

PROGRAMMERS

Zadaća 4

Jašarević Ahmed

Zenica, juli 2022. godine

1 Nizovi - nastavak

1.1

Napisati program u kojem korisnik sa tastature unosi niz od 10 cijelih brojeva, a zatim se u unesenom nizu pronalazi i ispisuje minimum parnih brojeva.

Rješenje:

```
#include <iostream>

using namespace std;

int main()
{
    int niz[10], temp, i, j, min;
    cout << "Unesite elemente niza A: ";
    for (int i = 0; i < 10; i++){
        cin >> niz[i];
    }
    min = niz[0];
    for (int i = 0; i < 10; i++){
        if (niz[i] % 2 == 0){
            if (niz[i] < min)
                min = niz[i];
        }
    }
    cout << "Najmanji parni broj je: " << min;
    return 0;
}
```

Zadatak 1.1 (TEST)

```
Unesite elemente niza A: 1 2 3 4 5 6 7 8 9 10
Najmanji parni broj je: 1
```

1.2

Napisati program u kojem korisnik sa tastature unosi niz od 10 cijelih brojeva, a zatim se u unesenom nizu pronalazi i ispisuje maksimum neparnih brojeva

Rješenje:

```
#include <iostream>

using namespace std;

int main()
{
    int niz[10], temp, i, j, max;
    cout << "Unesite elemente niza A: ";
    for (int i = 0; i < 10; i++){
        cin >> niz[i];
    }
    max = niz[0];
    for (int i = 0; i < 10; i++){
        if (niz[i] % 2 == 0){
            continue;
        }
    }
}
```

```

        else{
            if(niz[i] > max)
                max = niz[i];
        }
    }
    cout << "Najveci neparni broj je: " << max;
    return 0;
}

```

Zadatak 1.2 (TEST)

```

Unesite elemente niza A: 1 2 3 4 5 6 7 8 9 10
Najveci neparni broj je: 9

```

1.3

Napisati program u kojem korisnik sa tastature unosi niz od 10 cijelih brojeva, a zatim se u unesenom nizu pronalazi i ispisuje srednja vrijednost brojeva na parnim pozicijama.

Rješenje:

```

#include <iostream>

using namespace std;

int main()
{
    int niz[10], temp, i, j, parniBrojac=0, suma=0;
    float srednjaVrijednost = 0.0;
    cout << "Unesite elemente niza A: ";
    for (int i = 0; i<10; i++){
        cin >> niz[i];
    }
    for (int i = 0; i<10; i++){
        if(niz[i] % 2 == 0){
            parniBrojac++;
            suma+=niz[i];
        }
    }
    srednjaVrijednost =suma / parniBrojac;
    cout << "Srednja vrijednost parnih brojeva je: " <<
    srednjaVrijednost;
    return 0;
}

```

Zadatak 1.3 (TEST)

```

Unesite elemente niza A: 1 2 3 4 5 6 7 8 9 10
Srednja vrijednost parnih brojeva je: 6

```

1.4

Napisati program u kojem korisnik sa tastature unosi niz od 10 cijelih brojeva, a zatim se na ekranu ispisuje razlike sume parnih i sume neparnih brojeva.

Rješenje:

```

#include <iostream>

```

```

using namespace std;

int main()
{
    int niz[10], i, suma=0, suma2=0;
    float srednjaVrijednost = 0.0;
    cout << "Unesite elemente niza A: ";
    for (int i = 0; i<10; i++){
        cin >> niz[i];
    }
    for (int i = 0; i<10; i++){
        if(niz[i] % 2 == 0){
            suma+=niz[i];
        }
        else {
            suma2+=niz[i];
        }
    }
    cout << "Razlika sume parnih i sume neparnih brojeva je: " << suma
    -suma2;
    return 0;
}

```

Zadatak 1.4 (TEST)

```

Unesite elemente niza A: 1 2 3 4 5 6 7 8 9 10
Razlika sume parnih i sume neparnih brojeva je: 5

```

1.5

Napisati program u kojem korisnik sa tastature unosi niz od 10 cijelih brojeva, a zatim se u unesenom nizu pronalazi i ispisuje srednja vrijednost neparnih brojeva.

Rješenje:

```

#include <iostream>

using namespace std;

int main()
{
    int niz[10], i, neparniBrojac=0, suma=0;
    float srednjaVrijednost = 0.0;
    cout << "Unesite elemente niza A: ";
    for (int i = 0; i<10; i++){
        cin >> niz[i];
    }
    for (int i = 0; i<10; i++){
        if(niz[i] % 2 == 0){
            continue;
        }
        else {
            neparniBrojac++;
            suma+=niz[i];
        }
    }
    srednjaVrijednost = suma /neparniBrojac;
    cout << "Srednja vrijednost neparnih brojeva je: " <<
    srednjaVrijednost;
}

```

```
    return 0;
}
```

Zadatak 1.5 (TEST)

```
Unesite elemente niza A: 1 2 3 4 5 6 7 8 9 10
Srednja vrijednost neparnih brojeva je: 5
```

1.6

Napisati program u kojem korisnik sa tastature unosi niz od 10 cijelih brojeva, a zatim se na ekranu ispisuje korijen proizvoda najvećeg i najmanjeg elementa.

Rješenje:

```
#include <iostream>
#include <math.h>
using namespace std;

int main()
{
    int niz[10], i, min, max;
    float rezultat;
    cout << "Unesite elemente niza A: ";
    for (int i = 0; i < 10; i++){
        cin >> niz[i];
    }
    min = niz[0];
    max = niz[0];
    for (int i = 0; i < 10; i++){
        if (niz[i] < min){
            min = niz[i];
        }
        if (niz[i] > max){
            max = niz[i];
        }
    }
    rezultat = sqrt(max*min);
    cout << "Korijen proizvoda najveceg i najmanjeg elementa u nizu je
: " << rezultat;

    return 0;
}
```

Zadatak 1.6 (TEST)

```
Unesite elemente niza A: 1 2 3 4 5 6 7 8 9 10
Korijen proizvoda najveceg i najmanjeg elementa u nizu je: 3.16228
```

1.7

Napisati program u kojem korisnik sa tastature unosi niz od 10 cijelih brojeva, a zatim se u unesenom nizu pronalazi i na ekranu ispisuje maksimalni element brojeva na parnim pozicijama.

Rješenje:

```
#include <iostream>
```

```
using namespace std;

int main()
{
    int niz[10], temp, i, j, max;
    cout << "Unesite elemente niza A: ";
    for (int i = 0; i < 10; i++){
        cin >> niz[i];
    }
    max = niz[0];
    for (int i = 0; i < 10; i++){
        if(niz[i] % 2 == 0){
            if(niz[i] > max)
                max = niz[i];
        }
    }
    cout << "Najveci parni broj je: " << max;
    return 0;
}
```

Zadatak 1.7 (TEST)

```
Unesite elemente niza A: 1 2 3 4 5 6 7 8 9 10
Najveci parni broj je: 10
```

1.8

Napisati program u kojem korisnik sa tastature unosi niz od 10 cijelih brojeva, a zatim se u unesenom nizu pronalazi i na ekranu ispisuje minimalni element brojeva na neparnim pozicijama.

Rješenje:

```
#include <iostream>

using namespace std;

int main()
{
    int niz[10], temp, i, j, min;
    cout << "Unesite elemente niza A: ";
    for (int i = 0; i < 10; i++){
        cin >> niz[i];
    }
    min = niz[0];
    for (int i = 0; i < 10; i++){
        if(niz[i] % 2 == 0){
            continue;
        }
        else{
            if(niz[i] < min)
                min = niz[i];
        }
    }
    cout << "Najmanji neparni broj je: " << min;
    return 0;
}
```

Zadatak 1.8 (TEST)

```
Unesite elemente niza A: 1 2 3 4 5 6 7 8 9 10
Najmanji neparni broj je: 1
```

1.9

Napisati program u kojem korisnik sa tastature unosi niz od 10 cijelih brojeva, a zatim se u unesenom nizu pronalazi i na ekranu ispisuje suma svih elemenata djeljivih sa 3 i nedjeljivih sa 9.

