;
}
```

2. დაწერეთ ფუნქცია, რომელიც გაუცვლის მნიშვნელობებს 2 სიმბოლოს. გამოიყენეთ ფუნქცია ძირითად პროგრამაში და შედეგი დაბეჭდეთ.

3. დაწერეთ ფუნქცია, რომელიც გაუცვლის მნიშვნელობებს 2 სტრიქონს. გამოიყენეთ ფუნქცია ძირითად პროგრამაში და დაბეჭდეთ. შედეგი დაბეჭდეთ.
4. გადატვირთეთ ფუნქცია swap, რომელიც გაუცვლის მნიშვნელობებს ერთი და იგივე ტიპის 2 ცვლადს.
5. დაწერეთ ფუნქცია, რომელიც:
   ა) გამოითვლის ორი მთელი რიცხვის ჯამსა და ნამრავლს. ჯამი უნდა იყოს ფუნქციის დასაბრუნებელი მნიშვნელობა, ნამრავლი კი მან უნდა დააბრუნოს ცვლადი პარამეტრის საშუალებით. გამოიძახეთ ფუნქცია ძირითად პროგრამაში და დაბეჭდეთ მიღებული შედეგები;

   გაარჩიეთ შესაბამისი C++ -პროგრამა:

```
#include <iostream>
using namespace std;
int calculation(int, int, int* );
int main(){
  int a, b, sum, product;
  cout<<"SemoitaneT 2 mTeli ricxvi : ";
  cin>>a>>b;
  sum = calculation(a, b, &product);
  cout<<"Ricxvebis jami = "<<sum<<endl
      <<"Namravli = "<<product<<endl;
  return 0;
}
int calculation(int x, int y, int* p){
   *p = x * y;
   return x + y;
}
```

   პროგრამის შესრულების შედეგია:

```
SemoitaneT 2 mTeli ricxvi : 12 10
Ricxvebis jami = 22
Namravli = 120
Press any key to continue . . .
```

   ბ) გამოითვლის სამკუთხედის ფართობსა და პერიმეტრს სამი გვერდის მიხედვით (სამკუთხედის გვერდები ფუნქციის პარამეტრებია, პერიმეტრი – ფუნქციის ცვლადი პარამეტრი). გამოიყენეთ ფუნქცია ძირითად პროგრამაში და მოცემული სამკუთხედისათვის დაბეჭდეთ მისი ფართობი და პერიმეტრი;

   გ) გამოითვლის სამკუთხედის ფართობს გვერდისა და მასზე დაშვებული სიმაღლის მიხედვით. გამოიყენეთ ფუნქცია ძირითად პროგრამაში და მოცემული სამკუთხედისათვის დაბეჭდეთ მისი ფართობი.
6. ვექტორი შეიცავს N მთელ შემთხვევით რიცხვს. დაწერეთ პროგრამა, რომელიც იპოვის და დაბეჭდავს ვექტორის:
   ა) უდიდესი ელემენტის მნიშვნელობას და მის ინდექსს;

   გაარჩიეთ შესაბამისი C++ -პროგრამა:

```
#include <iostream>
#include <vector>
#include <ctime>
using namespace std;
void printVector(vector<int> );
int maxIndex(vector<int> );
int main(){
  const int N =15;
  vector<int> X;
```

```
  srand( time(NULL) );
  for(int count =1; count <=N; count++)
     X.push_back( rand() );
  printVector(X);
  int index = maxIndex(X);
  cout<<"Udidesi elementi = "<<X[index]<<endl
      <<"VeqtorSi misi indeqsia "<<index<<endl;
  return 0;
}
void printVector(vector<int> x){
   for(int i=0; i<x.size(); i++)
       cout<<x[i]<<' ';
   cout<<endl;
}
int maxIndex(vector<int> x){
   int ind = 0;
   for(int i=1; i<x.size(); i++)
      if(x[i] > x[ind]) ind = i;
   return ind;
}
```

პროგრამის შესრულების შედეგია:

```
30161  13246  26412  9137  20062  16277  8810  30701
22795 15790 12544 1996 29652 5732 4843
Udidesi elementi = 30701
VeqtorSi misi indeqsia 7
Press any key to continue . . .
```

ბ) ელემენტების ჯამს;

გ) ელემენტების საშუალო არითმეტიკულს.

7. ვექტორი შეიცავს N მთელ რიცხვს. დაწერეთ პროგრამა, რომელიც იპოვის და დაბეჭდავს მოცემული b რიცხვის ტოლ ვექტორის პირველივე ელემენტის ინდექსს, ან, თუ ასეთი ელემენტი არ აღმოჩნდა, შესაბამის გზავნილს. ვექტორი შეავსეთ შემთხვევითი რიცხვებით, b შემოიტანეთ კლავიატურიდან.

გაარჩიეთ შესაბამისი C++ –პროგრამა:

```
#include <iostream>
#include <vector>
#include <ctime>
using namespace std;
void printVector(vector<int> );
int get_index(vector<int> , int);
int main(){
  const int N =10;
  vector<int> A;
  srand( time(NULL) );
  for(int count =1; count <=N; count++)
     A.push_back( rand() );
  printVector(A);
  int b;
  cout<<"b = ";
  cin>>b;
  int index = get_index(A, b);
  if(index == -1)
    cout<<b<<"-s toli ricxvi veqtorSi ar aris\n";
  else
    cout<<b<<"-s toli ricxvis indeqsia "
        <<index<<endl;
```

```
  return 0;
}
void printVector(vector<int> x){
   for(int i=0; i<x.size(); i++)
       cout<<x[i]<<' ';
   cout<<endl;
}
int get_index(vector<int> x, int b){
   int ind = -1;
   for(int i=0; i<x.size(); i++)
      if(x[i] == b) return i;
   return ind;
}
```

დავალება:

შეასრულეთ პროგრამა b –ს სხვადასხვა მნიშვნელობისთვის და დააკვირდით შედეგს.

8. words.in ფაილი შეიცავს ჰარის სიმბოლოთი გამოყოფილ სიტყვებს. დაწერეთ ფუნქცია, რომელიც დაადგენს, არის თუ არა ფაილში სიტყვა exam. თუ ეს სიტყვა ფაილში არის, ფუნქციამ უნდა დააბრუნოს მისი რიგითი ნომერი, თუ კი არა – დააბრუნოს -1. გამოიყენეთ ფუნქცია პროგრამაში და დაბეჭდეთ სათანადო გზავნილი.

ლაბორატორიული მეცადინეობა № 9

ლაბორატორიული სამუშაოს ამოცანა:

- პოინტერის გამოყენება ცვლადებთან
- პოინტერის გამოყენება ფუნქციის პარამეტრად

ლაბორატორიული სამუშაო:

ამოცანა 1: შეასრულეთ და გააანალიზეთ შემდეგი პროგრამა:

```
//////////////////////////////////////////
//  პროგრამა: პოინტერის გამოყენება ცვლადებთან
//////////////////////////////////////////
#include <iostream>
using namespace std;
int main(){
   int a =9, *p =&a;
   cout<<"a = "<<a<<endl;
   cout<<"a-s misamarti = "<< &a <<endl
       <<"p-s mnishvneloba = "<< p <<endl;
   cout<<"p misamartze mnishvneloba = "
       << *p <<endl;
   return 0;
}
```

ამოცანა 2: დაწერეთ ფუნქცია, რომელიც დაადგენს და დააბრუნებს მართკუთხედის ფართობსა და პერიმეტრს. მართკუთხედის გვერდები ნამდვილი რიცხვებია, რომელიც შეგვაქვს კლავიატურიდან. გამოთვლილი ფართობი დააბრუნეთ ფუნქციიდან **return** ოპერატორის, პერიმეტრი კი – პარამეტრის გამოყენებით.

შესაბამის C++ პროგრამას აქვს სახე:

```
//////////////////////////////////////////////////////
//  პროგრამა: პოინტერის გამოყენება ფუნქციის პარამეტრად
//////////////////////////////////////////////////////
#include <iostream>
using namespace std;
double Rectangle(double, double, double* );
int main(){
  double a, b, area, perimeter;
  puts("SemoitaneT marTkuTxedis gverdebi :\n");
  cin>>a>>b;
  area = Rectangle(a, b, &perimeter);
  cout<<"FarTobi = "<<area
      <<"\nPerimetri = "<<perimeter<<endl;
   return 0;
}
double Rectangle(double x, double y, double* per){
    double area;
    area = x * y;
    *per = 2 * (x + y);
    return area;
}
```

დავალება:

ა). გაუშვით პროგრამა შესრულებაზე და შეიტანეთ გვერდების სიდიდეები 2.5 და 3.0. დარწმუნდით, რომ პროგრამამ დაბეჭდა

```
FarTobi = 7.5
Perimetri = 11
```

ბ). შეასრულეთ პროგრამა მაძრთკუთხედის გვერდების სხვა მნიშვნელობებისთვის და დაწრმუნდით პასუხის სისწორეში.

ამოცანა 3: ვექტორი შეიცავს სიტყვებს. დაწერეთ ფუნქცია, რომელიც იპოვის ყველაზე გრძელ სიტყვას და მის სიგრძეს. გამოიყენეთ ფუნქცია `main()`-ში და დაბეჭდეთ შედეგი. ვექტორში სიტყვები ჩაწერეთ `words.txt` ფაილიდან.

შესაბამის C++ -პროგრამას აქვს სახე:

```
//////////////////////////////////////////////////
//  პროგრამა: ფაილიდან სიტყვების წაკითხვა და ყველაზე
//            გრძელი სიტყვის და მისი სიგრძის პოვნა
//////////////////////////////////////////////////
#include <iostream>
#include <vector>
#include <fstream>
#include <string>
using namespace std;
string get_word(vector<string>, int* );
int main(){
  vector<string> v;
  string w;
  ifstream fin("words.txt");
  while(fin>>w)
    v.push_back(w);
  int wordLength;
  string k = get_word(v, &wordLength);
  cout<<"Yvelaze grZeli sityvaa '"
      <<k<<"' sigrZiT "<<wordLength<<endl;
   return 0;
}
string get_word(vector<string> x, int* len){
  string max; int l = 0;
  for(int i=0; i<x.size(); i++)
       if(x[i].length() > l){
          l = x[i].length();
          max = x[i];
       }
  *len = l;
  return max;
}
```

დავალება:

ა). შექმენით `words.txt` ფაილი, ჩაწერეთ მასში ჰარიო გამოყოფილი სიტყვები და დაამატეთ ფაილი პროექტში. შეასრულეთ პროგრამა. მიიღეთ სწორი შედეგი?

ბ). ახსენით `get_word` ფუნქციის ალგორითმი;

გ). ახსენით `*len = l;` შეტყობინების არსი.

დამოუკიდებელი სამუშაო:

1. `input.txt` ფაილში წერია მთელი რიცხვები. შექმენით ფუნქცია, რომელიც შეიტანს ამ რიცხვებს ვექტორში და დააბრუნებს პირველივე დადებითის ინდექსს. თუ დადებითი რიცხვი მათ შორის არ არის, დააბრუნებს -1-ს. გამოიყენეთ ფუნქცია პროგრამაში და დაბეჭდეთ სათანადო გზავნილი.

გაარჩიეთ შესაბამისი C++ -პროგრამა:

```
//////////////////////////////////////////////////
// პროგრამა: ფაილიდან რიცხვების წაკითხვა და პირველივე
//           დადებითი რიცხვის ინდექსის პოვნა
//////////////////////////////////////////////////
```

```
#include <iostream>
#include <vector>
#include <fstream>
using namespace std;
int get_index();
int main(){
   int dad_ind = get_index();
   if(dad_ind == -1)
     cout<<"DadebiTi ricxvebi failSi ar iyo\n";
   else
     cout<<"Pirvelive dadebiTis indeqsia "<<dad_ind<<endl;
   return 0;
}
int get_index(){
  vector<int> v;
  int a, ind = -1;
  ifstream fin("input.txt");
  while(fin>>a)
    v.push_back(a);
  for(int i=0; i<v.size(); i++)
    if(v[i] > 0) return i;
  return ind;
}
```

დავალება:

ა). input.txt ფაილში ჩაწერეთ რამდენიმე მთელი რიცხვი ისე, რომ მათ შორის იყოს ერთი მაინც დადებითი, შეასრულეთ პროგრამა და გააანალიზეთ შედეგი.

ბ). input.txt ფაილში ჩაწერეთ რამდენიმე მთელი რიცხვი ისე, რომ მათ შორის არც ერთი დადებითი არ იყოს, შეასრულეთ პროგრამა და გააანალიზეთ შედეგი.

2. data.in ფაილში ნამდვილი რიცხვებია. დაწერეთ ფუნქცია, რომელიც წაიკითხავს რიცხვებს ფაილიდან და დააბრუნებს პირველივე უარყოფითი რიცხვის რიგით ნომერს. თუ უარყოფითი რიცხვი მათ შორის არ არის, დააბრუნებს -1-ს. გამოიყენეთ ფუნქცია პროგრამაში და დაბეჭდეთ სათანადო გზავნილი.

შესაბამისი C++ -პროგრამის სახე:

```
//////////////////////////////////////////////////////
//  პროგრამა: ფაილიდან რიცხვების წაკითხვა და პირველივე
//            უარყოფითი რიცხვის რიგითი ნომრის პოვნა
//////////////////////////////////////////////////////
#include <iostream>
#include <fstream>
using namespace std;
int get_rn();
int main(){
   int negative_rn = get_rn();
   if(negative_rn == -1)
     cout<<"UaryofiTi ricxvebi failSi ar iyo\n";
   else
     cout<<"Pirvelive uaryofiTis rigiTi nomeria "
         <<negative_rn<<endl;
   return 0;
}
int get_rn(){
   double a;
   int count = 0;
   ifstream ifs("input.txt");
   while ( ifs>>a ){
```

```
      count++;
      if(a < 0)
        return count;
     }
   return -1;
}
```

დავალება: შეასრულეთ და გააანალიზეთ მოცემული პროგრამა. ახსენით `get_rn` ფუნქციის ალგორითმი.

3. მე-9 პრაქტიკული მეცადინეობის ყველა დავალებისთვის დაწერეთ, გამართეთ და შეასრულეთ შესაბამისი პროგრამები.

პრაქტიკული მეცადინეობა № 10

პრაქტიკული მეცადინეობის თემები:

- მითითება (reference). მოქმედებები მითითებაზე
- ფუნქციის პარამეტრი/არგუმენტი – მითითება

მასალა დამოუკიდებლად გასაცნობისათვის:

1. მითითების გამოყენების მარტივი მაგალითები

№	პროგრამის ტრანსგმენტი	შედეგი

1	`int x =-15;`
	`// განსაზღვრი მთელი ტიპის y მითითებაზე`
	`int& y = x;`
	`/* y მთელი რიცხვია, რომელსაც იგივე`
	`მისამართი აქვს, რაც x -ს */`
	`cout<<"x = "<<x<<endl`
	`<<"y = "<<y<<endl;`
		`x = -15`
		`y = -15`

2	`int a = 35;`
	`int& b =a;`
	`cout<<"a-s misamarTia "<<&a<<endl`
	`<<"b-s misamarTia "<<&b<<endl;`
	`cout<<"a = "<<a<<' '<<"b = "<<b<<endl;`
		`a-s misamarTia`
		`0012FF60`
		`b-s misamarTia`
		`0012FF60`
		`a = 35 b = 35`

3	`char ch = 'M';`
	`char& p = ch;`
	`cout<<"p = "<<p<<endl;`
	`++p;`
	`cout<<"p = "<<p<<' '<<"ch = "<<ch<<endl;`
		`p = M`
		`p= N ch= N`

4	`double x =0.067;`
	`double& y =x;`
	`x += 2;`
	`cout<<"x = "<<x<<' '<<"y = "<<y<<endl;`
	`cout<<(int)&x<<' '<<(int)&y<<endl;`
		`x = 2.067 y = 2.067`
		`1245020 1245020`

5	`string s ="TSU";`
	`string& q =s;`
	`cout<<"s : "<<s<<' '<<"q : "<<q<<endl;`
	`cout<<(int)&q<<' '<<(int)&s<<endl;`
		`s : TSU q : TSU`
		`1244980 1244980`

6	`float y = 4.25;`
	`float& p = y;`
	`p = 0.78;`
	`--y;`
	`cout<<"y = "<<y<<' '<<"p = "<<p<<endl;`
		`y = -0.22 p = -0.22`

7	`char s = 97;`
	`char& k = s;`
	`cout<<"k : "<<k<<" code = "<<(int)k<<endl;`
		`k : a code = 97`

8	`string s ="C++";`
	`string& k =s;`
	`// k სტრიქონის შებრუნება`
	`reverse(k.begin(), k.end());`
	`cout<<"s : "<<s<<endl`
	`<<"k : "<<k<<endl;`
		`s : ++C`
		`k : ++C`

9	`string s ="amocana";`
	`string& k =s;`
	`// s სტრიქონის დასაწყისში g ასოს ჩასმა`
	`s.insert(0,"g");`
	`cout<<"s : "<<s<<endl`
	`<<"k : "<<k<<endl;`
		`s : gamocana`
		`k : gamocana`

10	double t =0.067;
	double& u =t;
	u *= 2;
	t++;
	cout<<"t = "<<t<<' '<<"u = "<<u<<endl;
	cout<<(int)&t<<' '<<(int)&u<<endl;
		t = 1.134 u = 1.134
		1245020 1245020

საუდიტორიო სამუშაო:

1. რა შეცდომას შეიცავს შემდეგი ფრაგმენტი?

