



Gjuhë programuese C++

MSc. Vehbi NEZIRI


Java e 13-të

- ❑ Funkshonet rekurzive
- ❑ Dukshmëria e variablave
 - ❑ Variablat lokale
 - ❑ Variablat globale
- ❑ Kthimi i vlerave të shumta
- ❑ Thirrja sipas vlerës/referencës
- ❑ Fushat numerike si parametra të funksioneve

Funksionet rekurzive

- Funksionet mund të thirrin vetveten - funksione rekurzive.
- $fakt(n) = n * fakt(n-1)$ mund të shprehet lehtë nëpërmjet një zbatimi rekurziv.

```
int fakt(int numri)
{
    if (numri == 0)
        return 1;
    return numri * fakt(numri - 1);
}
```



Funksionet rekurzive

- Funksionet mund të thirrin vetveten - funksione rekurzive.
- $fakt(n) = n * fakt(n-1)$ mund të shprehet lehtë nëpërmjet një zbatimi rekurziv.

```
int fakt(int numri)
{
    if (numri == 0)
        return 1;
    return numri * fakt(numri - 1);
}
```

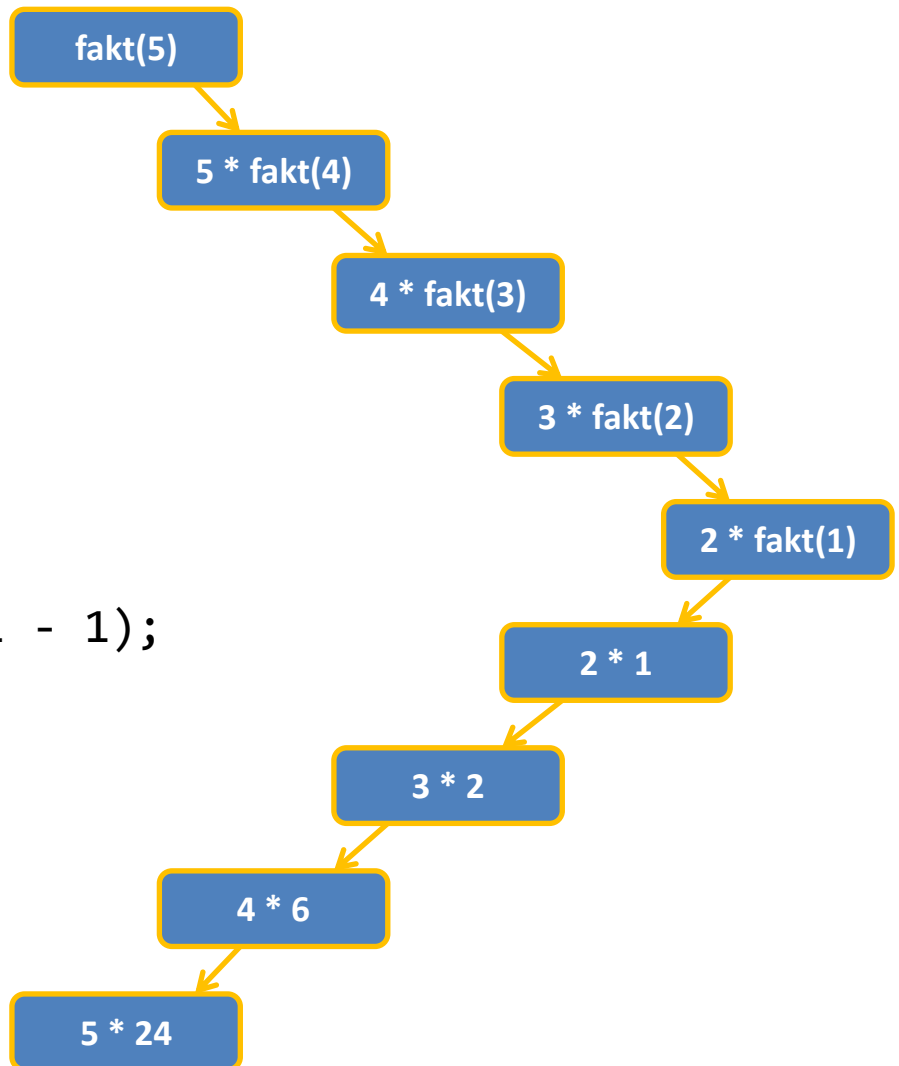


Hapi rekurziv

Funksionet rekurzive

Për numri=5

```
int fakt(int numri)
{
    if (numri == 0)
        return 1;
    return numri * fakt(numri - 1);
}
```



Shembull 1

- Të shkruhet programi për llogaritjen e shumës së numrave prej 1 deri në n . Llogaritja të bëhet përmes funksionit rekursiv të quajtur **shuma**.