Rješenje:

```
#include <iostream>

using namespace std;

int main()
{
    int niz[10], temp, i, j, suma=0;
    cout << "Unesite elemente niza A: ";
    for (int i = 0; i<10; i++){
        cin >> niz[i];
    }
    for (int i = 0; i<10; i++){
        if(niz[i] % 3 == 0 && niz[i] % 9 != 0){
            suma += niz[i];
        }
    }
    cout << "Suma elemenata djeljivih sa 3 i nedjeljivih sa 9 je: " <<
    suma;
    return 0;
}
```

Zadatak 1.9 (TEST)

```
Unesite elemente niza A: 1 2 3 4 5 6 7 8 9 10
Suma elemenata djeljivih sa 3 i nedjeljivih sa 9 je: 9
```

1.10

Napisati program u kojem korisnik sa tastature unosi niz od 10 cijelih brojeva, a zatim se pronalazi suma parnih i suma neparnih brojeva pa se veća ispisuje.

Rješenje:

```
#include <iostream>

using namespace std;

int main()
{
    int niz[10], i, suma=0, suma2=0;
    float srednjaVrijednost = 0.0;
    cout << "Unesite elemente niza A: ";
    for (int i = 0; i<10; i++){
        cin >> niz[i];
    }
    for (int i = 0; i<10; i++){
        if(niz[i] % 2 == 0){
```

```

        suma+=niz[i];
    }
    else {
        suma2+=niz[i];
    }
}
if(suma > suma2){
    cout << "Suma parnih brojeva: " << suma;
}
else{
    cout << "Suma neparnih brojeva: " << suma2;
}
return 0;
}

```

Zadatak 1.10 (TEST)

```

Unesite elemente niza A: 1 2 3 4 5 6 7 8 9 10
Suma parnih brojeva: 30

```

1.11

Napisati program u kojem korisnik sa tastature unosi niz od 10 cijelih brojeva, a zatim nalazi srednju vrijednost brojeva djeljivih sa 3.

Rješenje:

```

#include <iostream>

using namespace std;

int main()
{
    int niz[10], i, suma=0, brojac=0;
    float srednjaVrijednost = 0.0;
    cout << "Unesite elemente niza A: ";
    for (int i = 0; i<10; i++){
        cin >> niz[i];
    }
    for (int i = 0; i<10; i++){
        if(niz[i] % 3 == 0){
            brojac++;
            suma += niz[i];
        }
    }
    srednjaVrijednost = suma / brojac;
    cout << "Srednja vrijednost brojeva djeljivih sa 3 je: " <<
    srednjaVrijednost;
    return 0;
}

```

Zadatak 1.11 (TEST)

```

Unesite elemente niza A: 1 2 3 4 5 6 7 8 9 10
Srednja vrijednost brojeva djeljivih sa 3 je: 6

```


1.12

Napisati program koji od korisnika traži unos 10 brojeva koje će ispisati u obrnutom redoslijedu od redoslijeda unosa. Elemente ispisivati razdvojene znakom zarez. Primjer ispisa:

Primjer:

```
Unesite 10 brojeva: 10 7 -3 7 2 46 1 2 1 0
0, 1, 2, 1, 46, 2, 7, -3, 7, 10,
```

Rješenje:

```
#include <iostream>

using namespace std;

int main()
{
    int niz[10], i;
    cout << "Unesite elemente niza A: ";
    for (int i = 0; i < 10; i++){
        cin >> niz[i];
    }
    for (int i = 9; i >= 0; i--){
        cout << niz[i] << ", ";
    }

    return 0;
}
```

Zadatak 1.12 (TEST)

```
Unesite elemente niza A: 1 2 3 4 5 6 7 8 9 10
10, 9, 8, 7, 6, 5, 4, 3, 2, 1,
```

1.13

Prepravite prethodni program tako da korisnik unosi n, broj elemenata niza (ne više od 50). Ukoliko je n izvan opsega ispisati poruku i zatražiti ponovni unos. Primjer ispisa:

Primjer:

```
Unesite broj elemenata niza: 55
Unos neispravna!
Unesite broj elemenata niza: 0
Unos neispravna!
Unesite broj elemenata niza: 5
Unesite 5 brojeva: 1 2 3 4 5
5, 4, 3, 2, 1,
```

Rješenje:

```
#include <iostream>

using namespace std;

int main()
{
    int niz[50], i, n;
    cout << "Unesite broj elemenata niza: ";
    cin >> n;
    while (n < 1 || n > 50){
```

```

        cout << "Unos neispravan !" << endl;
        cout << "Unesite broj elemenata niza: ";
        cin >> n;
    }
    cout << "Unesite " << n << " brojeva: ";
    for (int i = 0; i<n; i++){
        cin >> niz[i];
    }
    for (int i = n - 1; i>=0; i--){
        cout << niz[i] << ", ";
    }

    return 0;
}

```

Zadatak 1.13 (TEST)

```

Unesite broj elemenata niza: 3
Unesite 3 brojeva: 1 2 3
3, 2, 1,

```

1.14

Napisati program koji od korisnika traži unos n realnih elemenata niza (ne više od 100) koji će te elemente ispisati sortirane od manjeg ka većem unutar vitčastih zagrada, razdvojene znakom zarez (iza posljednjeg elementa nema znaka zarez). Ukoliko je broj n van opsega, ponovo tražiti unos.

Primjer:

```

Unesite broj elemenata niza: 7
Unesite 7 brojeva: 1 0 2.1 -2 5 7 3
{-2, 0, 1, 2.1, 3, 5, 7}

```

Rješenje:

```

#include <iostream>

using namespace std;

int main()
{
    float niz[100];
    int i,n;
    cout << "Unesite broj elemenata niza: ";
    cin >> n;
    while (n < 1 || n > 100){
        cout << "Unos neispravan !" << endl;
        cout << "Unesite broj elemenata niza: ";
        cin >> n;
    }
    cout << "Unesite " << n << " brojeva: ";
    for (int i = 0; i<n; i++){
        cin >> niz[i];
    }
    cout << "{";
    for (int i = n - 1; i>=0; i--){
        if(i == 0){
            cout << niz[i];
        }
        else {

```

```

        cout << niz[i] << ", ";
    }
}
cout << "}";

return 0;
}

```

Zadatak 1.14 (TEST)

```

Unesite broj elemenata niza: 4
Unesite 4 brojeva: 1 2 3 4
{4, 3, 2, 1}

```

1.15

Napisati program koji će od korisnika tržiti unos n realnih elemenata niza (ne više od 20) i koji će svaki elemenat niza zaokružiti na 2 decimale. Pri tome se misli na pravo zaokruživanje, a ne samo zaokruživanje prilikom ispisa. Tako modificiran niz ispisati sa preciznošću na 4 decimalna mjesta, unutar vitičastih zagrada.

Primjer:

```

Unesite broj elemenata niza: 7
Unesite 7 brojeva: 12.123 12.127 0.467 5 9.456 9.116 3.51
{12.1200, 12.1300, 0.4700, 5.0000, 9.5500, 9.1200, 3.5100}

```

Rješenje:

```

#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

int main()
{
    float niz[20];
    int i,n;
    cout << "Unesite broj elemenata niza: ";
    cin >> n;
    while (n < 1 || n > 20){
        cout << "Unos neispravan !" << endl;
        cout << "Unesite broj elemenata niza: ";
        cin >> n;
    }
    cout << "Unesite " << n << " brojeva: ";
    for (int i = 0; i<n; i++){
        cin >> niz[i];
    }
    cout << "{";
    for (int i = 0 ; i<n; i++){
        if(i == n-1){
            cout << setprecision(4) <<fixed << floor((niz[i]
]+0.005)*100.0) / 100.0;
        }
        else {
            cout << setprecision(4) << fixed << floor((niz[i]+0.005)
*100.0) / 100.0 << ", ";
        }
    }
    cout << "}";
}

```

```

    return 0;
}

```

Zadatak 1.15 (TEST)

```

Unesite broj elemenata niza: 4
Unesite 4 brojeva: 1 2 3 4
{1.0000, 2.0000, 3.0000, 4.0000}

```

1.16

Modificirati prethodni program tako da korisnik pored unosa elemenata niza unosi i broj decimala na koje želi zaokružiti elemente niza kao i broj decimala za ispis.