```
int a = 25;
int& p;
p = &a;
cout<<p;
```

ფრაგმენტის მეორე სტრიქონი

```
int& p;
```

გამოიწვევს კომპილაციის შეცდომას

```
'p' : references must be initialized
```

რაც ნიშნავს, რომ მითითებაზე განაცხადის დროს **აუცილებელია** მითითების ინიციალიზება უკვე არსებული მნიშვნელობით, ჩვენ შემთხვევაში, მთელი a რიცხვის სიდიდით.

ფრაგმენტის მესამე სტრიქონი

```
p = &a;
```

გამოიწვევს კომპილაციის შეცდომას

```
cannot convert from 'int *' to 'int'
```

ეს ნიშნავს, რომ მესამე სტრიქონში ჩვენ ვცდილობთ მთელ p რიცხვს მივანიჭოთ a -ს მისამართი.

სწორი ფრაგმენტი ასე გამოიყურება:

```
int a = 25;
int& p = a;
cout<<p;
```

2. `double x, y; double & z = x ;` განაცხადებში რა ტიპისაა x, y და z ცვლადები?
x, y და z წარმოადგენენ `double` ტიპის რიცხვებს, ამასთან z და x ნამდვილი რიცხვები ერთ მისამართზე არიან განთავსებული. ე.ი. z არის x რიცხვის სხვა სახელი (ფსევდონიმი)

სწორია თუ არა შემდეგი მინიშნებები?

```
y = 0.09;    სწორია, y -ის მეხსიერებაში ჩაიწერება ნამდვილი რიცხვი 0.09
x = y;       სწორია, x -ის მეხსიერებაში ჩაიწერება ნამდვილი რიცხვი 0.09
z = &y;      შეცდომაა, უნდა იყოს  z = y;
z = x + y;   სწორია, x, y და z - ნამდვილი რიცხვებია.
```

3. დაწერეთ ფუნქცია, რომელიც გაუცვლის მნიშვნელობებს 2 სტრიქონს. გამოიყენეთ ფუნქცია ძირითად პროგრამაში კლავიატურიდან შემოტანილი ორი სტრიქონისთვის და შედეგი დაბეჭდეთ.

```
#include <iostream>
#include <string>
using namespace std;
void swap(string & , string &);
```

```
int main(){
  string a, b;
  cout<<"ShemoitaneT 2 striqoni :  ";
  cin>>a>>b;
  cout<<"swap funqciis shesrulebamde :\n";
  cout<<"a = "<<a<<' '<<"b = "<<b<<endl;
  swap(a, b);
  cout<<"swap funqciis shesrulebis shemdeg :\n";
  cout<<"a = "<<a<<' '<<"b = "<<b<<endl;
  return 0;
}
void swap (string& x, string& y){
   string dam = x;
   x = y;
   y = dam;
}
```

პროგრამის შესრულების შედეგია:

```
ShemoitaneT 2 striqoni :  ananasi banani
swap funqciis shesrulebamde :
a = ananasi b = banani
swap funqciis shesrulebis shemdeg :
a = banani b = ananasi
Press any key to continue . . .
```

4. დაწერეთ ფუნქცია, რომელიც მთელ ვექტორში დაითვლის პირველი p ელემენტის ჯამს. ვექტორში რიცხვები ჩაწერეთ ints.txt ფაილიდან. ძირითად პროგრამაში გამოიძახეთ ფუნქცია პირველი:
ა) 10 რიცხვის ჯამის დასადგენად;
ბ) 15 რიცხვის ჯამის დასათვლელად;
გ) კლავიატურიდან შემოტანილი რადენობის რიცხვის ჯამის დასათვლელად.
რიცხვების ვექტორში ჩასაწერად და ვექტორის ელემენტების დასაბეჭდად ასევე შექმნით ფუნქციები.

```
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
void fillVector(vector<int>& );
void printVector(vector<int> );
int Sum(vector<int> , int =10);
int main(){
   vector<int> A;
   fillVector(A);
   cout<<"Vectori :\n";
   printVector(A);
   cout<<"Pirveli 10 ricxvis jami tolia : "
       <<Sum(A)<<endl;
   cout<<"Pirveli 15 ricxvis jami tolia : "
       <<Sum(A, 15)<<endl;
   cout<<"Ramdeni ricxvis jami vipovoT ? ";
   int quantity;
   cin>>quantity;
   cout<<"Pirveli "<<quantity<<" ricxvis jami tolia : "
       <<Sum(A, quantity)<<endl;
   return 0;
}
```

```
void fillVector(vector<int>& a){
   int number;
   ifstream fin("ints.txt");
   while( fin>>number )
      a.push_back(number);
}
void printVector(vector<int> a){
   for(int i=0; i<a.size(); i++)
       cout<<a[i]<<' ';
   cout<<endl;
}
int Sum(vector<int> a, int p){
   int s = 0;
   for(int k=0; k<p; k++)
      s +=a[k];
   return s;
}
```

პროგრამის შესრულების შედეგია:

```
Vectori :
5 -7 100 3 78 123 -124 0 90 -10 -100 120 9 -21 0 4 -5 65 43 20
Pirveli 10 ricxvis jami tolia : 258
Pirveli 15 ricxvis jami tolia : 266
Ramdeni ricxvis jami vipovoT ? 5
Pirveli 5 ricxvis jami tolia : 179
Press any key to continue . . .
```

დამოუკიდებელი სამუშაო:

1. შეასრულეთ და გააანალიზეთ შემდეგი პროგრამები

ა)

```
#include <iostream>
using namespace std;
int main(){
   int j, k = 121;
   int &i = j;
   j = 10;
   cout<<j<<" "<<i<<endl;
   i = k;
   cout<<j<<" "<<i<<endl;
   i++;
   cout<<j<<" "<<i<<endl;
   return 0;
}
```

ბ)

```
#include <iostream>
using namespace std;
int main(){
   double a, b = 10.05;
   double &c = b;
   a = c;
   cout<<c<<" "<<a<<endl;
   c = -3.75;
   cout<<c<<" "<<a<<endl;
   b -= 0.5;
   cout<<c<<" "<<b<<endl;
   return 0;
}
```

2. დაწერეთ ფუნქცია, რომელიც გაუცვლის მნიშვნელობებს 2 სიმბოლოს. სიმბოლოები ფუნქციის ცვლადი პარამეტრები – მითითებებია. გამოიყენეთ ფუნქცია ძირითად პროგრამაში და შედეგი დაბეჭდეთ.

3. დაწერეთ ფუნქცია, რომელიც გაუცვლის მნიშვნელობებს 2 ნამდვილ რიცხვს. გამოიყენეთ ფუნქციის ცვლადი პარამეტრები – მითითებები. გამოიძახეთ ფუნქცია ძირითად პროგრამაში და შედეგი დაბეჭდეთ.

გაარჩიეთ შესაბამისი C++-პროგრამა:

4. დაწერეთ ფუნქცია, რომელიც გაუცვლის მნიშვნელობებს 2 მთელ რიცხვს. გამოიყენეთ ფუნქციის ცვლადი პარამეტრები – მითითებები. გამოიძახეთ ფუნქცია ძირითად პროგრამაში და დაალაგეთ კლებადობით 3 მოცემული მთელი რიცხვი. შედეგი დაბეჭდეთ.

გაარჩიეთ შესაბამისი C++-პროგრამა:

```
#include <iostream>
using namespace std;
void swap (int & , int & );
int main(){
    int a, b, c;
    cout<<"ShemoitaneT 3 mTeli ricxvi :\n";
    cin>>a>>b>>c;
    if(a<b) swap(a, b);
    if(b<c) swap(b, c);
    if(a<b) swap(a, b);
    cout<<"KlebadobiT dalagebuli 3 ricxvi :\n";
    cout<<a<<' '<<b<<' '<<c<<endl;
    return 0;
}
void swap (int& x, int& y){
   int temp;
   temp = x;
   x = y;
   y = temp;
}
```

პროგრამის შესრულების შედეგია:

```
ShemoitaneT 3 mTeli ricxvi :
5 7 9
KlebadobiT dalagebuli 3 ricxvi :
9 7 5
Press any key to continue . . .
```

5. დაწერეთ ფუნქცია, რომელიც გამოთვლის წრის ფართობსა და წრეწირის სიგრძეს. წრის რადიუსი ნამდვილი რიცხვია და ფუნქციის პარამეტრს წარმოადგენს. გამოთვლილი ფართობი დააბრუნეთ ფუნქციიდან **return** ოპერატორის, წრეწირის სიგრძე კი – პარამეტრის გამოყენებით. გამოიძახეთ ფუნქცია ძირითად პროგრამაში კონკრეტული წრისთვის, რომლის რადიუსი შეგვაქვს კლავიატურიდან.

გაარჩიეთ შესაბამისი C++-პროგრამა:

```
#include <iostream>
using namespace std;
const double pi =3.14159;
double Circle(double, double & );
int main(){
   double R;
   double area,length;
   cout<<"Enter radius : ";
   cin>>R;
```

```
    area = Circle(R, length);
    cout<<"Area = "<<area<<endl;
    cout<<"Length = "<<length<<endl;
    return 0;
}
double Circle(double r, double & len){
    len = 2*pi*r;
    return  pi*r*r;
}
```

პროგრამის შესრულების შედეგია:

```
Enter radius : 5
Area = 78.5397
Length = 31.4159
Press any key to continue . . .
```

5. დაწერეთ ფუნქცია, რომელიც:
   ა) გამოითვლის მართკუთხედის ფართობს და პერიმეტრს. პერიმეტრი უნდა იყოს ფუნქციის დასაბრუნებელი მნიშვნელობა, ფართობი კი ფუნქციამ უნდა დააბრუნოს ცვლადი პარამეტრის საშუალებით. გამოიძახეთ ფუნქცია ძირითად პროგრამაში და დაბეჭდეთ მიღებული შედეგები;
   ბ) გამოითვლის მართკუთხა სამკუთხედის ფართობსა და პერიმეტრს კათეტების მიხედვით (სამკუთხედის კათეტები ფუნქციის პარამეტრებია, პერიმეტრი – ფუნქციის ცვლადი პარამეტრი). გამოიყენეთ ფუნქცია ძირითად პროგრამაში და მოკლემული მართკუთხა სამკუთხედისათვის დაბეჭდეთ მისი ფართობი და პერიმეტრი;
6. ვექტორი შეიცავს N მთელ შემთხვევით რიცხვს [-20,80] შუალედიდან. დაწერეთ პროგრამა, რომელიც იპოვის და დაბეჭდავს ვექტორის:
   ა) უმცირესი ელემენტის მნიშვნელობას და მის ინდექსს;
   ბ) ელემენტების ჯამის მეხუთედს;
   გ) ელემენტების ნამრავლს;
   დ) კენტი ელემენტების საშუალო არითმეტიკულს:

   გაარჩიეთ შესაბამისი C++ პროგრამა:

```
#include <iostream>
#include <vector>
#include <ctime>
using namespace std;
const int N =20;
void fillVector(vector<int> & );
void printVector(vector<int> );
bool isOdd(int );
double averageOdds(vector<int> );
int main(){
  vector<int> V;
  fillVector(V);
  printVector(V);
  double average = averageOdds(V);
  cout<<"Kentebis saSualo ariTmetikuli = "
      <<average<<endl;
  return 0;
}
void fillVector(vector<int> & x){
   srand( time(NULL) );
   for(int count =1; count <=N; count++)
      x.push_back( rand()%101 -20 );
}
```

```
void printVector(vector<int> x){
   for(int i=0; i<x.size(); i++)
       cout<<x[i]<<' ';
   cout<<endl;
}
bool isOdd(int k){
   return k%2 != 0;
}
double averageOdds(vector<int> x){
   double sum =0; int count =0;
   for(int i=0; i<x.size(); i++)
      if( isOdd(x[i]) ){
         sum += x[i];
         ++count;
      }
   return sum/count;
}
```

პროგრამის შესრულების შედეგია:

```
49 22 26 3 47 -19 40 22 5 -4 16 -3 79 68 19 78 25 41 44 -16
Kentebis saSualo ariTmetikuli = 24.6
Press any key to continue . . .
```

7. ვექტორში ზრდადობით დალაგებული მთელი რიცხვებია. დაწერეთ ფუნქცია პროტოტიპით `int BinarySearch(vector<int> a, int c);` რომელიც a ვექტორში მოძებნის c –ს ტოლ რიცხვს. ფუნქცია დააბრუნებს მის ინდექსს ან –1 –ს, თუ ასეთი რიცხვი ვექტორში არ არის. გამოიყენეთ ორობითი ძიების ალგორითმი.

პროგრამაში ისარგებლეთ BinarySearch ფუნქციით ვექტორისთვის, რომელიც შეივსებულია sorted_numbers.txt ფაილიდან. საძიებელი რიცხვი შემოიტანეთ კლავიატურიდან.

გაარჩიეთ შესაბამისი C++ –პროგრამა:

```
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
void fillVector(vector<int> & );
void printVector(vector<int> );
int BinarySearch(vector<int> ,int );
int main(){
  vector<int> sorted;
  fillVector(sorted);
  printVector(sorted);
  int check;
  cout<<"\nEnter checked number : ";
  cin>>check;
  int index = BinarySearch(sorted, check);
  if(index == -1)
     cout<<"\nVeqtor does not contain "
         <<check<<endl;
  else cout<<"\nVeqtor contains "<<check
           <<", index is "<<index<<endl;
  return 0;
}
void fillVector(vector<int> & a){
   int number;
   ifstream ifs("sorted_numbers.txt");
   while( ifs>>number )
```

```
        a.push_back(number);
}
int BinarySearch(vector<int> a, int c){
   int l =0, r =a.size()-1, m;
   while( l <= r ){
     m = (l+r)/2;
     if(c == a[m]) return m;
     if(c > a[m]) l =m+1;
     else r =m-1;
   }
   return -1;
}
void printVector(vector<int> x){
   cout<<"Vector : \n\n";
   for(int i=0; i<x.size(); i++)
       cout<<x[i]<<' ';
   cout<<endl;
}
```

პროგრამის შესრულების შედეგია:

sorted_numbers.txt ფაილი	გამოსატანი ეკრანი
1 3 5 7 9 13 24 55 67	Vector : 1 3 5 7 9 13 24 55 67 Enter checked number : 55 Veqtor contains 55, index is 7 Press any key to continue . . .

8. ვექტორში N ნამდვილი განსხვავებული რიცხვია. დაწერეთ ფუნქცია, რომელიც დაადგენს და დააბრუნებს ვექტორის უმცირესი და უდიდესი ელემენტების ინდექსებს. ძირითად პროგრამაში შეაგსეთ ვექტორი კლავიატურიდან, გამოიძახეთ ფუნქცია და დაბეჭდეთ ვექტორის უმცირესი და უდიდესი ელემენტების ჯამი.

გაარჩიეთ შესაბამისი C++ –პროგრამა:

```
#include <iostream>
#include <vector>
using namespace std;
int Indexes(vector<double> , int& );
void fillVector(vector<double>& );
int main(){
  vector<double> D;
  fillVector(D);
  int max_ind, min_ind;
  max_ind = Indexes(D, min_ind);
  double sum = D[max_ind] + D[min_ind];
  cout<<D[max_ind]<<" + "<<D[min_ind]
      <<" = "<<sum<<endl;
  return 0;
}
int Indexes(vector<double> x, int& min_i){
   int max_i;
   max_i = min_i =0;
   for(int i=1; i<x.size(); i++)
      if(x[i] > x[max_i]) max_i = i;
      else
        if( x[i] < x[min_i] ) min_i = i;
   return max_i;
}
```

```
void fillVector(vector<double>& x){
  double real;
  cout<<"SemoitaneT namdvili ricxvebi, "
        "bolos - simbolo\n";
  while( cin>>real )
     x.push_back(real);
}
```

პროგრამა დაბეჭდავს:

```
SemoitaneT namdvili ricxvebi, bolos - simbolo
1.2  0  75.6  -65.2  47.9  -34.75  100.95 *
100.95 + -65.2 = 35.75
Press any key to continue . . .
```

ლაბორატორიული სამუშაოს ამოცანა:

- მითითების გამოყენება ცვლადებთან
- მითითების გამოყენება ფუნქციის პარამეტრად

ლაბორატორიული სამუშაო:

ამოცანა 1: შეასრულეთ და გააანალიზეთ შემდეგი პროგრამა:

```
//////////////////////////////////////////////
//  ავტორი:
//  პროგრამა: მითითების გამოყენება ცვლადებთან
//////////////////////////////////////////////
#include <iostream>
using namespace std;
int main(){
   int a =9;
   int& p =a;
   cout<<"a = "<<a<<endl;
   cout<<"a-s misamarti = "<< &a <<endl
       <<"p-s mnishvneloba = "<< p <<endl
       <<"p-s misamarti = "<< &p <<endl;
   a+=5;
   cout<<"a = "<< a <<endl
       <<"p = "<< p <<endl;
   p-=10;
   cout<<"a = "<< a <<endl
       <<"p = "<< p <<endl;
   return 0;
}
```

ამოცანა 2: დაწერეთ ფუნქცია, რომელიც დაითვლის კვადრატის ფართობსა და დიაგონალის სიგრძეს. კვადრატის გვერდი ნამდვილი რიცხვია, რომელიც შეგვაქვს კლავიატურიდან. კვადრატის ფართობი დააბრუნეთ ფუნქციიდან **return** ოპერატორის, დიაგონალის სიგრძე კი – პარამეტრი-მითითების გამოყენებით.

შესაბამის C++ -პროგრამას აქვს სახე:

```
//////////////////////////////////////////////////////
// პროგრამა: მითითების გამოყენება ფუნქციის პარამეტრად
//////////////////////////////////////////////////////
#include <iostream>
using namespace std;
double Square(double, double & );
int main(){
   double G;
   double area,diagonal;
   cout<<"Enter side : ";
   cin>>G;
   area = Square(G, diagonal);
   cout<<"Area = "<<area<<endl;
   cout<<"Diagonal = "<<diagonal<<endl;
   return 0;
}
double Square (double s, double& d){
   d = s * sqrt(2.0);
   return  s * s;
}
```

დავალება:

ა). გაუშვით პროგრამა შესრულებაზე, შეიტანეთ გვერდის სიდიდე 2.0 და დარწმუნდით, რომ პროგრამა დაბეჭდავს

```
Area = 4
Diagonal = 2.82843
```

ბ). შეასრულეთ პროგრამა კვადრატის გვერდის სხვა მნიშვნელობებისთვის და დარწმუნდით პასუხების სისწორეში.

ამოცანა 3: ჩაწერეთ ვექტორში 150 შემთხვევითი რიცხვი [100, 200] შუალედიდან, შემდეგ დაბეჭდეთ ვექტორის ელემენტები 7 სვეტად. ვექტორის შევსება და ბეჭდვა გაავთორმეთ ფუნქციების სახით.

შესაბამის C++ -პროგრამას აქვს სახე:

```
//////////////////////////////////////////////////////
// ავტორი:
// პროგრამა: ვექტორის ელემენტების ბეჭდვა ცხრილის სახით
//////////////////////////////////////////////////////
#include <iostream>
#include <vector>
#include <iomanip>
#include <ctime>
using namespace std;
const int N =150;
void fillVector(vector<int>& );
void printVector(vector<int> );
int main(){
   vector<int> A;
   fillVector(A);
   printVector(A);
   return 0;
}
void fillVector(vector<int>& x){
   srand( time(NULL) );
   for(int count =1; count <=N; count++)
      x.push_back( rand()%101+100 );
}
void printVector(vector<int> x){
   for(int i=0; i<x.size(); i++){
      cout<<setw(5)<<x[i];
       if( (i+1)%7 == 0)cout<<endl;
   }
   cout<<endl;
}
```

დავალება:

ა). შეასრულეთ პროგრამა რამდენიმეჯერ, ახსენით `setw` მანიპულატორის დანიშნულება;

ბ). ახსენით

```
if( (i+1)%7 == 0)cout<<endl;
```

შეტყობინების მოქმედება;

გ). გადაწერეთ ფუნქცია ისე, რომ ვექტორის ელემენტები დაიბეჭდის: 4 სვეტად; 5 სვეტად შებრუნებული რიგით.

დამოუკიდებელი სამუშაო:

1. ვექტორში ჩაწერილია ზრდადობით დალაგებული სახელები. დაწერეთ ფუნქცია, რომელიც ჩაამატებს ვექტორში კლავიატურიდან შემოტანილ სახელს ისე, რომ სახელების დალაგება არ დაირღვეს. გამოიყენეთ შექმნილი ფუნქცია და დაბეჭდეთ ვექტორი. ვექტორში სახელები ჩაწერეთ `names.txt` ფაილიდან. **შენიშვნა:** ფუნქცია დაწერეთ sort ალგორითმის გამოყენების გარეშე.

გავარჩიეთ შესაბამისი C++ პროგრამა:

```
//////////////////////////////////////////////////////
// პროგრამა: ფაილიდან სტრიქონების ჩაწერა ვექტორში,
//           სტრიქონის ჩასმა  ვექტორში სათანადო ადგილას
//////////////////////////////////////////////////////
#include <iostream>
#include <vector>
#include <string>
#include <fstream>
using namespace std;
void fillVector(vector<string>& );
void printVector(vector<string> );
void insertName(vector<string>& , string );
int main(){
   vector<string> N;
   fillVector(N);
   printVector(N);
   cout<<"Enter name to insert : ";
   string name;
   cin>>name;
   insertName(N, name);
   printVector(N);
   return 0;
}
void printVector(vector<string> x){
   for(int i=0; i<x.size(); i++)
       cout<<x[i]<<' ';
   cout<<endl;
}
void fillVector(vector<string>& x){
   string name;
   ifstream fin("names.txt");
   while( fin>>name )
     x.push_back(name);
}
void insertName(vector<string>& x, string n){
   if( n > x[x.size()-1] ) x.push_back(n);
   else {
     int i =0;
     while( n > x[i] )i++;
     x.insert(x.begin()+i,n);
   }
}
```

დავალება:

ა). შეასრულეთ პროგრამა კლავიატურიდან შემოტანილი სხვადასხვა სახელებისთვის. ახსენით ფუნქციის ალგორითმი.

ბ). შეცვალეთ პროგრამის `insertName` ფუნქცია გამარტივებული ვერსიით: გამოიყენეთ `sort` ალგორითმი. კვლავ შეასრულეთ პროგრამა კლავიატურიდან შემოტანილი სხვადასხვა სახელებისთვის.

2. A და B ვექტორებში კლებადობით დალაგებული მთელი რიცხვებია. დაწერეთ ფუნქცია, რომელიც A და B ვექტორების საფუძველზე შექმნის ასევე კლებადობით დალაგებულ C ვექტორს. გამოიყენეთ ფუნქცია და დაბეჭდეთ C ვექტორი `C.txt` ფაილში.. A და B ვექტორებში რიცხვები შეიტანეთ `A.txt` და `B.txt` ფაილებიდან შესაბამისად. ვექტორის შევსება და ბეჭდვა გაავთორმეთ ფუნქციების სახით. ამოცანა ამოხსენით ორი გზით: 1) `sort` ალგორითმის გამოყენების გარეშე; 2) `sort` ალგორითმის გამოყენებით.

1) გაარჩიეთ შესაბამისი C++ პროგრამა:

```
//////////////////////////////////////////////////
// პროგრამა: დალაგებული მასივების შერწყმა დალაგების
//             შენარჩუნებით
//////////////////////////////////////////////////
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
void fillVector(vector<int>& ,string );
void printVector(vector<int> );
void merge(vector<int> ,vector<int> ,vector<int>& );
int main(){
  vector<int> A, B, C;
  fillVector(A,"A.txt");
  fillVector(B,"B.txt");
  merge(A, B, C);
  printVector(C);
  return 0;
}
void printVector(vector<int> x){
   ofstream fout("C.txt");
   for(int i=0; i<x.size(); i++)
       fout<<x[i]<<' ';
}
void fillVector(vector<int>& x, string fname){
   ifstream fin(fname);
   int n;
   while( fin >> n ){
     x.push_back(n);
   }
}
void merge(vector<int> a,vector<int> b,vector<int>& c){
   int i, j, k;
   i =j =k =0;
   while( i <a.size() && j < b.size() )
      if( a[i] > b[j] ) c.push_back(a[i++]);
      else  c.push_back(b[j++]);
   if(i == a.size())
      for(j; j < b.size(); j++) c.push_back( b[j] );
   else  for(i; i < a.size(); i++) c.push_back( a[i] );
}
```

დავალება:

ა). შექმენით A.txt და B.txt ფაილები და ჩაწერეთ მათში კლებადობით დალაგებული მთელი რიცხვები. შეასრულეთ პროგრამა კომპიუტერზე და დარწმუნდით, რომ მიღებულ ფაილში რიცხვები დალაგებულია კლებადობით.

ბ). ახსენით merge ფუნქციის ალგორითმი.

2) **მითითება:** X ვექტორში მთელი რიცხვების კლებადობით დალაგებას უზრუნველყოფს sort ალგორითმი შემდეგი სახით: `sort(X.rbegin(), X.rend());`

ან `sort(X.begin(), X.end(), greater<int>());`

3. მე-10 პრაქტიკული მეცადინეობის ყველა დავალებისთვის დაწერეთ, გამართეთ და შეასრულეთ შესაბამისი პროგრამები.

პრაქტიკული მეცადინეობა № 11

პრაქტიკული მეცადინეობის თემები:

- ვექტორის ელემენტი – ვექტორი
- ორგანზომილებიანი ვექტორის გადაცემა ფუნქციაში

მასალა დამოუკიდებლად გასაცნობისათვის:

1. ორგანზომილებიან ვექტორზე განაცხადი, მისთვის მეხსიერების გამოყოფა, ვექტორის ინიციალიზება და ელემენტების ბეჭდვა

№	პროგრამის ფრაგმენტი	შედეგი

1	`/* განაცხადი ორგანზომილებიან Matrix ვექტორზე, რომელიც შეიცავს 3 ელემენტს – Row ვექტორს, თითოეულ Row ვექტორში 4 მთელი რიცხვია */`
	`vector<int> Row(4);`
	`vector<vector <int> > Matrix(3, Row);`
		მეხსიერებაში გამოიყოფა უწყვეტი უბანი 12 (3 * 4) მთელი რიცხვისთვის

2	`/* ორგანზომილებიან Matrix ვექტორში რიცხვების ჩაწერა */`
	`vector<int> Row(4);`
	`vector<vector <int> > Matrix(3, Row);`
	`for(int i=0; i<3; i++)`
	`  for (int j=0; j<4; j++)`
	`  Matrix[i][j] = i+j;`
		განაწილებულ მეხსიერებაში ჩაიწერა რიცხვები
		`0 1 2 3 1 2 3 4 2 3 4 5`

3	`/* ორგანზომილებიან Matrix ვექტორში ჩაწერილი რიცხვების ბეჭდვა */`
	`for(int i=0; i<3; i++)`
	`  {`
	`  for (int j=0; j<4; j++)`
	`  cout <<Matrix[i][j]<<'\t';`
	`  cout<<endl;`
	`  }`
		`0 1 2 3`
		`1 2 3 4`
		`2 3 4 5`

4	`/* ორგანზომილებიან ნამდვილ რიცხვთა 2 x 3 ვექტორში კლავიატურიდან შემოტანილი რიცხვების ჩაწერა და შემდეგ ბეჭდვა */`
	`vector<double> row(3);`
	`vector<vector<double>> matrix(2, row);`
	`for(int i=0; i<2; ++i)`
	`  for(int j=0; j<3; ++j)`
	`  cin>>matrix[i][j];`
	`  for(int i=0; i<2; ++i)  {`
	`  for(int j=0; j<3; ++j)`
	`  cout<<setw(9)<<matrix[i][j];`
	`  cout<<endl;`
	`  }`
	`  cout<<endl;`
		`// შეტანა`
		`23.5 -456.75 0.0987`
		`1.065 0.0 -64.25`
		`// გამოტანა`
		`  23.5  -456.75  0.0987`
		`1.065  0  -64.25`

5	`/* ორგანზომილებიან 5 x 9 ვექტორში ჩაწეროთ სიმბოლოები symbols.dat ფაილიდან და შემდეგ დაბეჭდოთ კრებულის სახით table.out ფაილში */`
	`vector<char> row(9);`
		`// symbols.dat ფაილი`
		`asdfashgdfhgfxcbvnxmc`
		`vbiupoiupoiuoiujk;lkj`
		`cxvbvcbvcxbfgshfjayiqywo`

	vector<vector<char>> matrix(5, row);
	ifstream ifs("symbols.dat");
	for(int i=0; i<5; ++i)
	  for(int j=0; j<9; ++j)
	  ifs>>matrix[i][j];
	ofstream ofs("table.out");
	for(int i=0; i<5; ++i) {
	  for(int j=0; j<9; ++j)
	  ofs<<matrix[i][j];
	  ofs<<endl;
	}
		// table.out ფაილი
		asdfashgd
		fhgfxcbvn
		xmcvbiupo
		iupoiuoiu
		jk;lkjcxv

6	/* ორგანზომილებიან მთელ რიცხვთა MxN ვექტორში ჩაწერით შემთხვევითი რიცხვები [-50,100] შუალედიდან, შემდეგ ვიპოვოთ მათი ჯამი და დავბეჭდოთ */
	const int M = 5, N = 4;
	vector<int> row(N);
	vector<vector<int>> matrix(M, row);
	for(int i=0; i<M; ++i)
	  for(int j=0; j<N; ++j)
	  matrix[i][j] = rand()%151 -50;
	int sum = 0;
	for(int i=0; i<M; ++i)
	  for(int j=0; j<N; ++j)
	  sum += matrix[i][j];
	cout<<"sum = "<<sum<<endl;
		// კომპიუტერმა დაგენერირა
		// რიცხვები
		-9 -5 93 25
		93 -30 -48 14
		34 -48 68 9
		-23 16 96 -12
		76 -37 96 -50
		
		// შედეგი
		sum = 358

7	/* ორგანზომილებიან მთელ რიცხვთა MxN ვექტორში შემთხვევითი რიცხვებია. ვიპოვოთ მათ შორის კენტების ჯამი და დავბეჭდოთ */
	const int M = 3, N = 4;
	vector<int> row(N);
	vector<vector<int>> matrix(M, row);
	for(int i=0; i<M; ++i)
	  for(int j=0; j<N; ++j)
	  matrix[i][j] = rand();
	int sum = 0;
	for(int i=0; i<M; ++i)
	  for(int j=0; j<N; ++j)
	  if(matrix[i][j]%2 == 1)
	  sum += matrix[i][j];
	cout<<"sum of odd numbers = "
	  <<sum<<endl;
		// კომპიუტერმა დაგენერირა
		// რიცხვები
		41 18467 6334 26500
		19169 15724 11478 29358
		26962 24464 5705 28145
		
		// შედეგი
		sum of odd numbers = 71527

2. ორგანზომილებიან ვექტორთან მუშაობის ფუნქციების მაგალითები

№	ფუნქციის განსაზღვრა	ფუნქციის გამოძახება

1	/* ორგანზომილებიანი მთელ რიცხვთა 3x4 ვექტორის კლავიატურიდან შემოტანილი რიცხვებით შევსების ფუნქცია */
	void fillMatrix(vector<vector<int>>& A)
	{
	  for(int i=0; i<A.size(); i++)
	  for(int j=0; j< A[i].size(); j++)
	  cin >> A[i][j];
	}
		int main(){
		 vector<int> row(4);
		vector< vector<int> > M(3,row);
		
		 fillMatrix(M);
		  . . .
		}

2	/* იგივე ფუნქცია: ორგანზომილებიანი ვექტორისთვის შემოვიღეთ ტიპი matrix */
	typedef vector<vector <int> > matrix;
	void fillMatrix(matrix & A){
	for(int i=0; i<3; i++)
	for(int j=0; j<4; j++)
	cin >> A[i][j];
	}
		int main(){
		vector<int> row(4);
		matrix M(3, row);
		fillMatrix(M);
		. . .
		}

3	/* ორგანზომილებიანი მთელ რიცხვთა MxN ვექტორის შემთხვევითი რიცხვებით შევსების ფუნქცია: სტრიქონისთვის შემოვიღეთ ტიპი row, ხოლო ორგანზომილებიანი ვექტორისთვის – ტიპი matrix */
	const int M =3, N =4;
	typedef vector<int> Row;
	typedef vector<Row> Matrix;
	void fillMatrix(Matrix & a){
	for(int i=0; i<M; i++)
	for (int j=0; j<N; j++)
	a[i][j] = rand();
	}
		int main(){
		Row R(N);
		Matrix A (M, R);
		fillMatrix(A);
		. . .
		}

	/* ორგანზომილებიანი მთელ რიცხვთა MxN ვექტორის ბეჭდვის ფუნქცია */
	void printMatrix(vector<vector<int>> A){
	for(int i=0; i<M; i++)
	{
	for(int j=0; j<N; j++)
	cout<<A[i][j]<<'\t';
	cout<<endl;
	}
	cout<<endl;
	}
		int main(){
		vector<int> row(N);
		vector< vector<int> > B(M,row);
		fillMatrix(B);
		printMatrix(B);
		. . .
		}

3	/* იგივე ფუნქცია: ორგანზომილებიანი ვექტორისთვის შემოვიღეთ ტიპი matrix */
	typedef vector<vector <int> > matrix;
	void printMatrix(matrix A){
	for(int i=0; i<M; i++)
	{
	for (int j=0; j<N; j++)
	cout<<A[i][j]<<'\t';
	cout<<endl;
	}
	cout<<endl;
	}
		int main(){
		vector<int> Row(N);
		matrix B(M, Row);
		fillMatrix(B);
		printMatrix(B);
		. . .
		}

4	/* ორგანზომილებიანი ნამდვილ რიცხვთა 2x3 ვექტორის კლავიატურიდან შევსების ფუნქცია */
	typedef vector<vector<double>> matrix;
	void fillMatrix(matrix & X){
	for(int i=0; i<2; i++)
	for (int j=0; j<3; j++){
	cin >>X[i][j];
	}
	}
		int main(){
		vector<double> row(3);
		matrix M(2, row);
		fillMatrix(M);
		. . .
		}

5	/* ორგანზომილებიანი ნამდვილ რიცხვთა K x P ვექტორის ბეჭდვის ფუნქცია */
	const int K =2, P =3;
	typedef vector<vector<double>> matrix;
	void printMatrix(matrix X){
	for(int i=0; i<K; i++)
	{
	for (int j=0; j<P; j++)
	cout <<X[i][j]<<'\t';
	cout<<endl;
	}
	cout<<endl;
	}
		int main(){
		vector<double> row(P);
		matrix M(K, row);
		fillMatrix(M);
		printMatrix(M);
		. . .
		}

6	/* ფუნქცია პოულობს და აბრუნებს ნამდვილ რიცხვთა M x N ვექტორის ელემენტების ჯამს */
	typedef vector<double> Row;
	typedef vector< Row > Matrix;
	double Sum(Matrix x){
	double s =0;
	for(int i=0; i<M; i++)
	for(int j=0; j<N; j++)
	s +=x[i][j];
	return s;
	}
		int main(){
		Row r(N);
		Matrix A(M, r);
		// ვექტორის შეყვანა
		. . .
		double s = Sum(A);
		cout<<"sum = "<<s
		<<endl;
		. . .
		}

7	/* ფუნქცია ითვლის მთელ რიცხვთა კვადრატული ვექტორის მთავარი დიაგონალის ელემენტების ჯამს */
	const int M = 3;
	typedef vector<int> Row;
	typedef vector< Row > Matrix;
	int Diagonal(Matrix x){
	int s =0;
	for(int i=0; i<M; ++i)
	s +=x[i][i];
	return s;
	}
		int main(){
		Row r(M);
		Matrix B(M, r);
		// ვექტორის შეყვანა
		. . .
		int s = Diagonal(B);
		cout<<"sum of diagonal"
		" elements = "<<
		s<<endl;
		. . .
		}

საკონტროლო სამუშაო:

1. დაწერეთ პროგრამა, რომელიც შეავსებს 3x4 ორგანზომილებიან ვექტორს შემთხვევითი მთელი რიცხვებით [0,50] შუალედიდან და შემდეგ დაბეჭდავს ვექტორის სტრიქონ-სტრიქონ.