Primjer:

```

Unesite broj elemenata niza: 3
Unesite 3 brojeva: 123.45678 123.45477 12.3512
Unesite broj decimala za zaokruživanje: 3
Unesite broj decimala za ispis: 7
{123.4570000, 123.4550000, 12.3510000}

```

Rješenje:

```

#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

int main()
{
    float niz[20], decimala;
    int i, n, decimalaZaokruzivanje, decimalaIspis;
    cout << "Unesite broj elemenata niza: ";
    cin >> n;
    while (n < 1 || n > 20){
        cout << "Unos neispravan !" << endl;
        cout << "Unesite broj elemenata niza: ";
        cin >> n;
    }
    cout << "Unesite " << n << " brojeva: ";
    for (int i = 0; i < n; i++){
        cin >> niz[i];
    }
    cout << "Unesite broj decimala za zaokruživanje:";
    cin >> decimalaZaokruzivanje;
    cout << "Unesite broj decimala za ispis:";
    cin >> decimalaIspis;
    if(decimalaZaokruzivanje == 1){
        decimalaZaokruzivanje = 10.0;
    }
    if(decimalaZaokruzivanje == 2){
        decimalaZaokruzivanje = 100.0;
    }
    if(decimalaZaokruzivanje == 3){
        decimalaZaokruzivanje = 1000.0;
    }
    if(decimalaZaokruzivanje == 4){
        decimalaZaokruzivanje = 10000.0;
    }
    cout << "{";

```

```

    for (int i = 0 ; i<n; i++){
        if(i == n-1){
            cout << setprecision(decimalaIspis) <<fixed <<
            floor((niz[i]+0.0005)*decimalaZaokruzivanje) /
            decimalaZaokruzivanje;
        }
        else {
            cout << setprecision(decimalaIspis) << fixed << floor((
            niz[i]+0.0005)*decimalaZaokruzivanje) / decimalaZaokruzivanje << "
            ,";
        }
    }
    cout << "}";

    return 0;
}

```

Zadatak 1.16 (TEST)

```

Unesite broj elemenata niza: 2
Unesite 2 brojeva: 12.23 45.123
Unesite broj decimala za zaokruzivanje:3
Unesite broj decimala za ispis:7
{12.2300000 ,45.1230000}

```

1.17

Napisati program koji od korisnika traži unos n, broj elemenata niza (najviše 100). Nakon toga korisnik unosi elemente u niz, a program treba da iz niza izbaci sve parne elemente. Elemente izbacivati tako da se na njihovo mjesto pomjere ostali elementi. Program treba da ispiše novonastali niz kao i sumu njegovih elemenata.

Primjer:

```

Unesite broj elemenata niza: 10
Unesite 10 brojeva: 1 2 3 4 6 5 7 8 10 1
Niz je: 1, 3, 5, 7, 1,
Suma je: 17

```

Rješenje:

```

#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

int main()
{
    int niz[100];
    int i,n,j,suma=0;
    cout << "Unesite broj elemenata niza: ";
    cin >> n;
    while (n < 1 || n > 100){
        cout << "Unos neispravan !" << endl;
        cout << "Unesite broj elemenata niza: ";
        cin >> n;
    }
    cout << "Unesite " << n << " brojeva: ";
    for (int i = 0; i<n; i++){
        cin >> niz[i];
    }
}

```

```

    }
    for(i = 0; i < n; i++){
        if(niz[i] % 2 == 0){
            for(j=i; j < n - 1; j++){
                niz[j] = niz[j + 1];
            }
            n--;
            i--;
        }
    }
    cout << "Niz je: ";
    for(i = 0; i < n; i++){
        cout << niz[i] << " ";
        suma += niz[i];
    }
    cout << endl;
    cout << "Suma je: " << suma;
    return 0;
}

```

Zadatak 1.17 (TEST)

```

Unesite broj elemenata niza: 3
Unesite 3 brojeva: 1 2 3
Niz je: 1, 3,
Suma je: 4

```

1.18

Napisati program koji od korisnika traži unos n elemenata niza (ne više od 200), unos se prekida brojem -1 (koji se ne unosi u niz). Program nakon toga treba pronaći i ispisati najmanji i drugi najmanji broj u unesenom nizu.

Primjer:

```

Unesite 1. element (-1 za kraj unosa): 10
Unesite 2. element (-1 za kraj unosa): 5
Unesite 3. element (-1 za kraj unosa): 8
Unesite 4. element (-1 za kraj unosa): 1
Unesite 5. element (-1 za kraj unosa): 3
Unesite 6. element (-1 za kraj unosa): 8
Unesite 7. element (-1 za kraj unosa): -1
Najmanji broj je: 1
Drugi najmanji broj je: 3
Unesite 1. element (-1 za kraj unosa): 1
Unesite 2. element (-1 za kraj unosa): 1
Unesite 3. element (-1 za kraj unosa): 1
Unesite 4. element (-1 za kraj unosa): 1
Najmanji broj je: 1
Drugi najmanji broj je: 1

```

Rješenje:

```

#include <iostream>
#include <iomanip>
#include <math.h>
#include <limits.h>
using namespace std;

```

```

int main()
{
    int niz[200];
    int i,n,min,drugiMin,brojac=0;
    for( i=0; i<200; i++){
        cout << "Unesite " << i+1 << " element (-1 za kraj unosa): ";
        cin >> niz[i];
        if (niz[i] == -1) break;
        brojac++;
    }
    min = niz[0];
    for (i=0; i<brojac; i++){
        if(niz[i] < min){
            min = niz[i];
        }
    }
    drugiMin = INT_MAX;
    for (i=0; i<brojac; i++){
        if(niz[i] != min && niz[i] < drugiMin){
            drugiMin = niz[i];
        }
    }
    cout << "Najmanji broj je: " << min << endl;
    cout << "Drugi najmanji broj je: " << drugiMin << endl;
    return 0;
}

```

Zadatak 1.18 (TEST)

```

Unesite 1 element (-1 za kraj unosa): 1
Unesite 2 element (-1 za kraj unosa): 2
Unesite 3 element (-1 za kraj unosa): 3
Unesite 4 element (-1 za kraj unosa): -1
Najmanji broj je: 1
Drugi najmanji broj je: 2

```

1.19

Napisati program koji od korisnika traži unos najviše 100 elemenata (sa -1 se unos prekida). Program zatim od korisnika traži da unese neki broj n i ispituje da li je broj n među unesenim brojevima. Ako nije, treba ispisati odgovarajuću poruku, a ukoliko jeste, treba reći koji je po redu broj n unesen (gleda se samo prvi unos n broja).

Primjer:

```

Unesite 1. element (-1 za kraj unosa): 1
Unesite 2. element (-1 za kraj unosa): 5
Unesite 3. element (-1 za kraj unosa): 3
Unesite 4. element (-1 za kraj unosa): 7
Unesite 5. element (-1 za kraj unosa): 3
Unesite 6. element (-1 za kraj unosa): 7
Unesite 7. element (-1 za kraj unosa): -1
Unesite broj za pretragu: 7
Broj 7 je unesen kao 4 broj po redu

```

Rješenje:

```

#include <iostream>
#include <iomanip>

```

```

#include <math.h>
using namespace std;

int main()
{
    int niz[100];
    int i,n,element,brojac=0,index=1,flag=0;
    for( i=0; i<100; i++){
        cout << "Unesite " << i+1 << " element (-1 za kraj unosa): ";
        cin >> niz[i];
        if (niz[i] == -1) break;
        brojac++;
    }
    cout << "Unesite broj za pretragu: " ;
    cin >> element;
    for (i=0; i<brojac; i++){
        if(niz[i] == element){
            flag = 1;
            break;
        }
        index++;
    }
    if(flag == 0){
        cout << "Broj " << element << " se ne nalazi medju unesenim
        brojevima!";
    }
    else {
        cout << "Broj " << element << " je unesen kao " << index << ". " << "
        po redu.";
    }

    return 0;
}

```

Zadatak 1.19 (TEST)

```

Unesite 1 element (-1 za kraj unosa): 1
Unesite 2 element (-1 za kraj unosa): 2
Unesite 3 element (-1 za kraj unosa): 3
Unesite 4 element (-1 za kraj unosa): -1
Unesite broj za pretragu: 2
Broj 2 je unesen kao 2. po redu.

```

1.20

Prepraviti prethodni zadatak tako da ispisuje poziciju posljednjeg unesenog broja kojeg pretražuje. U primjeru 1 iz prethodnog zadaka za broj 7 bi trebalo ispisati da je unesen kao 6. broj po redu.

Rješenje:

```

#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

```



```

int main()
{
    int niz[100];
    int i,n,element,brojac=0,index=0,flag=0;
    for( i=0; i<100; i++){
        cout << "Unesite " << i+1 << " element (-1 za kraj unosa): ";
        cin >> niz[i];
        if (niz[i] == -1) break;
        brojac++;
    }
    cout << "Unesite broj za pretragu: " ;
    cin >> element;
    index=brojac;
    for (i=brojac-1; i>0; i--){
        if(niz[i] == element){
            flag = 1;
            break;
        }
        index--;
    }
    if(flag == 0){
        cout << "Broj " << element << " se ne nalazi medju unesenim
        brojevima!";
    }
    else {
        cout << "Broj " << element << " je posljednji unesen kao " << index <<
        ". " << " po redu.";
    }

    return 0;
}

```

Zadatak 1.20 (TEST)

```

Unesite 1 element (-1 za kraj unosa): 1
Unesite 2 element (-1 za kraj unosa): 2
Unesite 3 element (-1 za kraj unosa): 3
Unesite 4 element (-1 za kraj unosa): 4
Unesite 5 element (-1 za kraj unosa): 3
Unesite 6 element (-1 za kraj unosa): 3
Unesite 7 element (-1 za kraj unosa): -1
Unesite broj za pretragu: 3
Broj 3 je posljednji unesen kao 6. po redu.