```
#include <iostream>
#include <vector>
using namespace std;
int main(){
   vector<int> Row(4);
   vector<vector<int>> Matrix(3, Row);
   for(int i=0; i<3; i++)
       for(int j=0; j<4; j++){
           Matrix[i][j] = rand()%51;
   }
   for(int i=0; i<3; i++)
   {
       for(int j=0; j<4; j++)
```

```
            cout<<Matrix[i][j]<<'\t';
        cout<<endl;
    }
    cout<<endl;
    return 0;
}
```

პროგრამის შესრულების შედეგია:

```
41      5       10      31
44      16      3       33
34      35      44      44
Press any key to continue . . .
```

2. დაწერეთ პროგრამა, რომელიც შეავსებს $K \times P$ ორგანზომილებიან ვექტორს შემთხვევითი მთელი რიცხვებით [0,50] შუალედიდან და შემდეგ დაბეჭდავს ვექტორს სტრიქონ–სტრიქონ. K და P – შესაბამისად ორგანზომილებიანი ვექტორის სტრიქონებისა და სვეტების რაოდენობა – შემოიღეთ მუდმივების სახით.

```
#include <iostream>
#include <vector>
using namespace std;
const int K = 3, P = 4;
int main(){
    vector<int> Row(P);
    vector<vector<int>> Matrix(K, Row);
    for(int i=0; i<K; i++)
        for(int j=0; j<P; j++){
            Matrix[i][j] = rand()%51;
     }
    for(int i=0; i<K; i++)
     {
        for(int j=0; j<P; j++)
            cout<<Matrix[i][j]<<'\t';
        cout<<endl;
     }
     cout<<endl;
    return 0;
}
```

3. იგივე პირობისთვის ორგანზომილებიანი ვექტორის შემთხვევითი მთელი რიცხვებით შევსება და მისი სტრიქონ-სტრიქონ ბეჭდვა გაავთორმეთ ფუნქციების სახით. გამოიძახეთ ორივე ფუნქცია ძირითად პროგრამაში.

ა).

```
#include <iostream>
#include <vector>
#include <ctime>
using namespace std;
const int K = 3, P = 4;
void fillMatrix(vector< vector<int> > & a);
void printMatrix(vector< vector<int> > a);
int main(){
    vector<int> R(P);
    vector< vector<int> > M(K, R);
    fillMatrix(M);
    printMatrix(M);
    return 0;
}
```

```
void fillMatrix(vector< vector<int> > & a){
   srand(time(NULL));
   for(int i=0; i<K; i++)
      for (int j=0; j<P; j++)
         a[i][j] = rand()%51;
}
void printMatrix(vector< vector<int> > a){
    for(int i=0; i<K; i++)
    {
       for(int j=0; j<P; j++)
          cout <<a[i][j]<<'\t';
       cout<<endl;
    }
    cout<<endl;
}
```

ბ). წინასწარ შემოვიღოთ ორგანზომილებიანი ვექტორის ტიპი matrix და მისი სტრიქონის ტიპი row.

```
#include <iostream>
#include <vector>
#include <ctime>
using namespace std;
typedef vector<int> row;
typedef vector< row > matrix;
void fillMatrix(matrix & a);
void printMatrix(matrix a);
int main(){
   int K, P;
   cout<<"SemoitaneT striqonSi elementebis raodenoba : ";
   cin >> P;
   cout<<"SemoitaneT striqonebis raodenoba : ";
   cin >> K;
    row R(P);
    matrix M(K, R);
    fillMatrix(M);
    printMatrix(M);
    return 0;
}
void fillMatrix(matrix & a){
   srand(time(NULL));
   for(int i=0; i<a.size(); i++)
      for (int j=0; j<a[i].size(); j++)
         a[i][j] = rand()%51;
}
void printMatrix(matrix a){
    for(int i=0; i<a.size(); i++)
    {
       for(int j=0; j<a[i].size(); j++)
          cout <<a[i][j]<<'\t';
       cout<<endl;
    }
    cout<<endl;
}
```

პროგრამის შესრულების ერთ-ერთი შედეგია:

```
29      48      31      17
38      36      42      13
0       32      50      13
```

```
Press any key to continue . . .
```

4. მონაცემების ფაილი წარმოადგენს $M \times N$ მართკუთხა ფორმის ნაკვეთის გეგმას წვიმის შემდეგ, სადაც სიმბოლო '+'-ით აღნიშნულია მშრალი ნაწილები, გუბეები კი – სიმბოლო '-'-ით. დაწერეთ ფუნქცია, რომელიც წაიკითხავს ინფორმაციას ფაილიდან, ჩაწერს მას სიმბოლური ტიპის ორგანზომილებიანი ვექტორის სახით და შემდეგ დაითვლის და დააბრუნებს გუბეების რაოდენობას ნაკვეთზე.

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
using namespace std;
const int M = 4, N = 5;
int Gubeebi();
int main (){
    int k;
    k =Gubeebi();
    cout<<"nakveTze "<<k<<" gubea\n ";
    return 0;
}
int Gubeebi(){
    vector< string > gegma;
    int counter =0;
     string row;
    ifstream fin("nakveTi.txt");
    while(fin>>row)
        gegma.push_back(row);
    for(int i=0; i<gegma.size(); i++)
      for(int j=0; j<gegma[i].size(); ++j)
         if(gegma[i][j] == '-') counter++;
    return counter;
}
```

პროგრამის მუშაობის შედეგია:

nakveTi.txt ფაილი	გამოტანის ეკრანი
-+--+ +-++- ++-++ ---++	nakveTze 9 gubea

დამოუკიდებელი სამუშაო:

1. დაწერეთ ფუნქცია, რომლის პარამეტრიც არის მთელ რიცხვთა ორგანზომილებიანი ვექტორი $M \times N$. ვექტორის განზომილებები M და N წარმოადგენს კონსტანტებს. ფუნქციამ უნდა დაითვალოს ვექტორის 5-ის ჯერადი რიცხვების ჯამი. გამოიყენეთ ფუნქცია პროგრამაში კლავიატურიდან შეყვანილი ორგანზომილებიანი ვექტორისთვის. რიცხვების ჩაწერა ვექტორში ასევე გაავთორმეთ ფუნქციის სახით.

2. დაწერეთ ფუნქცია, რომლის პარამეტრი არის ნამდვილ რიცხვთა ორგანზომილებიანი ვექტორი $M \times N$. ვექტორის განზომილებები M და N წარმოადგენს კონსტანტებს. ფუნქციამ უნდა დაითვალოს და დააბრუნოს ვექტორის:
   ა) არაუარყოფითი ელემენტების რაოდენობა;
   ბ) იმ ელემენტების ჯამი, რომლებიც ეკუთვნიან [-10, 15] შუალედს;
   გ) იმ ელემენტების რაოდენობა, რომლებიც არ ეკუთვნიან [2, 30] შუალედს.

   გამოიყენეთ ფუნქცია პროგრამაში კლავიატურიდან შეყვანილი ორგანზომილებიანი ვექტორისთვის და დაბეჭდეთ შედეგი.

3. დაწერეთ პროგრამა, რომლებიც ნამდვილ რიცხვთა M x N ვექტორის შეავსებს კლავიატურიდან შემოტანილი რიცხვებით, დაითვლის რიგით მეორე სტრიქონის ელემენტების ჯამს, რიგით მესამე სვეტის ელემენტების ჯამს და დაბეჭდავს შედეგს. ვექტორთან ყველა მოქმედებისთვის დაწერეთ და პროგრამაში გამოიყენეთ სათანადო ფუნქციები.

გაარჩიეთ შესაბამისი პროგრამა :

```
#include <iostream>
#include <iomanip>
#include <vector>
using namespace std;
const int M = 3, N = 4;
typedef vector<double> row;
typedef vector< row > matrix;
void fillMatrix(matrix & x);
void printMatrix(matrix x);
int Row2(matrix x);
int Column3(matrix x);
int main(){
   row r(N);
   matrix vec(M, r);
   fillMatrix(vec);
   printMatrix(vec);
   int sum = Row2(vec);
   cout<<"Sum of second row's elements = "<<sum<<endl;
   sum = Column3(vec);
   cout<<"Sum of third column's elements = "<<sum<<endl;
   return 0;
}
void fillMatrix(matrix & x){
    for(int i=0; i<x.size(); i++)
       for(int j=0; j<x[i].size(); j++)
          cin >> x[i][j];
}
void printMatrix(matrix x){
   for(int i=0; i< x.size(); i++){
      for(int j=0; j< x[i].size(); j++)
         cout <<setw(7)<<x[i][j];
      cout<<endl;
   }
   cout<<endl;
}
int Row2(matrix x){
    int s =0;
    for(int k=0; k<x[1].size(); ++k)
        s += x[1][k];
    return s;
}
int Column3(matrix x){
    int s =0;
    for(int k=0; k<x.size(); ++k)
        s += x[k][2];
    return s;
}
```

პროგრამის შესრულების შედეგია:

```
1 4 6 8 0 3 5 12 -7 -21 45 10
      1       4       6       8
```

```
      0       3       5      12
     -7     -21      45      10
Sum of second row's elements = 20
Sum of third column's elements = 56
Press any key to continue . . .
```

4. დაწერეთ ფუნქცია, რომლის პარამეტრი არის მთელ რიცხვთა ორგანზომილებიანი ვექტორი A(M×K). ვექტორის განზომილებები M და K წარმოადგენს კონსტანტებს. ფუნქციამ უნდა დაადგინოს და დაბეჭდოს ვექტორის უდიდესი ელემენტის ინდექსები. გამოიყენეთ ფუნქცია ძირითად პროგრამაში კონკრეტული ორგანზომილებიანი ვექტორისთვის.

5. მონაკვეთის თვალი წარმოადგენს M×N მართკუთხა ფორმის ნაკვეთის გეგმას წვიმის შემდეგ, სადაც სიმბოლო '.'-ით აღნიშნულია მშრალი ნაწილები, გუბეები კი – სიმბოლო '0'-ით. დაწერეთ ფუნქცია, რომელიც წაიკითხავს ინფორმაციას თვალიდან და შემდეგ დაითვლის და დაბრუნებს:
   ა) ნაკვეთის მშრალი უბნების რაოდენობას;
   ბ)* იმ გუბეების რაოდენობას, რომლებიც ნაკვეთის ნაპირებთან მდებარეობს.

   ფუნქცია შექმენით სამი ხერხით: 1). მონაცემების განთავსებით ორგანზომილებიან სიმბოლური ტიპის ვექტორში; 2). მონაცემების განთავსებით სტრიქონების ვექტორში; 3). ვექტორის შემოდების გარეშე.

6. დაწერეთ ფუნქცია, რომელიც დაითვლის მთელ რიცხვთა A(N×N), N=4 ვექტორის რიგით პირველი და მესამე სტრიქონის ელემენტების ჯამს. გამოიყენეთ ფუნქცია ძირითად პროგრამაში კონკრეტული ორგანზომილებიანი ვექტორისთვის.

7. დაწერეთ ფუნქცია, რომელიც დაბეჭდავს ნამდვილ რიცხვთა B(N×M), N=3, M=4 ვექტორის:
   ა) მაქსიმალური ელემენტის სიდიდეს;
   ბ) მინიმალური ელემენტის ინდექსებს;
   გ) უდიდესი ელემენტის სიდიდესა და მის ინდექსებს;
   დ) უდიდესი და უმცირესი ელემენტების ჯამს.

   გამოიყენეთ ფუნქცია ძირითად პროგრამაში ორგანზომილებიანი ვექტორისთვის, რომლის ელემენტებიც წაიკითხეთ `reals.txt` თვალიდან.

8. დაწერეთ ფუნქცია, რომელიც A(N×M) ვექტორში, სადაც N და M მთელი კონსტანტებია
   ა) მოძებნის K რიცხვის ტოლ ელემენტს და დააბრუნებს **true**-ს, თუ ასეთი ელემენტი ვექტორში არის. წინააღმდეგ შემთხვევაში დააბრუნებს **false**-ს. K-ს მნიშვნელობა შემოიტანეთ კლავიატურიდან;
   ბ) მოძებნის [-5, 100] დიაპაზონიდან ერთ-ერთ რიცხვს;
   გ) დაითვლის მინიმალური ელემენტის ტოლი ელემენტების რაოდენობას;
   დ) დაითვლის მაქსიმალური ელემენტისაგან განსხვავებული ელემენტების რაოდენობას.

9. დაწერეთ ფუნქცია, რომელიც დაითვლის მთელ რიცხვთა P(M×N), M=4, N=3 ვექტორის:
   ა) 17-ის ჯერადი ელემენტების ჯამს;
   ბ) დადებითი რიცხვების რაოდენობას;
   გ) კენტი რიცხვების საშუალო არითმეტიკულს;
   დ) კენტინდექსიანი სტრიქონების ელემენტების ჯამს.

10. მთელ რიცხვთა A(NxN)(მაგალითად, N=5) ორგანზომილებიანი ვექტორი შეივსებულია კლავიატურიდან შემოტანილი რიცხვებით. შეამოწმეთ, წარმოადგენს თუ არა იგი

“მაგიურ კვადრატს”. კვადრატულ ვექტორს უწოდებენ “მაგიურ კვადრატს”, თუ მისი ყველა სტრიქონის, ყველა სვეტის და ორივე დიაგონალის ელემენტების ჯამი ტოლია.

11. დაწერეთ ფუნქცია, რომლის პარამეტრებია მთელ რიცხვთა ორგანზომილებიანი ვექტორი და მთელი X რიცხვი და რომელიც გაამრავლებს ვექტორს X–ზე.

12*. დაწერეთ ფუნქცია, რომელიც პარამეტრებად მიიღებს ნამდვილ რიცხვთა სამ ორგანზომილებიან ვექტორს და პირველი ორის ჯამს ჩაწერს მესამეში.

13*. დაწერეთ ფუნქცია, რომელიც პარამეტრებად მიიღებს მთელ რიცხვთა სამ ორგანზომილებიან ვექტორს და პირველი ორის ნამრავლს ჩაწერს მესამეში.

ლაბორატორიული სამუშაოს ამოცანა:

- ორგანზომილებიან ვექტორთან მუშაობა

ლაბორატორიული სამუშაო:

ამოცანა 1: ჩაწერეთ ორგანზომილებიან B(M x N),(M=5; N=3) ვექტორში 15 შემთხვევითი რიცხვი [-10, 30] შუალედიდან, შემდეგ იპოვეთ ვექტორის ლუწი ელემენტების ჯამი და დაბეჭდეთ. ვექტორის შევსება და ლუწების ჯამის გამოთვლა გაანაწილეთ ფუნქციების სახით.

შესაბამის C++ პროგრამას აქვს სახე:

```
//////////////////////////////////////////////////////////
// პროგრამა: ორგანზომილებიანი ვექტორის შემთხვევითი რიცხვებით
//           შევსება და ლუწი ელემენტების ჯამის პოვნა
//////////////////////////////////////////////////////////
#include <iostream>
#include <iomanip>
#include <ctime>
#include <vector>
using namespace std;
const int M = 5, N = 3;
void fillMatrix(vector< vector<int> > & a);
void printMatrix(vector< vector<int> > a);
bool Even(int x);
int sumEven(vector< vector<int> > a);
int main(){
   vector<int> row(N);
   vector< vector<int> > B(M, row);
   fillMatrix(B);
   printMatrix(B);
   int S = sumEven(B);
   cout<<"Sum of even elements = "<<S<<endl;
   return 0;
}
void fillMatrix(vector< vector<int> > & a){
     srand(time(NULL));
     for(int i=0; i<M; i++)
       for(int j=0; j<N; j++)
          a[i][j] = rand()%41 - 10;
}
bool Even(int x){
   return x%2 == 0;
}
int sumEven(vector< vector<int> > a){
   int sum =0;
   for(int i=0; i<M; i++)
      for(int j=0; j<N; j++)
         if( Even(a[i][j]) )
            sum += a[i][j];
   return sum;
}
void printMatrix(vector< vector<int> > a){
   for(int i=0; i<M; i++)
   {
      for(int j=0; j<N; j++)
         cout <<setw(9)<<a[i][j];
```

```
      cout<<endl;
   }
   cout<<endl;
}
```

დავალება:

ა). შეასრულეთ პროგრამა რამდენიმეჯერ, ახსენით `setw` მანიპულატორის დანიშნულება;

ბ). ახსენით `srand(time(NULL));` ფუნქციის მოქმედება.

ამოცანა 2: `V(MxN)` ორგანზომილებიან ვექტორში ნამდვილი რიცხვებია. დაწერეთ ფუნქცია, რომელიც დააბრუნებს ვექტორის უდიდეს ელემენტს. რიცხვები ვექტორში ჩაწერეთ `reals.in` ფაილიდან. ამისათვის ასევე დაწერეთ ფუნქცია. პროგრამაში გამოიყენეთ ორივე ფუნქცია და დაბეჭდეთ უდიდესი ელემენტი.

```
//////////////////////////////////////
//  პროგრამა: ორგანზომილებიან ვექტორში
//            უდიდესი ელემენტის პოვნა
//////////////////////////////////////
#include <iostream>
#include <fstream>
#include <iomanip>
#include <vector>
using namespace std;
const int M = 3, N = 4;
typedef vector<double> row;
typedef vector< row > matrix;
void fillMatrix(matrix & x);
void printMatrix(matrix x);
double maxElem(matrix x);
int main(){
   row R(N);
   matrix V(M, R);
   fillMatrix(V);
   printMatrix(V);
   double Max = maxElem(V);
   cout<<"Udidesia "<<Max<<endl;
   return 0;
}
void fillMatrix(matrix & x){
    ifstream fin("reals.in");
     for(int i=0; i<x.size(); i++)
       for(int j=0; j<x[i].size(); j++)
          fin >> x[i][j];
}
void printMatrix(matrix x){
   for(int i=0; i< x.size(); i++)
   {
      for(int j=0; j< x[i].size(); j++)
         cout <<setw(10)<<x[i][j];
      cout<<endl;
   }
   cout<<endl;
}
double maxElem(matrix x){
   double max = x[0][0];
   for(int i=0; i<x.size(); i++)
      for(int j=0; j<x[i].size(); j++)
         if(x[i][j] > max)
            max = x[i][j];
```

```
    return max;
}
```

დავალება:

ა). შექმენით reals.in ფაილი და ჩაწერეთ მასში 12 ნამდვილი რიცხვი

ბ). შეასრულეთ პროგრამა და შეამოწმეთ მიღებული შედეგი.

ამოცანა 3: A(M x N) ორგანზომილებიან ვექტორში მთელი რიცხვებია. დაწერეთ ფუნქცია, რომელიც გაცვლის ვექტორის უმცირესი ელემენტის მქონე სტრიქონს და ბოლო სტრიქონს. რიცხვები ვექტორში ჩაწერეთ კლავიატურიდან.. პროგრამაში დაბეჭდეთ გარდაქმნილი ვექტორი.

შესაბამის C++ –პროგრამას აქვს სახე:

```
//////////////////////////////////////
//  პროგრამა: ორგანზომილებიან ვექტორში
//           სტრიქონების გაცვლა
//////////////////////////////////////
#include <iostream>
#include <iomanip>
#include <vector>
using namespace std;
const int M = 3, N = 4;
typedef vector<int> row;
typedef vector< row > matrix;
void fillMatrix(matrix & x);
void printMatrix(matrix x);
void Transform(matrix& x);
int main(){
   row R(N);
   matrix A(M, R);
   fillMatrix(A);
   Transform(A);
   printMatrix(A);
   return 0;
}
void Transform(matrix& x){
   int min = x[0][0], rowIndex =0;
   for(int i=0; i<x.size(); i++)
      for(int j=0; j<x[i].size(); j++)
         if(x[i][j] < min){
            min = x[i][j];
            rowIndex = i;
         }
   row temp;
   temp = x[rowIndex]; x[rowIndex] = x[M-1];
   x[M-1] = temp;
}
void fillMatrix(matrix & x){
    for(int i=0; i<x.size(); i++)
       for(int j=0; j<x[i].size(); j++)
          cin >> x[i][j];
}
void printMatrix(matrix x){
   for(int i=0; i<x.size();  i++){
      for(int j=0; j<x[i].size();  j++)
         cout <<setw(9)<<x[i][j];
      cout<<endl;
   }
   cout<<endl;
```

```
}
```

დავალება:

ა). გაუშვით პროგრამა შესრულებაზე, შეიტანეთ 12 მთელი რიცხვი და დააკვირდით მიღებულ შედეგს;

ბ). ახსენით `Transform` ფუნქციის ალგორითმი.

დამოუკიდებელი სამუშაო:

1. `V(MxN)` ორგანზომილებიამ ვექტორში ნამდვილი რიცხვებია. დაწერეთ ფუნქცია, რომელიც დააბრუნებს ვექტორის უმცირესი ელემენტის ინდექსებს. რიცხვები ვექტორში ჩაწერეთ `reals.in` ფაილიდან. ამისათვის ასევე დაწერეთ ფუნქცია. პროგრამაში გამოიყენეთ ორივე ფუნქცია და დაბეჭდეთ უმცირესი ელემენტი და მისი ინდექსები.

გაარჩიეთ შესაბამისი C++ -პროგრამა:

```
//////////////////////////////////////////////
//  პროგრამა: ორგანზომილებიან ვექტორში უმცირესი
//            ელემენტის ინდექსების პოვნა
//////////////////////////////////////////////
#include <iostream>
#include <fstream>
#include <iomanip>
#include <vector>
using namespace std;
const int M = 4, N = 3;
typedef vector<double> row;
typedef vector< row > matrix;
void fillMatrix(matrix & x);
void printMatrix(matrix x);
int Indexes(matrix x, int& rowInd);
int main(){
   row R(N);
   matrix V(M, R);
   fillMatrix(V);
   printMatrix(V);
   int minRInd, minCInd;
   minCInd = Indexes(V, minRInd);
   cout<<"Umsiresia "<<V[minRInd][minCInd]<<endl
        <<minRInd<<" striqonSi, "<<minCInd<<" svetSi\n";
   system("PAUSE");
   return 0;
}
void fillMatrix(matrix & x){
    ifstream fin("reals.in");
     for(int i=0; i<M; i++)
       for(int j=0; j<N; j++)
          fin >> x[i][j];
}
void printMatrix(matrix x){
  for(int i=0; i<M; i++){
     for(int j=0; j<N; j++)
        cout <<setw(11)<<x[i][j];
     cout<<endl;
  }
  cout<<endl;
}
int Indexes(matrix x, int& rowInd){
   int columnInd;
```

```
    rowInd = columnInd =0;
    for(int i=0; i<M; i++)
       for(int j=0; j<N; j++)
           if(x[i][j] < x[rowInd][columnInd]){
                rowInd = i;
                columnInd = j;
           }
     return columnInd;
}
```

დავალება:

ა). შექმენით ფაილი reals.in და ჩაწერეთ მასში M x N ნამდვილი რიცხვი, მაგალითად:

```
0.25 -12.5 6.45 0 10.89 1.125 -5.7 19.65 -31.5 -3.67 9.75 0.05
```

ბ). შეასრულეთ პროგრამა. დარწმუნდით, რომ მიიღეთ შედეგი

```
   0.25   -12.5    6.45
      0   10.89   1.125
   -5.7   19.65   -31.5
  -3.67    9.75    0.05

Umsiresia -31.5
2 striqonSi, 2 svetSi
Press any key to continue . . .
```

2. დაწერეთ პროგრამა, რომელიც განსხვავებულ ნამდვილ რიცხვთა კვადრატული ვექტორისთვის გამოითვლის და დაბეჭდავს უდიდესი ელემენტის მქონე სტრიქონის და უმცირესი ელემენტის მქონე სვეტის სკალარულ ნამრავლს. სტრიქონის და სვეტის სკალარული ნამრავლი არის მათი რიგით შესაბამისი კომპონენტების ნამრავლების ჯამი. ვექტორის განზომილება შემოიღეთ მუდმივის სახით, ელემენტები შემოიტანეთ კლავიატურიდან.

3. მთელ რიცხვთა C(MxN) ორგანზომილებიანი ვექტორი შევსებულია კლავიატურიდან შემოტანილი რიცხვებით. დაბეჭდეთ ვექტორის ყველა "უნაგირა წერტილის" ინდექსები. ვექტორის ელემენტს უწოდებენ "უნაგირა წერტილს", თუ იგი უდიდესია თავის სტრიქონში და ამავე დროს უმცირესია თავის სვეტში.

4. მე-11 პრაქტიკული მეცადინეობის ყველა დავალებისთვის დაწერეთ, გამართეთ და შეასრულეთ შესაბამისი პროგრამები.

პრაქტიკული მეცადინეობა № 12

პრაქტიკული მეცადინეობის თემები:

- ფუნქციის არგუმენტების მნიშვნელობები გულისხმობის პრინციპით
- მომხმარებლის მიერ განსაზღვრული ტიპის შექმნა `class` –ის საშუალებით

მასალა დამოუკიდებლად გასაცნობისათვის:

1. ფუნქციის პარამეტრების ინიციალიზება (არგუმენტების მნიშვნელობები გულისხმობის პრინციპით)

№	ფუნქცია	ფუნქციის გამოძახება – main()–ის ფრაგმენტი

1	/* გამოძახების შესაბამისად ფუნქცია ითვლის ან ორი მთელი რიცხვის, ან სამი მთელი რიცხვის ჯამს */
	int Sum(int a, int b, int c = 0){
	return a + b + c;
	}
		int main(){
		cout<<Sum(-12,49)<<endl;
		cout<<Sum(15,-5,10)
		<<endl;
		. . .
		}

2	// იგივე ფუნქცია
	
	int Sum(int a, int b, int c = 0){
	return a + b + c;
	}
		int main(){
		int x =5, y =9, u =14;
		cout<<x<<" + "<<y<<" = "
		<<Sum(x,y)<<endl;
		cout<<x<<" + "<<y<<" + "
		<<u<<" = "
		<<Sum(x,y,u)<<endl;
		. . .
		}

3	/* გამოძახების შესაბამისად ფუნქცია ითვლის ან ორი ნამდვილი რიცხვის, ან სამი ნამდვილი რიცხვის ნამრავლს */
	double Product(double a, double b,
	double c = 1)
	{
	return a * b * c;
	}
		int main(){
		cout<<Product(2,3)
		<<endl;
		cout<<Product(2,3,5)
		<<endl;
		. . .
		}

4	// იგივე ფუნქცია
	
	double Product(double a, double b,
	double c = 1)
	{
	return a * b * c;
	}
		int main(){
		double x =5, y =7, z =3;
		cout<<x<<" * "<<y<<" = "
		<<Product(x,y)<<endl;
		cout<<x<<" * "<<y<<" * "
		<<z<<" = "
		<<Product(x,y,z)<<endl;
		. . .
		}

5	/* ფუნქცია ითვლის ორი ნამდვილი რიცხვის ჯამს */
	float Add(float a =2, float b=3){
	return a + b;
	}
		int main(){
		cout<<Add()<<endl;
		cout<<Add(3)<<endl;
		cout<<Add(-5,9)<<endl;
		. . .
		}

6	/* ფუნქცია ითვლის სამი მთელი რიცხვის ნამრავლს */
	int Mult(int a =2, int b=3, int c =4){
	return a * b * c;
	}
		int main(){
		cout<<Mult()<<endl;
		cout<<Mult(5)<<endl;
		cout<<Mult(7,4)<<endl;
		cout<<Mult(3,5,6)<<endl;
		. . .

		}

7	/* გამოძახების მიხედვით ფუნქცია ბეჭდავს შესაბამის სახელს */
	void Name(string s = "Alex"){
	cout<<s<<endl;
	}
		int main(){
		Name();
		Name("George");
		Name("Mari");
		. . .
		}

2. ახალი ტიპის შემოღება ღია (public) წევრებიანი class –ის საშუალებით:

№	ტიპის განსაზღვრა	პროგრამის ფრაგმენტი

1	// კლასი ადწერს წრეს:
	class Circle
	{
	public:
	double radius;
	double Area(void)
	{
	return 3.14* radius* radius;
	}
	double Perimeter(void)
	{
	return 2*3.14* radius;
	}
	};
		int main ()
		{
		Circle x;
		x.radius = 2.01;
		cout << "radius= "
		<<x.radius<<endl;
		cout << "Area= "
		<<x.Area()<<endl;
		cout << "perimeter= "
		<<x.Perimeter()<<endl;
		return 0;
		}

2	// მოცემული კლასის ერთი ობიექტისთვის მეორეს მინიჭება
	class Circle
	{
	public:
	double radius;
	double Area(void)
	{
	return 3.14* radius* radius;
	}
	double Perimeter(void)
	{
	return 2*3.14* radius;
	}
	};
		int main () {
		Circle x,y;
		y.radius = 2.01;
		x = y;
		cout << "radius="
		<<x.radius<<endl;
		cout << "Area= "
		<<x.Area()<<endl;
		cout << "perimeter= "
		<<x.Perimeter()<<endl;
		return 0;
		}

3	/* კლასი ადწერს წრეს ცხადი კონსტრუქტორით ახალი ობიექტის კონსტრუირებისთვის */
	class Circle
	{
	public:
	double radius;
	Circle(double value) {
	radius = value;
	}
	double Area() {
	return 3.14* radius* radius;
	}
	double Perimeter() {
	return 2*3.14* radius;
	}
	};
		/* ეს საშუალებას გვაძლევს, რომ კლასის ობიექტები განვსაზღვროთ ისევე ადვილად, როგორც მარტივი ტიპის ცვლადები: */
		int main ()
		{
		Circle x(2.1);
		cout << "radius= "
		<<x.radius<<endl;
		cout << "Area= "
		<<x.Area()<<endl;
		cout << "perimeter= "
		<<x.Perimeter()<<endl;
		return 0;
		}

4	/* იგივე კლასი */
	class Circle
	{
	  public:
	  double radius;
	  Circle(double value) {
	  radius = value;
	  }
	  double Area() {
	  return 3.14* radius* radius;
	  }
	  double Perimeter() {
	  return 2*3.14* radius;
	  }
	};
		/* მაგრამ, ამ შემთხვევაში ვკარგავთ წინა მაგალითში მოყვანილ ობიექტის ინიციალიზაციის გზას: */
		int main ()
		{
		  Circle x; // შეცდომაა !
		  x.radius = 2.1;
		  cout << "radius= "
		  <<x.radius<<endl;
		  cout << "Area= "
		  <<x.Area()<<endl;
		  cout << "perimeter= "
		  <<x.Perimeter()<<endl;
		  return 0;
		}

5	/* გამოსავალს წარმოადგენს წრის კლასზე განაცხადის გაკეთების დროს ორი, არაცხადი და ცხადი კონსტრუქტორის გათვალისწინება ახალი ობიექტის კონსტრუირებისთვის */
	class Circle
	{
	  public:
	  double radius;
	  Circle(double value)
	  {
	  radius = value;
	  }
	  Circle()
	  {
	  radius = 0;
	  }
	  double Area()
	  {
	  return 3.14* radius* radius;
	  }
	  double Perimeter()
	  {
	  return 2*3.14* radius;
	  }
	};
		int main ()
		{
		  Circle x;
		  Circle y(1.1);
		  cout << "x.radius= "
		  << x.radius<<endl;
		  cout << "x.Area= "
		  << x.Area()<<endl;
		  cout << "x.perimeter= "
		  << x.Perimeter()<<endl;
		  cout << "y.radius= "
		  << y.radius<<endl;
		  cout << "y.Area= "
		  << y.Area()<<endl;
		  cout << "y.perimeter= "
		  << y.Perimeter()<<endl;
		  return 0;
		}

საავტორიო სამუშაო:

1. დაწერეთ ფუნქცია Volume, რომელიც საზოგადოდ გამოითვლის მართკუთხა პარალელეპიპედის მოცულობას სამი პარამეტრის – სიმაღლე, ფუძის სიგრძე და სიგანე – მიხედვით. იგულისხმეთ, რომ ძალიან ხშირად გვხვდება ისეთი ფიგურები, რომლისთვისაც სიგრძე არის 3-ის, სიგანე 4-ის, ხოლო სიმაღლე 1-ის ტოლი. გამოიძახეთ ფუნქცია ძირითად პროგრამაში და დაბეჭდეთ: პარალელეპიპედის ფუძის ფართობი, სხვადასხვა პარალელეპიპედის მოცულობა.

```
#include <iostream>
using namespace std;
int Volume(int h =1, int x =3, int y =4);
// h - პარალელეპიპედის სიმაღლე
```

```
// x - პარალელეპიპედის ფუძის სიგრძე,
// y - პარალელეპიპედის ფუძის სიგანე,
int main()
{
   cout<< Volume()<<endl  // ფუძის ფართობი
       << Volume(2)<<endl
       << Volume(3,5)<<endl
       << Volume(4,5,8)<<endl;

   return 0;
}
int Volume(int h, int x, int y)
{
   return x * y * h;
}
```

პროგრამის შესრულების შედეგია:

```
12
24
60
160
Press any key to continue . . .
```

2. class –ის საშუალებით შემოიღეთ ტიპი, რომელიც აღწერს მართკუთხედს. მართკუთხედის კლასის დია (**public**) ველებია: სიგრძე და სიგანე. კლასის ფუნქციებია: ცხადი კონსტრუქტორი, მონაცემების (მისი სიგრძე, სიგანე და ფართობი) ბეჭდვის ფუნქცია, მართკუთხედის ფართობის გამოთვლის ფუნქცია. პროგრამაში გააკეთეთ განაცხადი ობიექტზე – მართკუთხედზე და დაბეჭდეთ მისი მონაცემები. შემდეგ გააორმაგეთ ობიექტის ველების მნიშვნელობები და დაბეჭდეთ მიღებული მართკუთხედის მონაცემები. შემდეგ მიანიჭეთ ველებს განსხვავებული მნიშვნელობები და კიდევ ერთხელ დაბეჭდეთ მიღებული მართკუთხედის მონაცემები.

```
#include <iostream>
using namespace std;
class Rectangle
{
  public:
     double length;
      double width;
      Rectangle(double lengtValue, double widthValue)
      {
          length = lengtValue;
          width = widthValue;
      }
      double Area()
      {
         return length * width;
      }
      void printInfo()
      {
         cout<<"Length is equal to  "<< length <<endl;
         cout<<"width is equal to  "<< width <<endl;
         cout<<"area is equal to  "<< Area() <<endl;
      }
};
int main()
{
   Rectangle rec(3.0,2.1);
   rec.printInfo();
```

```
    rec.length *=2;
    rec.width *=2;
    cout<< endl <<"new data:"<<endl<<endl;
    rec.printInfo();

    rec.length = 5.1;
     rec.width = 4.5;
    cout<< endl <<"new data:"<<endl<<endl;
    rec.printInfo();
    return 0;
}
```

პროგრამის შესრულების ერთ-ერთი შედეგია:

```
Length is equal to  3
width is equal to  2.1
area is equal to  6.3

new data:

Length is equal to  6
width is equal to  4.2
area is equal to  25.2

new data:

Length is equal to  5.1
width is equal to  4.5
area is equal to  22.95
Press any key to continue . . .
```

3. შექმენით კლასი Sea დია (**public**) მონაცემებით: ზღვის დასახელება და სიღრმე. კლასში იქონიეთ პარამეტრებიანი კონსტრუქტორი და ფუნქცია, რომელიც ბეჭდავს კლასის ობიექტის მონაცემებს. Seas.txt ფაილი შეიცავს ინფორმაციას ორი ზღვის შესახებ. პროგრამაში შექმენით კლასის ორ ობიექტი – Seas.txt ფაილის ინფორმაციის საფუძველზე და შემდეგ დაბეჭდეთ უდიდესი სიღრმის მქონე ზღვის მონაცემები.