```

1.21

Potrebno je napisati program koji će testirati simetričnost matrice po najvećim i najmanjim vandijagonalnim elementima. Matrica je simetrična ako ima jednake najveće elemente iz O1 i O4 oblasti i ako ima jednake najmanje elemente iz O2 i O3 oblasti (pogledati sliku). Korisnik sa tastature unosi cijeli broj n , koji predstavlja dimenziju matrice, te nakon toga unosi i elemente matrice formata $n \times n$. Unos napraviti takvim da se jedan red matrice unosi istovremeno (elementi se pri unosu razdvajaju razmacima – pogledati primjer). Uneseni broj n mora biti u intervalu $[3, 30]$, u suprotnom program ispisuje grešku i završava sa radom. Program, pored toga što treba ispitati simetričnost matrice, treba da ispiše najveće (odnosno najmanje) vandijagonalne elemente iz pojedinih oblasti (zaokruženo na 3 decimale).

Rješenje:

```
#include <limits.h>
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int n,i,j;
    float mat[30][30];
    cout<<"Unesite broj n: ";
    cin >>n;
    if(n < 3 || n > 30){
        cout << "Unos pogresan";
        return 0;
    }
    for(i=0; i<n; i++){
        cout << "Unesite elemente " << i+1 << " reda: ";
        for(j=0; j<n; j++){
            cin >> mat[i][j];
        }
    }
    float min = mat[0][0];
    float max1 = INT_MIN;
    float max = mat[0][0];
    float min1 = mat[0][0];
    for(i=0; i<n; i++){
        for(j=0; j<n; j++){
            if(i == j || i + j == n - 1){
                continue;
            }
            else{
                if(i + j < n - 1 && i > j){
                    if (mat[i][j] < min){
                        min = mat[i][j]; //lijevo
                    }
                }
                if(i + j > n - 1 && i < j){
                    if (mat[i][j] < min1){
                        min1 = mat[i][j]; //desno
                    }
                }

                if(i + j < n - 1 && i < j){
                    if (mat[i][j] > max){
                        max = mat[i][j]; //gore
                    }
                }
                if(i + j > n - 1 && i > j){
                    if (mat[i][j] > max1){
                        max1 = mat[i][j]; //dole
                    }
                }
            }
        }
    }
}
```

```

    }
}
if(min == min1 && max == max1){
    cout << "Matrica je simetricna!" << endl;
}
else{
    cout << "Matrica nije simetricna!" << endl;
}
cout << fixed << setprecision(3) << "Elementi: 01 = " << max << ",
02 = " << min << ", 03 = " << min1 << ", 04 = " << max1;

return 0;
}

```

Zadatak 1.21 (TEST)

```

Unesite broj n: 5
Unesite elemente 1 reda: 1 2 3 4 5
Unesite elemente 2 reda: 6 5 4 3 2
Unesite elemente 3 reda: 1 2 3 4 5
Unesite elemente 4 reda: 8 3 0 2 1
Unesite elemente 5 reda: 6 4 0 0 0
Matrica je simetricna!
Elementi: 01 = 4.000, 02 = 1.000, 03 = 1.000, 04 = 4.000

```

1.22

Napisati program koji od korisnika traži unos broja n (ne veći od 50) koji predstavlja dimenziju matrice formata $n \times n$. Program treba popuniti matricu tako da na glavnoj dijagonali budu nule, ispod dijagonale (paralelno sa glavnom dijagonalom) su elementi popunjeni sa 1, 2, ..., n , dok su elementi iznad glavne dijagonale (paralelno sa njom) popunjeni sa 1, 2, ..., n .

Primjer:

```

Unesite n: 5
0 1 2 3 4
-1 0 1 2 3
-2 -1 0 1 2
-3 -2 -1 0 1
-4 -3 -2 -1 0

```

Rješenje:

```

#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int n,i,j;
    float mat[50][50];
    cout<<"Unesite broj n: ";
    cin >>n;
    if(n < 1 || n > 50){
        cout << "Unos pogresan";
        return 0;
    }
    for(i=0; i<n; i++){

```

```

        cout << "Unesite elemente " << i+1 << " reda: ";
        for(j=0; j<n; j++){
            cin >> mat[i][j];
        }
    }
    for(i=0; i<n; i++){
        for(j=0; j<n; j++){
            if(i==j){
                mat[i][j] = 0;
                cout << mat[i][j] << '\t';
            }
            else if(i > j && (i + j) % 2 != 0){
                mat[i][j] = -1;
                cout << mat[i][j] << '\t';
            }
            else if(i > j && (i + j) % 2 == 0){
                mat[i][j] = -2;
                cout << mat[i][j] << '\t';
            }
            else if(i < j && (i + j) % 2 != 0){
                mat[i][j] = 1;
                cout << mat[i][j] << '\t';
            }
            else if(i < j && (i + j) % 2 == 0){
                mat[i][j] = 2;
                cout << mat[i][j] << '\t';
            }
            else {
                cout << mat[i][j] << '\t';
            }
        }
        cout << endl;
    }
    return 0;
}

```

Zadatak 1.22 (TEST)

```

Unesite broj n: 3
Unesite elemente 1 reda: 1 2 3
Unesite elemente 2 reda: 4 5 6
Unesite elemente 3 reda: 7 8 9
0      1      2
-1     0      1
-2     -1     0

```

1.23

Napisati program koji od korisnika traži unos dva niza (sa najviše od 50 elemenata), unos se prekida sa -1. Program u treći niz treba da smjesti elemente koji se javljaju i u prvom i u drugom nizu, bez ponavljanja. Konkretnije, ako nizove smatramo skupovima, program treba da nađe presjek dva skupa. Paziti da presjek može biti prazan skup, tj. da treći niz može imati nula elemenata. Također, iza posljednjeg elementa niza prilikom ispisa nema

znaka zarez.

Primjer:

```
Unesite elemente 1. niza (-1 za kraj): 1 2 3 4 4 5 6 7 7 8 9 -1
Unesite elemente 2. niza (-1 za kraj): 4 7 4 44 4 11 3 4 4 4 -1
Presjek unesenih nizova je: 3, 4, 7
```

Rješenje:

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int i,j,k,brojacA=0,brojacB=0;
    int A[50],B[50],C[50];
    cout << "Unesite elemente 1. niza (-1 za kraj): ";
    for(int i=0; i<50; i++){
        cin >> A[i];
        if(A[i] == -1) break;
        brojacA++;
    }
    cout << "Unesite elemente 2. niza (-1 za kraj): ";
    for(int i=0; i<50; i++){
        cin >> B[i];
        if(B[i] == -1) break;
        brojacB++;
    }
    for(i=0; i<brojacA-1; i++){
        for(j=i+1; j<brojacA; j++){
            if(A[i] == A[j]){
                for( k = j; k<brojacA; k++){
                    A[k] = A[k+1];
                }
                brojacA--;
                j--;
            }
        }
    }

    for(i=0; i<brojacB-1; i++){
        for(j=i+1; j<brojacB; j++){
            if(B[i] == B[j]){
                for( k = j; k<brojacB; k++){
                    B[k] = B[k+1];
                }
                brojacB--;
                j--;
            }
        }
    }

    cout << "Presjek unesenih nizova je: ";
    for(int i=0; i<brojacA; i++){
        for(int j=0; j<brojacB; j++){
            if(A[i] == B[j]){
                cout << A[i] << " ,";
            }
        }
    }
}
```

```

    return 0;
}

```

Zadatak 1.23 (TEST)

```

Unesite elemente 1. niza (-1 za kraj): 3 8 2 1 2 2 22 1 -1
Unesite elemente 2. niza (-1 za kraj): 3 4 2 3 2 2 3 2 -1
Presjek unesenih nizova je: 3 ,2 ,

```

1.24

Napisati program koji od korisnika zahtijeva unos dimenzije kvadratne matrice cijelih brojeva (dimenzija ne veće od 50), i elemente matrice. Pri tome osigurati da se matrica unese korektno. Program treba iz matrice izbaciti elemente na glavnoj dijagonali. Matricu treba modifikovati, a ne samo pri ispisu zanemariti elemente na dijagonali. Elemente matrice poravnati sa lijevom stranom i ostaviti 5 mjesta za ispis svakog broja. Primjer ispisa:

Primjer:

```

Unesite dimenziju matrice: 5
Unesite elemente 1. reda: 1 2 3 4 5
Unesite elemente 1. reda: 6 7 8 9 1
Unesite elemente 1. reda: 1 2 3 4 5
Unesite elemente 1. reda: 6 7 8 9 1
Unesite elemente 1. reda: 1 2 3 4 5
Matrica nakon izbacivanja diajgoanle
2 3 4 5
6 8 9 1
1 2 4 5
6 7 8 1
1 2 3 4

```

Rješenje:

```

#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int i,j,brojKolona,brojRedova;
    int mat[50][50];
    cout<<"Unesite broj n: ";
    cin >>brojRedova;
    brojKolona = brojRedova;
    if(brojRedova < 1 || brojRedova > 50){
        cout << "Unos pogresan";
        return 0;
    }
    for(i=0; i<brojRedova; i++){
        cout << "Unesite elemente " << i+1 << " reda: ";
        for(j=0; j<brojKolona; j++){
            cin >> mat[i][j];
        }
    }
    for(i=0; i<brojRedova; i++){
        for(j=i+1; j<brojKolona; j++){
            mat[i][j-1] = mat[i][j];
        }
    }
}

```

```

    }
}
brojKolona--;

for(i=0; i<brojRedova; i++){
    for(j=0; j<brojKolona; j++){
        cout << mat[i][j] << " ";
    }
    cout << endl;
}

return 0;
}

```

Zadatak 1.24 (TEST)

```

Unesite elemente 1 reda: 1 2 3
Unesite elemente 2 reda: 4 5 6
Unesite elemente 3 reda: 7 8 9
2      3
4      6
7      8

```

1.25

Modifikovati prethodni program tako da se, ukoliko je dimenzija matrice paran broj, iz matrice izbacuju i elementi sporedne dijagonale.

Primjer:

```

Unesite dimenziju matrice: 6
Unesite elemente 1. reda: 1 2 3 4 5 6
Unesite elemente 1. reda: 6 7 8 9 1 7
Unesite elemente 1. reda: 1 2 3 4 5 8
Unesite elemente 1. reda: 6 7 8 9 1 9
Unesite elemente 1. reda: 1 2 3 4 5 4
Unesite elemente 6. reda: 1 4 7 5 2 1
Matrica nakon izbacivanja diajgoanle
2 3 4 5
6 8 9 7
1 2 5 8
6 7 1 9
1 3 4 4
4 7 5 2

```

Rješenje:

```

#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int i,j,brojKolona,brojRedova,k1;
    int mat[50][50];
    cout<<"Unesite broj n: ";
    cin >>brojRedova;
    brojKolona = brojRedova;
    if(brojRedova < 1 || brojRedova > 50){

```

```

        cout << "Unos pogresan";
        return 0;
    }
    for(i=0; i<brojRedova; i++){
        cout << "Unesite elemente " << i+1 << " reda: ";
        for(j=0; j<brojKolona; j++){
            cin >> mat[i][j];
        }
    }
    int n = brojRedova;
    for(i=0; i<n; i++){
        int index = 0;
        for(j=0; j<n; j++){
            if(j != i && ((j+i) != (n-1))){
                mat[i][index] = mat[i][j];
                index++;
            }
        }
    }
    if(n % 2 == 0){
        cout << "Ispis matrice: " << endl;
        for(i=0; i<brojRedova; i++){
            for(j=0; j<brojKolona-2; j++){
                cout << mat[i][j] << " ";
            }
            cout << endl;
        }
    }
    else {
        cout << "Ispis matrice: " << endl;
        for(i=0; i<brojRedova; i++){
            for(j=0; j<brojKolona-1; j++){
                cout << mat[i][j] << " ";
            }
            cout << endl;
        }
    }
}

return 0;
}

```

Zadatak 1.25 (TEST)

```

Unesite broj n: 4
Unesite elemente 1 reda: 1 2 3 4
Unesite elemente 2 reda: 5 6 7 8
Unesite elemente 3 reda: 3 4 5 6
Unesite elemente 4 reda: 1 2 3 4
2      3
5      8
3      6
2      3

```


2 Funkcije - nastavak

2.1

Unesite sljedeći program:

```
#include <iostream>

using namespace std;

int f(int*a, int b) {
    int c;
    c = *a + b;
    *a = *a + 5;
    b = b + 5;
    return c;
}

int main()
{
    int a = 1, b = 1, c;
    cout<<"Prije funkcije f: a=" << a << ", b=" << b << endl;
    c = f(&a, b);
    cout<<"Poslije funkcije f: a=" << a << ", b=" << b << ", c=" << c
    << endl;
    return 0;
}
```

Rješenje:

1. Analizirajte program

F-ja kojoj se proslijeđuje adresa od a i varijabla b, te se a uvecava za 5, b ostaje isto, c je a + b

2. Kompajlirajte program.

3. Modifikujte funkciju f() i pokušajte predvidjeti šta će ispisati.

2.2

Korisnik sa tastature unosi osam realnih brojeva koji predstavljaju mjerenja temperature tokom jednog dana (temperatura se mjeri svaka 3 sata). Napišite program koji treba ispisati najveću temperaturu tokom jednog dana te prosječnu temperaturu.

Primjer:

```
Unesite temperaturu: 10 11 14 18 20 16 12 11
Maksimalna temperatura: 20.0
Prosjecna temperatura: 14.0
```

Rješenje:

```
#include <iostream>
#include <iomanip>

using namespace std;

float maxtemp(float p[8]){
    float max= p[0];
    for(int i=0; i<8; i++){
        if(p[i] > max){
```

```

        max = p[i];
    }
}
return max;
}

float prtemp(float p[8]){
    int suma = 0;
    for(int i=0; i<8; i++){
        suma += p[i];
    }
    return suma/8;
}

int main()
{
    float temperatura[8];
    cout <<"Unesite temperature: ";
    for(int i=0; i<8; i++){
        cin >> temperatura[i];
    }
    cout << fixed << setprecision(1) << "Maksimalna temperatura: " <<
        maxtemp(temperatura) << endl;
    cout << fixed << setprecision(1) << "Prosjecna temperatura: " <<
        prtemp(temperatura);
}

```

Zadatak 2.2 (TEST)

```

Unesite temperature: 2 4 6 8 10 12 14 16
Maksimalna temperatura: 16.0
Prosjecna temperatura: 9.0

```

2.3

Modificirati prethodni program tako da broj mjerenja nije konstantan (osam) nego se unosi sa tastature. Da bi se ove funkcije mogle koristiti i u drugim programima, trebale proslijediti funkcijama broj mjerenja kao parametar, npr. za funkciju maxtemp() prototip bi sada treba biti:.

```
float maxtemp(float p[], int br);
```

Rješenje:

```

#include <iostream>
#include <iomanip>
using namespace std;

float maxtemp(float p[], int br){
    float max= p[0];
    for(int i=0; i<br; i++){
        if(p[i] > max){
            max = p[i];
        }
    }
}

```

```

        return max;
    }

    float prtemp(float p[], int br){
        int suma = 0;
        for(int i=0; i<br; i++){
            suma += p[i];
        }
        return suma/br;
    }

    int main()
    {
        float temperatura[30], brojac=0;
        cout << "Unesite temperature (-1 za kraj unosa): ";
        for(int i=0; i<8; i++){
            cin >> temperatura[i];
            if(temperatura[i] == -1) break;
            brojac++;
        }
        cout << fixed << setprecision(1) << "Maksimalna temperatura: " <<
            maxtemp(temperatura, brojac) << endl;
        cout << fixed << setprecision(1) << "Prosjecna temperatura: " <<
            prtemp(temperatura, brojac);
    }

```

Zadatak 2.3 (TEST)

```

Unesite temperature (-1 za kraj unosa): 1 2 3 4 5 -1
Maksimalna temperatura: 5.0
Prosjecna temperatura: 3.0

```

2.4

Napišite funkciju f koja prima cijeli broj, a kao rezultat vraća sumu cifara tog broja.
Rješenje:

```

#include <iostream>

using namespace std;

int sumaCifara (int broj){
    int b,suma=0;
    while (broj != 0){
        b = broj % 10;
        suma = suma + b;
        broj = broj / 10;
    }
    return suma;
}

int main()
{
    int n;
    cout << "Unesite broj n: ";
    cin >> n;
    cout << "Suma cifara je: " << sumaCifara(n);

    return 0;
}

```

```
}
```

Zadatak 2.4 (TEST)

```
Unesite broj n: 123
Suma cifara je: 6
```

2.5

Napisati funkciju Ocisti bez parametara koja čisti ulazni tok nakon pogrešnog unosa. Pored toga, napisati funkciju Unesi koja unosi cijeli broj sa tastature. Ako unos nije ispravan (unesu se slova, realan broj), funkcija treba očistiti ulazni tok (pozivom funkcije Ocisti) te zatražiti ponovni unos. Konačno, napisati funkciju Prosjek koja prima cijeli broj, a vraća prosjek cifara prosljednog broja.

Primjer:

```
Unesite cijeli broj: osnove racunarstva
Unos pogresan!
Unesite cijeli broj: 324
Prosjek je: 3
```

Rješenje:

```
#include <iostream>

using namespace std;

void ocisti(){
    cout << "Unos pogresan" << endl;
    cin.clear();
    cin.ignore(256, '\n');
}

int Unesi(int n){
    cout << "Unesite broj n: ";
    cin >> n;
    while (!cin >> n){
        ocisti();
        cout << "Unesite broj n: ";
        cin >> n;
    }
    return n;
}

int prosjek(int broj){
    int b,suma=0,brojac=0;
    while (broj != 0){
        b = broj % 10;
        suma = suma + b;
        broj = broj / 10;
        brojac++;
    }
    return suma/brojac;
}

int main()
```

```
{
    int n, broj;
    broj = Unesi(n);
    cout << "Prosjek je: " << prosjek(broj);
    return 0;
}
```

Zadatak 2.5 (TEST)

```
Unesite broj n: ahmed
Unos pogresan
Unesite broj n: 123.23
Unos pogresan
Unesite broj n: 123
Prosjek je: 2
```

2.6

Napisati funkciju Ispisi koja ce ispisivati niz unutar vitičastih zagrada pri čemu su elementi razdvojeni znakom zarez. Funkcija prihvata dva parametra, niz (tačnije, pokazivač na prvi elemenat niza) i cijeli broj n koji predstavlja broj elemenata niza..

Primjer:

```
int niz1[] = {1, 3, 2},
int niz2[] = {0, 3, 2, 2, 2, 4, 10}
Ispisi (niz1, 3)
Ispisi (niz2, 10)
```

Rješenje:

```
#include <cmath>
#include <iostream>

using namespace std;

void
Ispisi (int niz[], int n)
{
    cout << "{";
    for (int i = 0; i < n; i++)
    {
        if (i == n - 1)
        {
            cout << niz[i] << "}";
        }
        else
        {
            cout << niz[i] << ",";
        }
    }
}

int main ()
{
    int niz1[] = { 1, 3, 2 };
    int niz2[10] = { 0, 3, 2, 2, 2, 4, 10};
```

```

Ispisi (niz1, 3);
cout << endl;
int count = 0;
int size1 = sizeof (niz2) / sizeof (niz2[0]);
for (int i = 0; i <= size1; i++)
{
    if (niz2[i] != '\0')
        count++;
}
Ispisi (niz2, count);

return 0;
}

```

Zadatak 2.6 (TEST)

```

Unesite elemente niza (-1 za kraj unosa): 1
2
3
-1
{1,2,3}

```

2.7

Napisati funkciju U dvostruci koja ce udvostručiti sve elemente niza proslijeđenog kao parametar. Funkcija ne vraća ništa, a modificira proslijeđeni joj niz. Parametri funkcije su identični kao i u prethodnom zadatku. Npr. za proslijeđeni niz 1, 0, 2 funkcija ga treba modificirati tako da on izgleda 2, 0, 4.

Napomena: Prilikom prosljeđivanja nizova funkcijama, sve izmjene koje funkcija napravi odraze se na originalni niz!

Rješenje:

```

#include <cmath>
#include <iostream>

using namespace std;

void Udvostruci (int niz[] , int n){
    cout << "{";
    for(int i = 0; i<n; i++){
        if(i == n - 1){
            niz[i] = niz[i] * 2;
            cout <<niz[i] << "}";
        }
        else{
            niz[i] = niz[i] * 2;
            cout <<niz[i] << ",";
        }
    }
}

void Ispisi (int niz[] , int n){
    cout << "{";
    for(int i = 0; i<n; i++){
        if(i == n - 1){

```

```

        cout << niz[i] << "}";
    }
    else{
        cout << niz[i] << ",";
    }
}
}

int main()
{
    int niz[50], brojac=0;
    cout << "Unesite elemente niza (-1 za kraj unosa): ";
    for(int i=0; i<50; i++){
        cin >> niz[i];
        if(niz[i] == -1) break;
        brojac ++;
    }
    Udvostruci(niz, brojac);
    Ispisi(niz, brojac);

    return 0;
}

```

Zadatak 2.7 (TEST)

```

Unesite elemente niza (-1 za kraj unosa): 1
2
3
-1
{2, 4, 6}

```

2.8

Napisati funkciju koja prihvata cijeli broj n , a vraća najveći stepen broja 2 koji je manji ili jednak broju n . Primjer, za $n = 35$ funkcija treba vratiti 5 jer je broj 32 najveći broj koji je stepen broja 2 ($2^5 = 32$) a koji je manji ili jednak n tj. od 35.

Rješenje:

```

#include <cmath>
#include <iostream>

using namespace std;

int stepen(int broj){
    int brojac=0;
    while(broj != 1){
        broj = broj / 2;
        brojac++;
    }
    return brojac;
}

int main()
{
    int n;

```

```

    cout << "Unesi cijeli broj n: ";
    cin >> n;
    cout << "Najveci stepen broja 2 je: " << stepen(n);
    return 0;
}

```

Zadatak 2.8 (TEST)

```

Unesi cijeli broj n: 64
Najveci stepen broja 2 je: 6

```

2.9

Napisati funkciju PrvaCifra koja za proslijedeni parametar n vraća prvu cifru toga broja. Npr. PrvaCifra (7392); treba da vrati broj 7. Predznak broja zanemariti.

Rješenje:

```

#include <cmath>
#include <iostream>

using namespace std;

int prvaCifra(int n){
    while(abs(n) >= 10){
        n /= 10;
    }
    n=abs(n);
    return n;
}

int main()
{
    int n;
    cout << "Unesi cijeli broj n: ";
    cin >> n;
    cout << "Prva cifra broja " << n << " je: " << prvaCifra(n);
    return 0;
}

```

Zadatak 2.9 (TEST)

```

Unesi cijeli broj n: 123
Prva cifra broja 123 je: 1

```

2.10

Napisati funkciju void Sortiraj (double niz[], int n) koja sortira elemente niza u opadajućem poretку korištenjem nekog od algoritama za sortiranje: Selection sort, bubble sort...

Rješenje:


```

#include <iostream>

using namespace std;

int temp,i,j;
void Sortiraj (double niz[], int n){
    for(i=0;i<n;i++){
        {
            for(j=i+1;j<n;j++){
                {
                    if(niz[i]<niz[j])
                    {
                        temp = niz[i];
                        niz[i] = niz[j];
                        niz[j]=temp;
                    }
                }
            }
        }
    }
}

int main()
{
    double  A[]={2,3,1,5};
    Sortiraj(A,4);
    for (i=0; i<4; i++){
        cout << A[i];
    }

    return 0;
}

```

Zadatak 2.10 (TEST)

```

Unesite elemente niza (-1 za kraj unosa): 0 5 4 3 9 8 -1
9 8 5 4 3 0

```

2.11

Napisati funkciju F ibonacci koja prihvata broj n i vraća n-ti Fibonaccijev broj. Npr. za n = 15 funkcija treba vratiti broj 610.

Rješenje:

```

#include <iostream>

using namespace std;

int Fibonaci (int n){
    if (n<= 1)
        return n;
    return Fibonaci(n-1)+ Fibonaci(n-2);
}

int main()
{
    int n;

```

```

    cout << "Unesi broj n: ";
    cin >> n;
    cout << Fibonaci(n);
    return 0;
}

```

Zadatak 2.11 (TEST)

```

Unesi broj n: 23
28657

```

2.12

Napisati funkciju `double Stepen (double b, int e)` koja vraća potenciju broja `b` e (`b` znači baza, `e` znači eksponent). Funkcija treba da radi i sa negativnim eksponentom.

Rješenje:

```

#include <cmath>
#include <iostream>

using namespace std;

double Stepen (double b, int e){
    return pow(b,e);
}

int main()
{
    double b,e;
    cout << "Unesi bazu: ";
    cin >> b;
    cout << "Unesite eksponent: ";
    cin >> e;
    cout << Stepen(b,e);
    return 0;
}

```

Zadatak 2.12 (TEST)

```

Unesi bazu: 3
Unesite eksponent: 3
27

```

2.13

Napisati funkciju `int DaLiJeStepenBroja2 (int n)` koja vraća logičku istinu (1) ako je broj `n` stepen broja 2 te logičku neistinu u suprotnom. Npr. za `n = 32` treba vratiti 1, a

za $n = 15$ treba vratiti 0.

Rješenje:

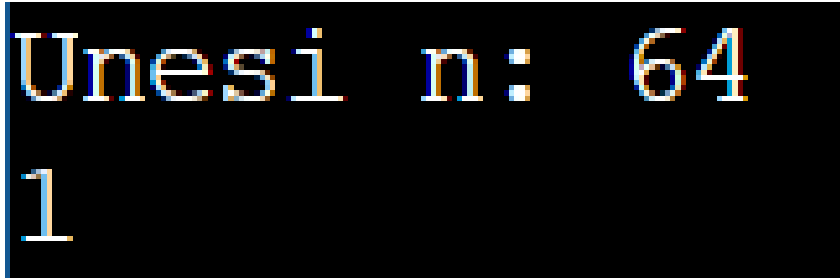
```
#include <cmath>
#include <iostream>

using namespace std;

int DaLiJeStepenBroj2 (int n){
    if(n>0){
        while(n % 2 == 0){
            n = n / 2;
        }
        if(n == 1){
            return 1;
        }
    }
    return 0;
}

int main()
{
    int n;
    cout << "Unesi n: ";
    cin >> n;
    cout << DaLiJeStepenBroj2(n);
    return 0;
}
```

Zadatak 2.13 (TEST)



```
Unesi n: 64
1
```

2.14

Napisati funkciju Armstrong koja za primljeni cjelobrojni parametar n ispituje da li je n Armstrongov broj, koji se definiše kao broj od k cifara koji je jednak suma svih cifara stepenovanih sa k . Npr. za broj 371 treba da vrati 1, jer 371 jeste Armstrongov broj ($3^3 + 7^3 + 1^3 = 371$), korišten je stepen $k = 3$ jer broj 371 ima 3 cifre. Još jedan primjer Armstrongovog broja je 1634 ($1^4 + 6^4 + 3^4 + 4^4 = 1634$), $k = 4$.

Rješenje:

```
#include <cmath>
#include <iostream>

using namespace std;

int Armstrong (int n){
    int rezultat = 0, ostatak, brojac=0, power;
    int praviBroj = n;
```

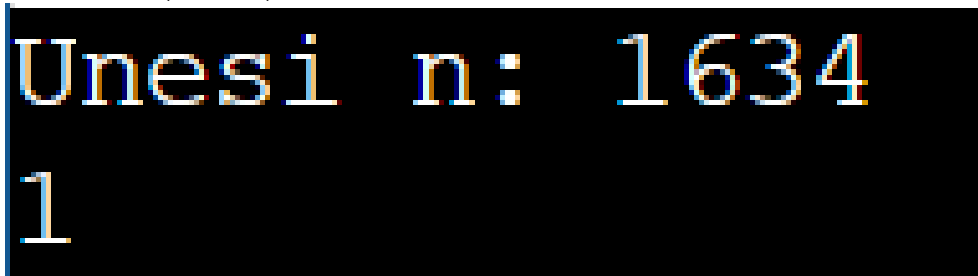
```

    while(praviBroj != 0){
        praviBroj /= 10;
        brojac++;
    }
    praviBroj = n;
    while (praviBroj != 0){
        ostatak = praviBroj % 10;
        power = round(pow(ostatak, brojac));
        rezultat += power;
        praviBroj /= 10;
    }
    if(rezultat == n){
        return 1;
    }
    else{
        return 0;
    }
    return 0;
}

int main()
{
    int n;
    cout << "Unesi n: ";
    cin >> n;
    cout << Armstrong(n);
    return 0;
}

```

Zadatak 2.14 (TEST)



```

Unesi n: 1634
1

```

2.15

Napisati funkciju `int NZD (int a, int b)` koja vraća najveći zajednički djelilac brojeva `a` i `b`. Npr. `NZD (8, 12)` je 4, `NZD (6, 366)` je 6. Napomena: funkcija je jako kratka ako se koristi rekurzija.

Rješenje:

```

#include <cmath>
#include <iostream>

using namespace std;

int NZD (int a, int b){
    int nzd;
    for(int i = 1; i<=a && i <= b; i++){
        if(a % i == 0 && b % i == 0){
            nzd = i;
        }
    }
    return nzd;
}

```

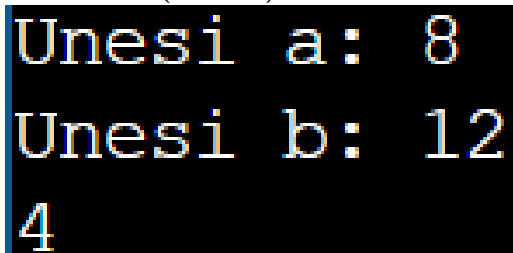
```

    }
}
return nzd;
}

int main()
{
    int a,b;
    cout << "Unesi a: ";
    cin >> a;
    cout << "Unesi b: ";
    cin >> b;
    cout << NZD(a,b);
    return 0;
}

```

Zadatak 2.15 (TEST)



```

Unesi a: 8
Unesi b: 12
4

```

2.16

Napisati funkciju `int Random (int a, int b)` koja vraća neki nasumični broj u intervalu $[a, b)$. Za generisanje slučajnih brojeva koristiti `rand` funkciju.

Rješenje:

```

#include <ctime>
#include <iostream>

using namespace std;

int Random (int a, int b){
    if(a > b){
        return b + rand() % ( a - b + 1 );
    }
    else{
        return a + rand() % ( b - a + 1 );
    }
}

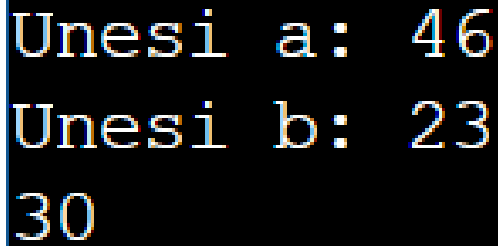
return 0;
}

int main()
{
    int a,b;
    cout << "Unesi a: ";
    cin >> a;
    cout << "Unesi b: ";
    cin >> b;
    cout << Random(a,b);
}

```

```
    return 0;  
}
```

Zadatak 2.16 (TEST)



```
Unesi a: 46  
Unesi b: 23  
30
```

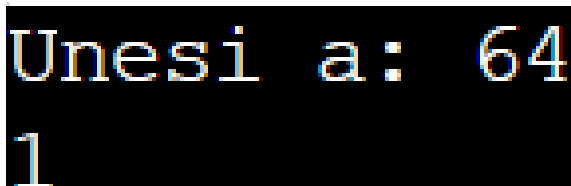
2.17

Napisati funkciju `int PotpunKvadrat (int a)` koja vraća vrijednost 1 ako je broj `a` potpun kvadrat (`a` je kvadrat nekog broja), a vraća vrijednost 0 u suprotnom. Na primjer, za broj 64 će biti vraćena vrijednost 1, jer je $64 = 8^2$. U slučaju da se funkciji proslijedi negativan broj, zanemariti predznak te nastaviti rad kao da je proslijeđen pozitivan broj.

Rješenje:

```
#include <cmath>  
#include <iostream>  
  
using namespace std;  
  
int PotpunKvadrat (int a){  
    if(a >= 0){  
        int kvadrat = sqrt(a);  
        return (kvadrat * kvadrat == a);  
    }  
    return false;  
}  
  
int main()  
{  
    int a;  
    cout << "Unesi a: ";  
    cin >> a;  
    cout << PotpunKvadrat(a);  
    return 0;  
}
```

Zadatak 2.17 (TEST)