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
class Sea
{
  public:
    string name;
    int depth;
    Sea(string n, int d) : name(n),depth(d){ }
    void printInfo(){
      cout<<name<<" Sea - "
          <<depth<<endl;
   }

};

int main()
{
   string N; int D;
   ifstream fin("Seas.txt");
   fin>>N>>D;
```

```
    Sea A(N, D);
    fin>>N>>D;
    Sea B(N, D);
    cout<<"Maximum depth have\n";
    if(A.depth > B.depth)
        A.printInfo();
    else B.printInfo();
    return 0;
}
```

პროგრამის შესრულების შედეგია:

Seas.txt ფაილი	გამოტანის ეკრანი
Black 2938 Norwegian 3860	Maximum depth have Norwegian Sea - 3860 Press any key to continue . . .

4. ფაილში "Volcanoes.txt" წერია რამდენიმე ვულკანის მონაცემი – სახელი, სიმაღლე და ადგილმდებარეობა. მაგალითად :

```
etna                 3340     italy
hekla                1491     island
vezuv                1277     italy
kluchevskaia_sopka   4750     kamchatka
fuziama              3776     japan
krakatau             813      azia
kamerun              4070     kamerun
strombol             926      italy
hipokatepetl         5452     mexico
```

შექმენით კლასი ვულკანისთვის, რომლის ველები იქნება სახელი, სიმაღლე და ადგილმდებარეობა, შემდეგ რიგრიგობით წაიკითხეთ ვულკანების მონაცემები ამ ფაილიდან და დაბეჭდეთ ეკრანზე მაღალი ვულკანის მონაცემები.

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
class Volcano
{
  public:
    string name;
    int height;
    string location;
     Volcano(string n, int h, string l) :
                name(n),height(h),location(l){ }
     void printVolcano(){
         cout<< name <<", its height is " << height <<" meters"<<endl
             <<"It is located in "<< location<<endl;
     }
};

int main()
{
    string N, Loc; int D;
    ifstream fin("Volcanoes.txt");
    fin>>N>>D>>Loc;
    Volcano maxVolcano(N, D, Loc);
    while(fin>>N>>D>>Loc)
    {
        Volcano nextxVolcano(N, D, Loc);
```

```
        if(maxVolcano.height < nextxVolcano.height)
            maxVolcano = nextxVolcano;
    }
    cout<<"Maximum height has volcano\n";
    maxVolcano.printVolcano();
    return 0;
}
```

პროგრამა დაბეჭდავს:

```
Maximum height has volcano
hipokatepetl, its height is 5452 meters
It is located in mexico
Press any key to continue . . .
```

დამოუკიდებელი სამუშაო:

1. დაწერეთ ფუნქცია **int** Sum(vector<**int**> a, **int** A =2, **int** B =10); რომელიც მთელ რიცხვთა a ვექტორისთვის შეკრებს მის ელემენტებს, დაწყებული A –ურიდან და დამთავრებული B –ურით (ჩათვლით), შემდეგ დააბრუნებს ამ ჯამს. რიცხვები ვექტორში შეიტანეთ კლავიატურიდან, არანაკლებ 15 მთელი რიცხვისა. ვექტორის შეცსება გაავთორმეთ ფუნქციის სახით. ძირითად პროგრამაში გამოიძახეთ Sum ფუნქცია ვექტორის:
   ა) რიგით მე-5-დან მე-11-მდე რიცხვის ჯამის დასადგენად;
   ბ) რიგით მე-7-დან მე-15-მდე რიცხვის ჯამის დასათვლელად;
   გ) რიგით მე-2-დან მე-10-მდე რიცხვის ჯამის დასათვლელად;
   დ) რიგით მე-3-დან მე-9-მდე რიცხვის ჯამის დასათვლელად;

   გაარჩიეთ გ) და დ) –ს შესაბამისი C++ –პროგრამა:

```
#include <iostream>
#include <vector>
using namespace std;
const int M =15;
void fillVector(vector<int>& a);
int Sum(vector<int> a, int A =2, int B =10);
int main()
{
   vector<int> V, X;
   fillVector(V);
   cout<<"RrigiT me-2-dan me-10-mde (maTi chaTvliT)"
         "\n elementebis jami = "
        <<Sum(V)<<endl;
   fillVector(X);
   cout<<"RrigiT me-3-dan me-9-mde (maTi chaTvliT)"
         "\n elementebis jami = "
        <<Sum(X, 3, 9)<<endl;
   return 0;
}
void fillVector(vector<int>& a){
  int element;
  cout<<"\nEnter "<<M<<" ints\n";
  for(int i=0; i<M; i++)
  {
     cin>>element;
     a.push_back(element);
  }
}
int Sum(vector< int > a, int A, int B){
```

```
    int s =0;
    for(int i=A-1; i<=B-1; i++)
       s += a[i];
    return s;
}
```

პროგრამის შესრულების შედეგია:

```
Enter 15 ints
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
RrigiT me-2-dan me-10-mde (maTi chaTvliT)
 elementebis jami = 54

Enter 15 ints
10 5 7 -5 15 25 33 -10 20 0 2 8 -13 38 1
RrigiT me-3-dan me-9-mde (maTi chaTvliT)
 elementebis jami = 85
Press any key to continue . . .
```

2. დაწერეთ ფუნქცია **double** sum(vector<**double**> x, **int** k =10); რომელიც ნამდვილ რიცხვთა x ვექტორისთვის დაითვლის ბოლო k ელემენტის ჯამს. ვექტორში რიცხვები შეიტანეთ reals.txt ფაილიდან, რომელშიც ჩაწერილია არანაკლებ 20 ნამდვილი რიცხვისა. ძირითად პროგრამაში გამოიძახეთ ფუნქცია ვექტორის ბოლო:

   ა) 10 რიცხვის ჯამის დასადგენად;

   ბ) 15 რიცხვის ჯამის დასათვლელად;

   გ) კლავიატურიდან შემოტანილი N (N<=20) რაოდენობის რიცხვის ჯამის დასათვლელად.

რიცხვების ვექტორში ჩასაწერად ასევე შექმენით ფუნქცია.

გაარჩიეთ გ) –ს შესაბამისი C++ –პროგრამა:

```
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
void fillVector(vector<double>& );
double Sum(vector<double> , int k =10);
int main()
{
   vector<double> V;
   fillVector(V);
   cout<<"Ramdeni ricxvis jami vipovoT ? ";
   int N;
   cin>>N;
   cout<<"Veqtoris bolo "<<N<<" ricxvis jami = "
       <<Sum(V, N)<<endl;
   return 0;
}
void fillVector(vector<double>& a){
   ifstream input("reals.txt");
   double element;
   while( input>>element )
       a.push_back(element);
}
double Sum(vector< double > a, int k){
   double s =0;
   for(int i=a.size()-1; i>=a.size()-k; i--)
      s +=a[i];
   return s;
}
```

პროგრამის შესრულების შედეგია:

reals.txt ფაილი	გამოტანის ეკრანი
0.5 -1.5 2.56 18.35 -22.45 40.0 12.75 -50.55 0.25 -11.25 27.33 112.5 0.45 -0.05 24.75 -100.65 95.0 1.125 -34.75 60.5	Ramdeni ricxvis jami vipovoT ? 7 Veqtoris bolo 7 ricxvis jami = 45.925 Press any key to continue . . .

3. კლასი აღწერს ქვეყანას და მონაცემებით: ქვეყნის დასახელება, მოსახლეობის რაოდენობა (მლნ.), დაკავებული ტერიტორიის ფართობი (კვ. კმ); კლასის ფუნქციებია: კონსტრუქტორი და ინფორმაციის ბეჭდვის ფუნქცია. პროგრამაში გააკეთეთ განაცხადი კლასის ობიექტზე, დაბეჭდეთ ქვეყნის სახელი, მოსახლეობის რაოდენობა და დაკავებული ტერიტორიის ფართობი (გამოიძახეთ ბეჭდვის ფუნქცია).

4. შექმენით კლასი Student, რომელიც ახასიათებს სტუდენტს და მონაცემებით: გვარი, სახელი, ჯგუფის ნომერი და ქულა. კლასში იქონიეთ ფუნქციები – კონსტრუქტორი და ფუნქცია, რომელიც დაბეჭდავს ინფორმაციას სტუდენტის შესახებ. Students.txt ფაილი შეიცავს ინფორმაციას ორი სტუდენტის შესახებ. პროგრამაში გააკეთეთ განაცხადი კლასის ორ ობიექტზე – სტუდენტზე და დაბეჭდეთ იმ სტუდენტის მონაცემები, რომლის ქულაც მეტია.

5. შემოიღეთ კლასი რომელიც შეინახავს კომპიუტერის შესახებ შემდეგ ინფორმაციას: მის სახელს, მეხსიერების მოცულობას და ღირებულებას. კლასში განსაზღვრეთ რამდენიმე საჭირო ფუნქცია.

   ა) ძირითად პროგრამაში გააკეთეთ განაცხადი ამ ტიპის სამ C1, C2 და C3 ობიექტზე და მოიყვანეთ პროგრამული კოდის ფრაგმენტი, რომლის საშუალებითაც ამ ობიექტების ველებში მნიშვნელობების შეტანა მოხდება კლავიატურიდან;

   ბ) მოიყვანეთ პროგრამული კოდის ფრაგმენტი, რომელიც დაადგენს თუ რომელია უფრო ძვირად ღირებული კომპიუტერი და დაბეჭდავს მის მონაცემებს;

   გ) მოიყვანეთ პროგრამული კოდის ფრაგმენტი, რომელიც გაარკვევს თუ ამ სამ კომპიუტერს შორის რომლის სახელია უფრო მოკლე და დაბეჭდავს მისი მეხსიერების მოცულობას.

6. შემოიღეთ კლასი, რომელიც შეინახავს ბიბლიოთეკის წიგნის შესახებ შემდეგ ინფორმაციას: ავტორის გვარს და მკითხველების მიერ ამ წიგნის გატანის რაოდენობას. კლასში განსაზღვრეთ რამდენიმე საჭირო ფუნქცია.

   ა) ძირითად პროგრამაში გააკეთეთ განაცხადი ამ ტიპის სამ B1, B2 და B3 ობიექტზე და მიანიჭეთ თითოეულის ველებს რაიმე მნიშვნელობები;

   ბ) მოიყვანეთ პროგრამული კოდის ფრაგმენტი, რომელიც დაადგენს ნაკლებად პოპულარულ წიგნს და დაბეჭდავს მისი ავტორის გვარს;

   გ) მოიყვანეთ პროგრამული კოდის ფრაგმენტი, რომელიც გაარკვევს თუ ამ სამ წიგნს შორის რომლის დასახელებაა უფრო გრძელი და დაბეჭდავს ინფორმაციას ამ წიგნის შესახებ.

7. შექმენით კლასი bankAccount, რომელიც აღწერს ბანკის ანგარიშს. კლასის ღია მონაცემებია: ბანკის დასახელება, კლიენტის სახელი და გვარი, ანგარიშის ნომერი და ანგარიშზე არსებული თანხა. კლასში გაითვალისწინეთ კონსტრუქტორი და მონაცემების ბეჭდვის ფუნქცია. ფაილში "account.txt" წერია რამდენიმე ანგარიშის შესახებ ინფორმაცია. პროგრამაში რიგრიგობით წაიკითხეთ ანგარიშების მონაცემები ამ ფაილიდან და დაბეჭდეთ ყველა იმ ანგარიშის მონაცემები, სადაც თანხა 500 ლარზე მეტია.

8. შექმენით კლასი, რომელიც შეინახავს მწვერვალის შესახებ შემდეგ ინფორმაციას: მის სახელს და სიმაღლეს (ეს არის კლასის ღია მონაცემები). კლასში იქონიეთ კონსტრუქტორი და მწვერვალის მონაცემების ბეჭდვის ფუნქცია. "info.txt" ფაილში წერია რამდენიმე მწვერვალის შესახებ ინფორმაცია. პროგრამაში რიგრიგობით წაიკითხეთ მწვერვალების მონაცემები ამ ფაილიდან და დაბეჭდეთ ყველაზე მაღალი მწვერვალის მონაცემები.

9. "data.in" ფაილში წერია მდინარეების შესახებ ინფორმაცია. მაგალითად :

```
alazani    407
rioni      333
enguri     206
xrami      220
mtkvari    1515
iori       351
sufsa      117
aragvi     112
```

შემოიღეთ კლასი River, რომელიც აღწერს მდინარეს შემდეგი ღია მონაცემებით: დასახელება და სიგრძე. კლასში იქონიეთ კონსტრუქტორი და მონაცემების ბეჭდვის ფუნქცია. დაწერეთ პროგრამა, რომელიც რიგრიგობით წაიკითხეთ მდინარეების მონაცემებს ფაილიდან და

ა) "data.out" ფაილში დაბეჭდავს ყველა იმ მდინარის მონაცემებს, რომლის სახელი იწყება 'a' –ზე ;

ბ) ეკრანზე გამოიტანს ყველაზე მოკლე მდინარის მონაცემებს.

ლაბორატორიული მეცადინეობა № 12

ლაბორატორიული სამუშაოს ამოწერა:

- ფუნქციის პარამეტრების ინიციალიზება
- მარტივი კლასის მაგალითი

ლაბორატორიული სამუშაო:

ამოცანა 1: დაწერეთ ფუნქცია, პროტოტიპით

```
void fullName(string ="John", string ="Doe");
```

რომელიც ბეჭდავს მის ორ პარამეტრს – სახელსა და გვარს. ძირითად პროგრამაში მოახდინეთ ფუნქციის ტესტირება.

შესაბამის C++ პროგრამას აქვს სახე:

```
//////////////////////////////////////////////
// პროგრამა: ფუნქციის არგუმენტების მნიშვნელობები
//           გულისხმობის პრინციპით
//////////////////////////////////////////////
#include <iostream>
#include <string>
using namespace std;
void fullName(string ="John", string ="Doe");
int main(){
   fullName();
   string first_name;
   cout<<"First Name ? ";
   cin>>first_name;
   fullName(first_name);
   string last_name;
   cout<<"Last Name ? ";
   cin>>last_name;
   fullName(first_name, last_name);
   return 0;
}
void fullName(string first, string last){
    cout<<first<<' '<<last<<endl;
}
```

დავალება:

ა). შეასრულეთ პროგრამა, ახსენით რა რიგით ხდება პარამეტრების ჩანაცვლება არგუმენტებით;

ბ). დაამატეთ ფუნქციას სტრიქონის ტიპის მესამე პარამეტრი სანწყისი მნიშვნელობით `boss`, რომელიც ანიშნავს თანამდებობას. მოახდინეთ ახალი ფუნქციის ტესტირება.

მიაქციეთ ყურადღება ფუნქციის პროტოტიპის სინტაქსს და განსაზღვრის დროს ფუნქციის სათაურის სინტაქსს.

ამოცანა 2: შექმენით კლასი `Point2D`, რომელიც აღწერს წერტილს. წერტილის დიამონაკემებია მისი ორი კოორდინატი. კლასის ფუნქციებია – უპარამეტრო კონსტრუქტორი, კოორდინატების ბეჭდვის ფუნქცია. `main()`–ში გააკეთეთ განაცხადი კლასის ობიექტზე – წერტილზე, დაბეჭდეთ მისი მონაცემები, შემდეგ შეცვალეთ წერტილის კოორდინატები და ისევ დაბეჭდეთ მისი მონაცემები.

შესაბამისი C++ პროგრამა შემდეგი სახისაა:

```
//////////////////////////////////////////////
//  პროგრამა: Point2D კლასის შექმნა და ტესტირება
//////////////////////////////////////////////
```

```
#include <iostream>
using namespace std;
class Point2D
{
  public:
    int x, y;
    Point2D(){
        x = 0, y = 0;
    }
    void showPoint2D(){
       cout<<"( "<<x<<" , "
          <<y<<" )"<<endl;
    }
};

int main()
{
   Point2D P;
   P.showPoint2D();
   P.x = 10;
   P.showPoint2D();
   P.y = 20;
   P.showPoint2D();
   return 0;
}
```

დავალება:

ა). გაუშვით პროგრამა შესრულებაზე და დააკვირდით მიღებულ შედეგს;

ბ). შეცვალეთ წერტილის კოორდინატების მნიშვნელობები და ისევ შეასრულეთ პროგრამა;

ამოცანა 3: შექმენით კლასი Student, რომელიც ახასიათებს სტუდენტს და მონაცემებით გვარი, სახელი და ქულა. კლასში იქონიეთ ფუნქციები – უპარამეტრო კონსტრუქტორი და მონაცემების ბეჭდვის ფუნქცია. პროგრამაში გაიკეთეთ განაცხადი კლასის ობიექტზე – სტუდენტზე და დაბეჭდეთ ობიექტის მონაცემები.

შესაბამის C++ პროგრამას აქვს სახე:

```
////////////////////////////////////
//  პროგრამა: Student კლასის შექმნა
//            და ტესტირება
////////////////////////////////////
#include <iostream>
#include <string>
using namespace std;
class Student
{
  public:
      string firstName;
      string lastName;
      int qula;

      Student(){
         firstName = "Maia";
         lastName = "LomiZe";
         qula = 98;
      }
      void printInfo(){
          cout<<firstName<<' '
              <<lastName<<endl
```

```
            <<qula<<endl;
    }
};

int main()
{
   Student X;
   X.printInfo();
   return 0;
}
```

დავალება:

ა). რა ინფორმაციას დაბეჭდავს `printInfo` ფუნქცია, თუ პროგრამაში აღვწერთ კიდევ ერთ `Student` კლასის ობიექტს, და რატომ?

ბ). გადატვირთეთ კონსტრუქტორი: დაამატეთ კლასში პარამეტრებიანი კონსტრუქტორი

```
Student(string first, string last, int q = 100)
{
    firstName = first;
    lastName = last;
    qula = q;
}
```

`main` –ში გააკეთეთ განაკხადი კლასის ობიექტზე და მოახდინეთ მისი ინიციალიზება, მაგალითად ასე:

```
Student Y("Irakli","Beridze");
```

შემდეგ დაბეჭდეთ Y ობიექტის ყველა მონაცემი.

გ). დაამატეთ `main` –ში კოდის ფრაგმენტი, რომელიც:
შექმნის კლასის ახალ ობიექტს და მასზე განაკხადისთანავე მოახდინეთ მის ინიციალიზებას კლავიატურიდან შემოტანილი სტუდენტის გვარით, სახელით და ქულით, ხოლო შემდეგ დაბეჭდავს ამ სტუდენტის შესახებ ინფორმაციას.

დამოუკიდებელი სამუშაო:

1. შექმენით კლასი `Box` შემდეგი მონაცემებით: სიგრძე, სიგანე და სიმაღლე. კლასში იქონიეთ ფუნქციები – კონსტრუქტორი, მონაცემების ბეჭდვის ფუნქცია და ყუთის მოცულობის გამოთვლის ფუნქცია. `main` –ში გააკეთეთ განაკხადი `Box` კლასის ობიექტზე, დაბეჭდეთ მისი მონაცემები და მოცულობა.
2. მე -12 პრაქტიკული მეცადინეობის ყველა დავალებისთვის დაწერეთ, გამართეთ და შეასრულეთ შესაბამისი პროგრამები.

პრაქტიკული მეცადინეობა № 13

პრაქტიკული მეცადინეობის თემები:

- კლასის პარამეტრებიანი კონსტრუქტორი
- კლასის კონსტრუქტორი, რომლის პარამეტრებს მინიჭებული აქვთ საწყისი მნიშვნელობები
- კონსტრუქტორის გადატვირთვა

მასალა დამოუკიდებლად გაცნობისათვის:

1. class –ში სხვადასხვა ტიპის კონსტრუქტორების გამოყენება

№	კლასის განსაზღვრა	კლასის ობიექტის შექმნა და მასთან მუშაობა

1	// განაკვეთი კვადრატის კლასზე
	class Square
	{
	public:
	double side;
	// პარამეტრიანი კონსტრუქტორი
	Square(double);
	void printSquare();
	};
	// კლასის ფუნქციების განსაზღვრა
	Square::Square(double G){
	side = G;
	}
	void Square::printSquare(){
	cout<<"side = "<<side<<endl;
	}
		int main()
		{
		Square C(5);
		C.printSquare();
		. . .
		}

2	// კვადრატის კლასში პარამეტრიანი
	// კონსტრუქტორის განსაზღვრა სხვა სახით
	class Square
	{
	public:
	double side;
	Square(double);
	void printSquare();
	};
	Square::Square(double G): side(G){ }
	void Square::printSquare(){
	cout<<"side = "<<side<<endl;
	}
		int main()
		{
		Square A1(5), A2(2);
		A1.printSquare();
		A2.printSquare();
		. . .
		}

3	// კონსტრუქტორის გადატვირთვა - კვადრატის
	// კლასი ორი კონსტრუქტორით
	class Square
	{
	public:
	double side;
	// კონსტრუქტორები
	Square( ); // უპარამეტრო
	Square(double); // პარამეტრიანი
	void printSquare();
	};
	// კლასის რეალიზების ნაწილი – ფუნქციების
		int main(){
		Square A;
		A.printSquare();
		Square B(7);
		B.printSquare();
		. . .
		}

	// განსაზღვრა
	Square::Square( ){
	  side = 10;
	}
	Square::Square(double G){
	  side = G;
	}
	void Square::printSquare(){
	  cout<<"side = "<<side<<endl;
	}	

4	// კვადრატის კლასში კონსტრუქტორების
	// განსაზღვრა სხვა სახით
	class Square
	{
	  public:
	  double side;
	  Square( ):side(10){ }
	  Square(double G):side(G){ }
	  void printSquare();
	};
	void Square::printSquare()
	{
	  cout<<"side = "<<side<<endl;
	}
		int main()
		{
		  Square A;
		  A.printSquare();
		  Square B(7);
		  B.printSquare();
		  . . .
		}

5	/* კვადრატის კლასში კონსტრუქტორის
	პარამეტრის ინიციალიზება. ასეთი კონსტრუქტორი
	ცვლის ორივე – უპარამეტრო და პარამეტრიან –
	კონსტრუქტორს */
	class Square
	{
	  public:
	  double side;
	  Square(double = 3);
	  void printSquare();
	};
	// კლასის ფუნქციების იმპლემენტაცია
	Square::Square(double G)
	{
	  side = G;
	}
	void Square::printSquare()
	{
	  cout<<"side = "<<side<<endl;
	}
		int main()
		{
		  Square A;
		  A.printSquare();
		  Square B(7);
		  B.printSquare();
		  Square C(20), D;
		  C.printSquare();
		  D.printSquare();
		  . . .
		}

6	// კონსტრუქტორის, რომლის პარამეტრსაც აქვს
	// გულისხმობით მნიშვნელობა, განსაზღვრა სხვა
	// სახით
	class Square
	{
	  public:
	  double side;
	  Square(double G = 3) : side(G){ }
	  void printSquare();
	};
	// კლასის ფუნქციის იმპლემენტაცია
		int main()
		{
		  Square A;
		  A.printSquare();
		  Square B(7);
		  B.printSquare();
		  Square C(20), D;
		  C.printSquare();
		  D.printSquare();
		  . . .
		}

	`void Square::printSquare()`
	`{`
	`  cout<<"side = "<<side<<endl;`
	`}`	

7	`// მწვერვალის კლასის აღწერა`
	`class Peak`
	`{`
	`  public:`
	`  string name;`
	`  int height;`
	`  Peak(string = "Ushba", int = 4695);`
	`};`
	`// კლასის რეალიზების ნაწილი`
	`Peak::Peak(string n, int h)`
	`{`
	`  name = n; height = h;`
	`}`
		`int main(){`
		`  Peak P1;`
		`  cout<<"Name : "`
		`  <<P1.name <<endl`
		`  <<"Height : "`
		`  <<P1.height`
		`  <<endl;`
		`  Peak P2("Sxara",5068);`
		`  cout<<"Name : "`
		`  <<P2.name`
		`  <<endl`
		`  <<"Height : "`
		`  <<P2.height`
		`  <<endl;`
		`  . . .`
		`}`

საკუთარი სამუშაო:

1. შექმენით მართკუთხედის კლასი ორი მონაცემებით: მართკუთხედის სიგრძე და სიგანე; კლასის ფუნქციებია: უპარამეტრო კონსტრუქტორი, პარამეტრებიანი კონსტრუქტორი, ფართობის და პერიმეტრის გამოთვლის ფუნქციები. პროგრამაში შექმენით კლასის ორი ობიექტი-მართკუთხედი, პირველი – უპარამეტრო კონსტრუქტორის, ხოლო მეორე – პარამეტრებიანი კონსტრუქტორის საშუალებით. შემდეგ დაბეჭდეთ ორივე მართკუთხედის გვერდები, პერიმეტრი და ფართობი.

```
#include <iostream>
using namespace std;
class Rectangle
{
   public:
     double length, width;
     Rectangle();
     Rectangle(double, double );
     double Area();
    double Perimeter();
    void printRectangle();
};
// კლასის ფუნქციების იმპლემენტაცია
Rectangle::Rectangle()
{
   length =3, width =4;
}
Rectangle::Rectangle(double a, double b)
{
   length =a, width =b;
}
double Rectangle::Area()
{
   return  length * width;
}
double Rectangle::Perimeter()
```

```
{
   return 2 * (length + width);
}
void Rectangle::printRectangle()
{
   cout<<"marTkuTxedis gverdebia ";
   cout<< length <<" "<< width <<endl;
   cout<<"farTobi = "<<Area()<<endl;
   cout<<"perimetri = "
       <<Perimeter()<<endl;
}
int main(){
   Rectangle A;
   cout<<"A ";
   A.printRectangle();
   Rectangle B(5,6);
   cout<<"B ";
   B.printRectangle();
   return 0;
}
```

პროგრამის შესრულების შედეგია:

```
A marTkuTxedis gverdebia 3 4
farTobi = 12
perimetri = 14
B marTkuTxedis gverdebia 5 6
farTobi = 30
perimetri = 22
Press any key to continue . . .
```

2. Rectangle კლასის აღწერაში ორივე კონსტრუქტორი შეცვალეთ ერთით – კონსტრუქტორით, რომლის პარამეტრებსაც მინიჭებული აქვთ მნიშვნელობები. პროგრამაში გააკეთეთ განაცხადი კლასის ორ ობიექტზე – მართკუთხედზე, დაბეჭდეთ მათი გვერდები, ფართობები და პერიმეტრი. შემდეგ შეცვალეთ პირველი მართკუთხედის სიგანე და მეორე მართკუთხედის ორივე გვერდი და კვლავ დაბეჭდეთ მართკუთხედების ფართობები და პერიმეტრები.

```
#include <iostream>
using namespace std;
class Rectangle
{
   public:
     double length, width;
     Rectangle(double =3, double =4);
     double Area();
     double Perimeter();
     void printRectangle();
};
// კლასის ფუნქციების იმპლემენტაცია
Rectangle::Rectangle(double a, double b)
{
   length =a, width =b;
}
double Rectangle::Area()
{
   return  length * width;
}
double Rectangle::Perimeter()
```

```
{
   return 2 * (length + width);
}
void Rectangle::printRectangle(){
  cout<<"marTkuTxedis gverdebia ";
  cout<< length <<" "<< width <<endl;
  cout<<"farTobi = "<<Area()<<endl;
  cout<<"perimetri = "
      <<Perimeter()<<endl;
}
int main()
{
   Rectangle A, B(5,7);
   cout<<"A ";
   A.printRectangle();
   cout<<"B ";
   B.printRectangle();
   A.width = 3.5;
   cout<<"A marTkuTxedis farTobi = "
       <<A.Area()<<endl
       <<"perimetri = "<<A.Perimeter()<<endl;
   B.length = 9;  B.width = 5.5;
   cout<<"B marTkuTxedis farTobi = "
       <<B.Area()<<endl
       <<"perimetri = "<<B.Perimeter()<<endl;
   return 0;
}
```

პროგრამის შესრულების შედეგია:

```
A marTkuTxedis gverdebia 3 4
farTobi = 12
perimetri = 14
B marTkuTxedis gverdebia 5 7
farTobi = 35
perimetri = 24
A marTkuTxedis farTobi = 10.5
perimetri = 13
B marTkuTxedis farTobi = 49.5
perimetri = 29
Press any key to continue . . .
```

3. შექმენით კლასი Sea მონაცემებით: ზღვის დასახელება და სიღრმე. კლასში გაითვალისწინეთ ფუნქციები: პარამეტრებიანი კონსტრუქტორი, ზღვის მონაცემების ბეჭდვის ფუნქცია. Seas.txt ფაილი შეიცავს ინფორმაციას სამი ზღვის შესახებ. პროგრამაში შექმენით კლასის სამი ობიექტი – Seas.txt ფაილის ინფორმაციის საფუძველზე და შემდეგ დაბეჭდეთ უდიდესი სიღრმის მქონე ზღვის მონაცემები.

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
class Sea
{
  public:
    string name;
    int depth;
    Sea(string n, int d) : name(n),depth(d){ }
    void printSea();
};
```

```
void Sea::printSea()
{
    cout<<name<<" Sea - "
        <<depth<<endl;
}
int main()
{
   string N; int D;
   ifstream fin("Seas.txt");
   fin>>N>>D;
   Sea A(N, D);
   fin>>N>>D;
   Sea B(N, D);
   fin>>N>>D;
   Sea C(N, D);
   cout<<"Maximum depth have\n";
   Sea max = A;
   if(B.depth > max.depth)
      max = B;
   if(C.depth > max.depth)
      max = C;
   max.printSea();
   cout<<endl;
   return 0;
}
```

პროგრამის შესრულების შედეგია:

Seas.txt ფაილი	გამოტანის ეკრანი
Black 2938 Grenlandiis 4846 Norwegian 3860	Maximum depth have Grenlandiis Sea - 4846 Press any key to continue . . .

4. იგივე ამოცანაში უდიდესი სიღრმის მქონე ზღვის არჩევა მოახდინეთ კლასის ტიპზე პოინტერის გამოყენებით.

მითითება: კლასის ტიპზე პოინტერს უნდა მივანიჭოთ ობიექტის მისამართი, ხოლო მიმთითებლის გამოყენებით კლასის მონაცემებზე და ფუნქციებზე წვდომისათვის უნდა ვისარგებლოთ **->** ოპერატორით.

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
class Sea
{
  public:
     string name;
     int depth;
     Sea(string n, int d) : name(n),depth(d){ }
      void printSea();
};
void Sea::printSea()
{
    cout<<name<<" Sea - "
      <<depth<<endl;
}
int main()
{
```

```
    string N; int D;
    ifstream fin("Seas.txt");
    fin>>N>>D;
    Sea A(N, D);
    fin>>N>>D;
    Sea B(N, D);
    fin>>N>>D;
    Sea C(N, D);

    cout<<"Maximum depth have\n";
    Sea * max = &A;
    if(B.depth > max->depth)
        max = &B;
    if(C.depth > max->depth)
        max = &C;
    max->printSea();
    cout<<endl;
    return 0;
}
```

დამოუკიდებელი სამუშაო:

1. კლასის საშუალებით შემოიღეთ მონაცემთა ტიპი, რომელიც აღწერს ქნლაქს. კლასში გაითვალისწინეთ: მონაცემები – ქნლაქის სახელი, მოსახლეობის რაოდენობა, დაკავებული ტერიტორიის თვრთობე; კლასის ფუნქციები – პარამეტრებიანი კონსტრუქტორი, ქნლაქის მონაცემების ბეჭდვის ფუნქცია. პროგრამაში გაავკეთეთ განაცხადი კლასის ორ ობიექტზე – ქნლაქზე, შემდეგ დაბეჭდეთ მეტი მოსახლეობის მქონე ქნლაქის ინფორმაცია.

2. შექმენით კლასი Triangle, რომლის მონაცემებია სამკუთხედის სამი გვერდი. კლასში იქონიეთ ფუნქციები – უპარამეტრო კონსტრუქტორი; პარამეტრებიანი კონსტრუქტორი; მონაცემების ბეჭდვის ფუნქცია; სამკუთხედის ფართობის გამოთვლის ფუნქცია. პროგრამაში გაავკეთეთ განაცხადი კლასის ორ ობიექტზე – სამკუთხედზე, დაბეჭდეთ მათი მონაცემები და შემდეგ შეადარეთ მათი ფართობები (დაბეჭდეთ სათანადო გზავნილი).