```
Unesi a: 64  
1
```

2.18

Napisati funkciju `int Korijen (int a)` koja radi slično kao i funkcija `PotpunKvadrat` iz prethodnog zadatka, s tim da funkcija `Korijen` treba da vrati vrijednost korjena broja koji joj se šalje. Pri tome analizirati samo brojeve koji su potpuni kvadrati, a za ostale brojeve vratiti vrijednost `-1`.

Rješenje:

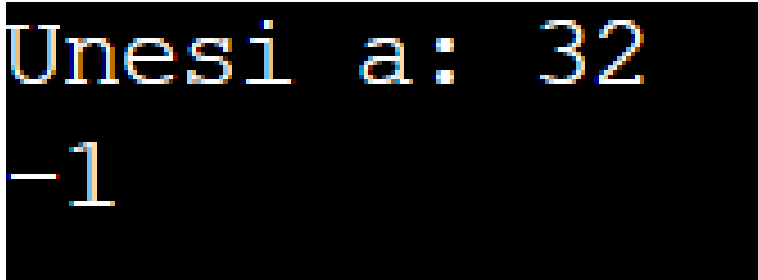
```
#include <cmath>
#include <iostream>

using namespace std;

int Korijen (int a){
    if(a >= 0){
        int kvadrat = sqrt(a);
        if (kvadrat * kvadrat == a)
            return sqrt(a);
    }
    return -1;
}

int main()
{
    int a;
    cout << "Unesi a: ";
    cin >> a;
    cout << Korijen(a);
    return 0;
}
```

Zadatak 2.18 (TEST)



```
Unesi a: 32
-1
```

2.19

Napisati funkciju koja prima parametre kvadratne jednačine (a , b , c), a kao rezultat vraća nule funkcije $f(x)$. Kvadratna jednačina je oblika: $f(x) = y = ax^2 + bx + c$

Nule funkcije tražiti numerički na nekom predefinisanoj intervalu (npr. od -10000 do 10000), a ne preko formule za X_1 i X_2 .

Rješenje:

```
#define epsilon 0.001
#include <iostream>
#include <math.h>
#include <complex>

using namespace std;
```

```

void kvadratnaJednacina (float a, float b , float c){
    int brojac = 0;
    float min = 100;
    float max = -100;
    for(float x = -100; x < 100; x+=0.0001){
        if(abs(a*x*x+b*x+c) < epsilon ){
            if(x < min) {
                min = x;
            }
            if(x > max) {
                max = x;
            }
        }
    }
}

int D = (b * b) - (4 * a * c);
if(min == 100 && max == -100){
    cout << "Rjesenja su imaginarna";
    cout << endl;
    cout << "x1: " << (-b/(2*a)) << " + " << sqrt(abs(D))/(2*a) << "i" << endl;
    cout << "x2: " << (-b/(2*a)) << " - " << sqrt(abs(D))/(2*a) << "i" << endl;
}
else if(max - min < 0.05){
    cout << "Nula f-je je: ";
    cout << endl;
    cout << "x1: " << max;
}
else{
    cout << "Nule f-je su: ";
    cout << endl;
    cout <<"x1: " << min << endl;
    cout << "x2: " << max;
}
}

int main()
{
    float a,b,c;
    cout << "Unesite a,b,c: ";
    cin>> a>> b>>c;
    if (a==0 && b == 0 && c == 0){
        cout << "Nije kvadratna jednacina";
        return 0;
    }
    if ( b == 0 && c == 0){
        cout << "Rjesenje je: 0";
        return 0;
    }
    kvadratnaJednacina(a,b,c) ;

    return 0;
}

```

Zadatak 2.19 (TEST)


```
Unesite a,b,c: 1
1
-1
Nule f-je su: -1.61835
0.617663
```

3 Ispisi

3.1

Šta će ispisati sljedeći programski isječak

```
int u slo v = 1 , v a r i j a b l a = 1 ;
2 while ( u slo v )
3 {
4 switch ( u slo v++%5)
5 {
6 case 1 :
7 u slo v+= v a r i j a b l a ;
8 break ;
9 case 2 :
10 u slo v =v a r i j a b l a ++;
11 break ;
12 case 3 :
13 u slo v = 0 ;
14 case 4 :
15 u slo v +=5;
16 break ;
17 de fau lt :
18 u slo v = 0 ;
19 }
20 p r i n t f ( %d %d\ n , u slo v , v a r i j a b l a ) ;
21 }
```

Rješenje:

```
0 1
```

3.2

Šta će ispisati sljedeći programski isječak

```
int i , j ;
2 int ni z [ 1 0 ] [ 1 2 ] = { { 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 1
0 } } ;
3
4 for ( i = 0 ; i < 1 0; i += 4 )
5 for ( j = 0 ; j < 1 2; j += 5 )
6 p r i n t f ( %d , ++ni z [ i ] [ j ] ) ;
```

Rješenje:

```
2700000000
```

3.3

Šta će ispisati sljedeći programski isječak?

```
int i , suma1 = 0 , suma2 = 0 ;
2 int ni z1 [ 5 ] = { 1 }, ni z2 [ 5 ] = { 1 };
3 for ( i = 0 ; i < 5 ; i++)
4 {
5 suma1+=ni z1 [ i ] ;
6 suma2+=ni z2 [ i ]++;
7 }
8
9 p r i n t f (      %d % d      , suma1 , suma2 ) ;
```

Rješenje:

```
0 -1
```

3.4

Šta će ispisati sljedeći programski isječak

```
1 int x = 1 , y = 2 ;
2 x=y++; p r i n t f (      %d %d\ n      , x , y ) ;
3 x=++y ; p r i n t f (      %d %d\ n      , x , y ) ;
4 x= y      ; p r i n t f (      %d %d\ n      , x , y ) ;
5 x= y      ; p r i n t f (      %d %d\ n      , x , y ) ;
6
7 x+=++y ; p r i n t f (      %d %d\ n      , x , y ) ;
8 x =y++; p r i n t f (      %d %d\ n      , x , y ) ;
9 x = y      ; p r i n t f (      %d %d\ n      , x , y ) ;
10 x+= y      ; p r i n t f (      %d %d\ n      , x , y ) ;
```

Rješenje:

```
2 3
3 3
2 2
5 3
2 4
-2 3
0 2
```

3.5

Šta će ispisati sljedeći programski isječak?

```
int x = E      , y = I      , z , w;
2 z = x      y++;
3 z += y ;
4 w = x++      y      ;
5 p r i n t f (      %d %d %d % d      , x , y , z , w ) ;
```

Rješenje:

```
70 73 70 -4
```

3.6

Šta će ispisati sljedeći programski isječak

```
.
```

3.7

Šta će ispisati sljedeći programski isječak?

```
#include <stdio.h>
2
3
4 int f ( int x ) {
5     return x + 2 ;
6 }
7
8
9 int main ( ) {
10     int x = 5 ;
11     printf (      %i %i %i %i      ,
12     f ( x ) ,
13     f ( x + 2 ) ,
14     f ( f ( x + 2 ) ) ,
15     f ( x + 2      f ( x + 2 ) + 2 ) ) ;
16     return 0 ;
17 }
```

Rješenje:

```
7 11 17 61
```

3.8

Šta će ispisati sljedeći programski isječak?

```
int f 2 ( int ) ;
2 int f 1 ( int a ) {
3     if ( a % 2 == 0 ) return a ;
4     f 2 ( a ) ;
5 }
6 int f 2 ( int a ) {
7     a      ;
8     return f 1 ( a ) ;
9 }
10 int main ( ) {
11     printf (      %d      , f 2 ( 5 ) ) ;
12 }
```

Rješenje:

3.9

Šta će ispisati sljedeći programski isječak

```

1 void f 2 ( int ) ;
2 void f 1 ( int a )
3 {
4   p r i n t f (    %d    , a ) ;
5   i f ( a > 15 ) return ;
6   a += 3 ;
7   f 1 ( a ) ;
8 }
9 void f 2 ( int a )

13

10 {
11   a    = 2 ;
12   f 1 ( a ) ;
13 }
14 int main ( )
15 {
16   f 2 ( 1 ) ;
17 }

```

Rješenje:

```

17

```

3.10

Šta će ispisati sljedeći programski isječak?

```

1 int f 1 ( int a )
2 {
3   a += 2 ;
4   return a ;
5 }
6 void f 2 ( int a )
7 {
8   a += 2 ;
9 }
10 int f 3 ( int a )
11 {
12   a++;
13   p r i n t f (    %d    , a ) ;
14   return a++;
15 }
16 int main ( )
17 {
18   int x = 5 ;
19   f 2 ( x ) ;
20   p r i n t f (    %d    , x ) ;

```

```
21 p r i n t f ( %d , f 1 ( x ) ) ;  
22 p r i n t f ( %d , f 3 ( x ) ) ;  
23 }
```

Rješenje:

```
7 9 10 10
```