3. შექმენით კლასი Circle, რომლის მონაცემებია წრის ცენტრის კოორდინატები და რადიუსი. კლასში იქონიეთ ფუნქციები – უპარამეტრო კონსტრუქტორი; პარამეტრებიანი კონსტრუქტორი; მონაცემების ბეჭდვის ფუნქცია; წრეწირის სიგრძის გამოთვლის ფუნქცია. პროგრამაში გაავკეთეთ განაცხადი კლასის ორ ობიექტზე – წრეზე ისე, რომ პირველს ჰქონდეს გულისხმობით მინიჭებული მონაცემები, მეორეს კი – თქვენ მიერ კლავიატურიდან შემოტანილი მონაცემები. დაბეჭდეთ წრეების მონაცემები და შემდეგ შეადარეთ მათი წრეწირების სიგრძეები (დაბეჭდეთ სათანადო გზავნილი).

4. შექმენით კლასი Square. კლასში გაითვალისწინეთ: მონაცემი – კვადრატის გვერდი; კლასის ფუნქციები – კონსტრუქტორი, რომლის პარამეტრსაც მინიჭებულია აქვს საწყისი მნისვნელობა; კვადრატის ფართობის გამოთვლის ფუნქცია; კვადრატის პერიმეტრის გამოთვლის ფუნქცია; კვადრატის დიაგონალის გამოთვლის ფუნქცია. პროგრამაში აღწერეთ კლასის ორი ობიექტი ისე, რომ პირველს ჰქონდეს გულისხმობით მინიჭებული მონაცემი, მეორეს კი – კლავიატურიდან შემოტანილი მონაცემი. დაბეჭდეთ ორივე კვადრატის მონაცემი, ფართობი და პერიმეტრი. შემდეგ შეადარეთ კვადრატების დიაგონალები და დაბეჭდეთ სათანადო გზავნილი.

5. შექმენით კლასი Student შემდეგი მონაცემებით – სტუდენტის სახელი, გვარი და რეიტინგის აღმნიშვნელი ასო. კლასის ფუნქციებია – კონსტრუქტორი, რომლის პარამეტრებსაც მინიჭებულია აქვთ საწყისი მნისვნელობები; ინფორმაციის ბეჭდვის ფუნქცია. პროგრამაში აღწერეთ კლასის ორი ობიექტი ისე, რომ პირველს ჰქონდეს ნაგულისხმევი მონაცემები, მეორეს კი – თქვენ მიერ კლავიატურიდან შემოტანილი

მონაცემები. დაბეჭდეთ ორივე სტუდენტის მონაცემები. შეუცვალეთ პირველ სტუდენტს რეიტინგის მონაცემი და შემდეგ დაბეჭდეთ მის შესახებ ინფორმაცია.

6. შექმენით კლასი `Student` მონაცემებით: სტუდენტის სახელი, გვარი, ასაკი და სქესი. კლასში გაითვალისწინეთ თქვენი აზრით საჭირო ფუნქციები. დაწერეთ პროგრამა, რომელშიც 3 სტუდენტის მონაცემების შეტანა ხდება კლავიატურიდან და შემდეგ იბეჭდება:
   ა) სტუდენტთა შორის რამდენია გოგონა;
   ბ) ყველაზე უფროსი სტუდენტის გვარი-სახელი;
   გ) ყველაზე ახალგაზრდა ვაჟის ინფორმაცია. თუ სტუდენტებს შორის ვაჟი არ არის, დაიბეჭდოს სათანადო გზავნილი.

7. `rectangles.txt` ფაილში ჩაწერილია სამი მართკუთხედის მონაცემები – თითოეულის სიგრძე და სიგანე. დაწერეთ პროგრამა, რომელიც დაბეჭდავს უმცირესი ფართობის მქონე მართკუთხედის მონაცემებს და თვით ფართობს. ამისათვის შექმენით მართკუთხედის კლასი. კლასში გაითვალისწინეთ თქვენი აზრით საჭირო ფუნქციები. უმცირესი ფართობის მქონე მართკუთხედის დასადგენად გამოიყენეთ კლასის ტიპზე პოინტერი.

8. `data.txt` ფაილში ჩაწერილია სამი სახლის მონაცემები – თითოეულის სართულების რაოდენობა, სადარბაზოების რიცხვი და ყოველ სართულზე ბინების რაოდენობა. დაწერეთ პროგრამა, რომელიც `answer.txt` ფაილში დაბეჭდავს იმ სახლის სრულ ინფორმაციას, რომელშიც ყველაზე მეტი ბინაა. ამისათვის შექმენით სახლის კლასი, რომელშიც იქონიეთ თქვენი აზრით საჭირო ფუნქციები. ამოცანის ამოხსნისას გამოიყენეთ კლასის ტიპზე პოინტერი.

ლაბორატორიული მეცადინეობა № 13

ლაბორატორიული სამუშაოს ამოწერა:

- მრავალფაილიანი პროექტი (multifile project) კლასის მონაწილეობით

ლაბორატორიული სამუშაო:

ამოცანა 1. შექმენით მართკუთხედის კლასი ღია წევრებით. ველები: მართკუთხედის სიგრძე და სიგანე; მეთოდები: უპარამეტრო კონსტრუქტორი, პარამეტრებიანი კონსტრუქტორი, თვრთობის გამოთვლის ფუნქცია, პერიმეტრის გამოთვლის ფუნქცია.

პროგრამაში შექმენით კლასის ორი ობიექტი-მართკუთხედი, პირველი – უპარამეტრო კონსტრუქტორის, ხოლო მეორე – პარამეტრებიანი კონსტრუქტორის საშუალებით. შემდეგ დაბეჭდეთ ორივე მართკუთხედის მონაცემები.

ამოხსნა. პრაქტიკულ მეცადინეობებზე ჩვენ ვხსნიდით ასეთ ამოცანებს მარტივი ერთფაილიანი პროექტის სახით, ვათავსებდით რა განაცხადს კლასზე `main()`-ის წინ, ხოლო კლასის მეთოდების იმპლემენტაციას – კლასის განაცხადში ან კლასის გარეთ. ჩვენი ამოცანისთვის პროექტის ერთადერთ ფაილს ექნება ასეთი სახე:

```
#include <iostream>
using namespace std;
class Rectangle{
   public:
     double length, width;
     Rectangle();
     Rectangle(double, double );
     double Area();
     double Perimeter();
     void printRectangle();
};
// კლასის რეალიზების ნაწილი
Rectangle::Rectangle() {
   length =3, width =4;
}
Rectangle::Rectangle(double a, double b) {
   length =a, width =b;
}
double Rectangle:: Perimeter() {
   return  2*(length + width);
}
double Rectangle::Area() {
   return  length * width;
}
void Rectangle::printRectangle(){
  cout<<"marTkuTxedis gverdebia ";
  cout<< length <<" "<< width <<endl;
  cout<<"farTobi = "<<Area()<<endl;
  cout<<"perimetri = "<< Perimeter()<<endl;
}
int main(){
   Rectangle A;
   cout<<"A ";
   A.printRectangle();
   Rectangle B(5,6);
   cout<<"B ";
   B.printRectangle();
   return 0;
}
```

ასეთ მიდგომას აქვს რამდენიმე სერიოზული ნაკლი. ადვნიშნოთ ზოგიერთი:

- კოდი ნაკლებად დაცულია: კლასის სახელი, ცვლადების და ფუნქციების სახელები ნაკლებად კონტროლდება კომპილერის მიერ. მაგალითად, როდესაც ქვემოთ გავაკეთებთ კლასს Visual Studio-ს საშუალებებით, კომპილერი არ მოგვცემს `Rectangle` სახელის დარქმევის საშუალებას, რადგან ეს სახელი რეზერვირებულია (ასეთი კლასი უკვე არსებობს).
- როდესაც პრაქტიკული დირებულების მქონე ამოცანაში იქმნება კლასი, მისი ზომები და სირთულე ბევრად აღემატება აქ განხილული მაგალითებისას. ამ შემთხვევებში Visual Studio-ს გარემოს დახმარება შეუცვლელია. მით უმეტეს, რომ იგი გვეხმარება ცვლადებისა და ფუნქციების შექმნაში, თვითონ გაანაწილებს სხვადასხვა საქაღალდეში header და cpp ფაილებს და სხვა.

შევქმნათ ამ ერთფაილიანი პროექტის ეკვივალენტური მრავალფაილიანი პროექტი. ამისთვის განვახორციელოთ შემდეგი მოქმედებები:

ვთქვათ უკვე შექმნილი გვაქვს ცარიელი პროექტი სახელით A და საქაღალდეში Source Code გახსნილი გვაქვს .cpp ფაილი (მაგალითად) სახელით main. ამ ფაილის შექმნა ყველაზე ადვილია, საკმარისია

```
#include <iostream>
```

სტრიქონის შემდეგ დავამატოთ

```
#include "Rect.h"
```

რაც ნიშნავს, რომ ვამატებთ ჩვენს მიერ შექმნილი კლასის ინტერფეისს, რომელიც სხვა ფაილშია და იმიტომ ჩართვა სჭირდება.

ახლა შევქმნათ კლასი და მისი იმპლემენტაცია. ამაში ჩვენ მნიშვნელოვნად ვისარგებლებთ იმ ქალიბებით, რაც უკვე გამზადებულია Visual Studio-ს გარემოში პროგრამისტების დასახმარებლად.

მივიყვანოთ მაუსი პროექტის სახელთან "A" (იგივე Solution Exsplorer ფანჯარაში), დავაჭიროთ მაუსის მარჯვენა ღილაკს ხელი და გადავიდეთ პუნქტზე Add, შემდეგ ავირჩიოთ სტრიქონი Class. გაიხსნება ფანჯარა Add Class – A. Installed Templates სვეტში მოვნიშნოთ C++ სტრიქონი და დავაჭიროთ Add-ს (მარჯვენა ქვედა კუთხეში). გაიხსნება Generic C++ Class Wizard – A. ტექსტურ ველში სახელით Class Name ჩაწერეთ თქვენი კლასის სახელი Rect (შეგიძლიათ სცადოთ სრული სახელის Rectangle –ის ჩაწერა, რასაც ვერ შევძლებთ, რადგან ეს სახელი რეზერვირებულია). ტექსტურ ველებში სახელებით .h-file და .cpp-file ავტომატურად ჩაიწერება Rect.h და Rect.cpp. Access ჩამონათვალში დავტოვოთ გულისხმობის პრინციპით ჩაწერილი public და დავაჭიროთ Finish ღილაკს.

შედეგად, ჩვენს პროექტს დაემატა ორი ფაილი: Rect.h და Rect.cpp. პირველი შეიცავს Rect კლასზე განაცხადის ესკიზს, მომზადებული კონსტრუქტორის და დესტრუქტორის პროტოტიპებით, ხოლო მეორე ფაილი წარმოადგენს კლასის იმპლემენტაციას და აქაც არის კონსტრუქტორის და დესტრუქტორის იმპლემენტაციის ჩანასახები.

ახლა, კლასს დავამატოთ წევრები: ველები და მეთოდები (ანუ ცვლადები და ფუნქციები). Class View ფანჯარაში (რომელიც, თუ უკვე არაა გახსნილი, იხსნება View მენიუდან), მივიყვანოთ მაუსი პროექტის სახელზე "A", ჩამოვშალოთ კლასის სახელი Rect. ახლა მივიყვანოთ მაუსი კლასის სახელზე Rect, დავაჭიროთ მარჯვენა ღილაკს და ავირჩიოთ Add-> Add Variable. იხსნება Add Member Variable Wizard - A ფანჯარა. წვდომის (Access) ველში დავტოვოთ public. Variable Type ველში ავირჩიოთ მონაცემის ტიპი. იმ კლასისთვის, რომელსაც ჩვენ ვქმნით, გვჭირდება ორი ველი – ორი ნამდვილი ცვლადი, სიგრძე და სიგანე. შევქმნათ პირველი მათგანი. ჩამოვშალოთ ველი სახელით Variable Type და ავირჩიოთ ტიპი

double. ტექსტურ ველში, რომელიც გამოყოფილია ცვლადის სახელისთვის, ჩაწერით length. შეგვიძლია ამ ცვლადს დავუროთოთ კომენტარიც, ეს სავალდებულო არაა (როგორც გნებავთ). დავაჭიროთ დამთავრების (Finish) ღილაკს და დავათვალიეროთ ფაილებში Rect.h და Rect.cpp მომხდარი ცვლილებები.

ანალოგიური მოქმედებებით შევქმნათ ცვლადი

```
double width;
```

კვლავ დავათვალიეროთ ფაილებში Rect.h და Rect.cpp მომხდარი ცვლილებები. ყურადღება მივაქციოთ უპარამეტრო კონსტრუქტორის ცვლილებას.

ახლა შევქმნათ კლასის მეთოდები.

Class View ფანჯარაში მარჯვენა ღილაკი დავაჭიროთ კლასის სახელს, ავირჩიოთ Add-> Add Function. იხსნება Add Member Function Wizard - A ფანჯარა, რომელიც ინტუიციურად ადვილად გამოსაყენებელია. დავიწყოთ

```
double Area();
```

ფუნქციის შექმნით. დასაბრუნებელ მნიშვნელობად (Return type), ცხადია ავიღოთ double, ფუნქციის სახელად Area, არგუმენტები ამ ფუნქციას არ სჭირდება და ამიტომ დავაჭერთ დამთავრების ღილაკს.

თუ ახლა კვლავ დავათვალიერებთ Rect.h და Rect.cpp ფაილებს, ვნახავთ რომ კლასში შეიქმნა ფუნქციის პროტოტიპი, ხოლო იმპლემენტაციის ფაილში – იმპლემენტაციის სკიზი. დავასრულოთ ეს სკიზი. ამისთვის, ფუნქციის ტანი შევცვალოთ ისე, რომ ფუნქციამ მიიღოს სახე

```
double Rect::Area()
{
    return  length * width;
}
```

და დავიმახსოვროთ ცვლილებები.

ანალოგიურად გავაკეთოთ Perimeter ფუნქცია.

განსაკუთრებით საინტერესოა ცხადი (პარამეტრებიანი) კონსტრუქტორის შექმნა, რადგან მას არა აქვს დასაბრუნებელი მნიშვნელობა, მაგრამ აქვს არგუმენტები.

დავიწყოთ ისევ Add-> Add Function-ის არჩევით. Add Member Function Wizard - A ფანჯრის საშუალებით შევქმნათ

```
Rect::Rect(double a, double b)
{
    length =a, width =b;
}
```

ფუნქცია. დასაბრუნებელ მნიშვნელობის ველი (Return type) დავტოვოთ ცარიელი (ანუ წავშალოთ რაც იქ წერია). ფუნქციის სახელად ავიღოთ კლასის სახელი Rect. შევადგინოთ არგუმენტები (პარამეტრების) სია. დავიწყოთ პირველი პარამეტრიდან. პარამეტრის ტიპად ავიღოთ double, პარამეტრის სახელად a. ამ მომენტში გააქტიურდება Add ღილაკი, რადგან პირველი პარამეტრი უკვე მზადაა დასამატებლად. დავაჭიროთ ამ ღილაკს და ვნახავთ, რომ პირველი პარამეტრი გადავა მარჯვენა ფანჯარაში, რომელსაც ჰქვია პარამეტრების სია. ეს აგრეთვე ნიშნავს, რომ ტიპის და სახელის ველების გამოყენება უკვე შეიძლება მეორე პარამეტრისთვის. პარამეტრის ტიპად ავიღოთ double, პარამეტრის სახელად b.

თუ ახლაც დავათვალიეროთ ფაილებში Rect.h და Rect.cpp მომხდარი ცვლილებებს, ვნახავთ, რომ კლასში შეიქმნა კონსტრუქტორის პროტოტიპი, ხოლო იმპლემენტაციის ფაილში –

კონსტრუქტორის იმპლემენტაციის ესკიზი. დავასრულოთ ეს ესკიზი. ამისთვის, ფუნქციის ტანი შევცვალოთ ისე, რომ მივიღოთ

```
Rect::Rect (double a, double b)
{
    length =a, width =b;
}
```

printRectangle ფუნქცია იქმნება ანალოგიურად.
ამით კლასების შექმნა დასრულებულია. ახლა საჭიროა მთავარი main() ფუნქციის დაწერა. საბოლოოდ ჩვენი მრავალფაილიანი პროექტის ფაილები მიიღებს სახეს:

```
// Rect.h (header) ფაილი
#pragma once
class Rect
{
   public:
      Rect(void);
      ~Rect(void);
      double length;
      double width;
      double Area(void);
      double Perimeter(void);
      Rect(double a, double b);
      void printRectangle(void);
};
```

```
// Rect.cpp კლასის იმპლემენტაციის ფაილი
#include "Rect.h"
#include <iostream>
using namespace std;

Rect::Rect(void)
      : length(0)
      , width(0)
{
}

Rect::~Rect(void)
{
}

double Rect::Area(void)
{
      return length * width;
}

double Rect::Perimeter(void)
{
      return 2 * (length + width);
}

Rect::Rect(double a, double b)
{
      length =a, width =b;
}
```

```
void Rect::printRectangle(void)
{
      cout << "martkutxedis gverdebia ";
      cout << length << " " << width << endl;
      cout << "fartibi = " << Area() <<endl;
      cout << "perimetri = " << Perimeter() <<endl;
}

// main.cpp ფაილი
#include <iostream>
#include "Rect.h"
using namespace std;

int main()
{
      Rect A;
      cout << "A: ";
      A.printRectangle();

      Rect B(5,6);
      cout << "B: ";
      B.printRectangle();

      return 0;
}
```

დამოუკიდებელი სამუშაო:

მე –13 პრაქტიკული მეცადინეობის ყველა დავალებისთვის დაწერეთ, გამართეთ და შეასრულეთ შესაბამისი პროგრამები.