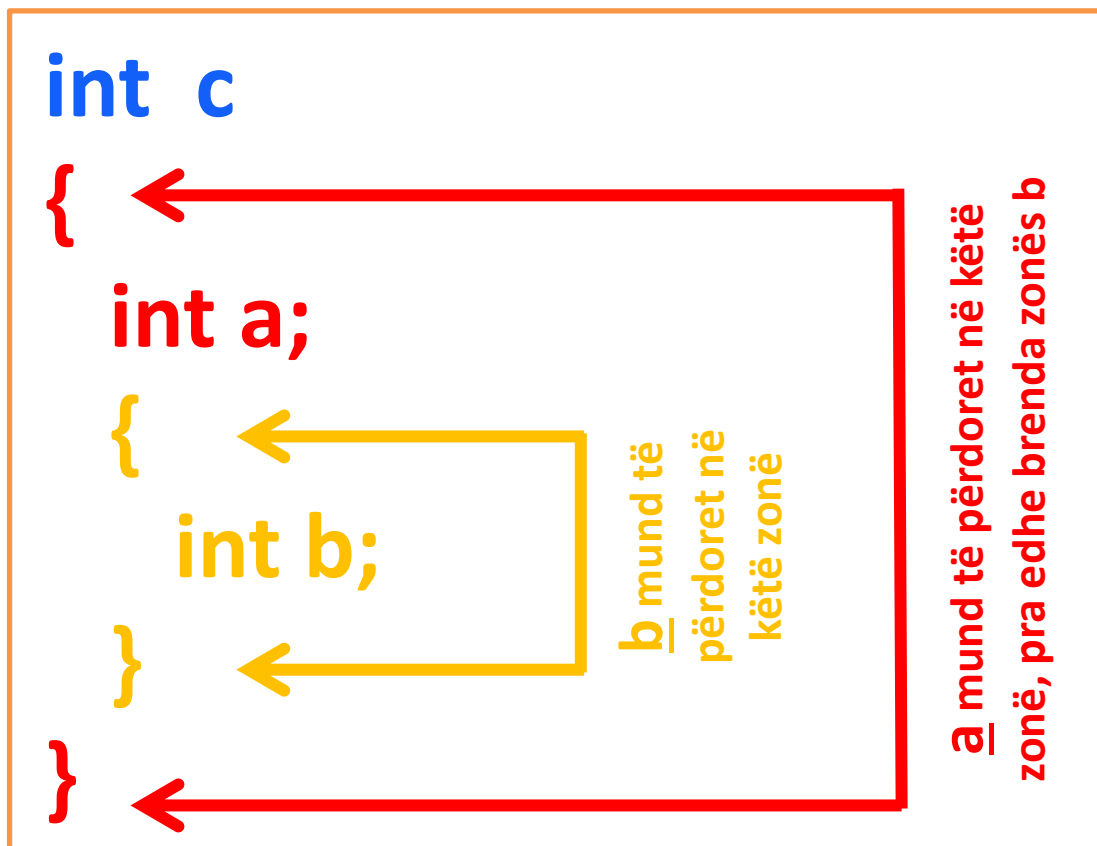
```
#include <iostream>
using namespace std;

int shuma (int numri)
{
    if (numri==0)
        return 0;
    return numri +shuma(numri-1);
}

int main()
{
    int numri=0, rez=0;
    cout<<"Shuma e numrave prej 1-n\n"
    <<"Jep vleren e n=";
    cin>>numri;
    rez=shuma(numri);
    cout<<"Shuma e numrave prej 1-"
    <<numri
    <<" eshte "
    <<rez
    <<endl;
    system("pause");
    return 0;
}
```

Dukshmëria (scope)

- Në varësi se ku deklarohen variablat, ato mund të jenë të disponueshme në fushëveprime të ndryshme (scope): **variabla lokale** dhe **variabla globale**.



Variablat lokale

- Variablat të cilat përcaktohen brenda një funksioni.

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

```
int main()
{
```

```
    short moshja;
    float numri1, numri2;
```

```
    cout<<"Moshja e juaj? ";
    cin>>moshja;
    ...
```

```
}
```



Variablat lokale

Variablat globale

- Variabla të cilat deklarohen jashtë funksioneve.

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

```
int femije;
char karakteri;
```

```
int main()
{
```

```
    short mosha;
    float numri1, numri2;
```

```
    cout<<"Mosha e juaj? ";
    cin>>mosha;
    ...
```

```
}
```



Variabla globale



Variabla lokale

Shembull 2

- Të shkruhet programi i cili mbledh dy numra të plotë përmes funksionit `shuma`. Të bëhet numërimi sa herë po thirret funksioni duke shfrytëzuar variablen globale `thirrja`.

```
Shuma e a+b=7
Shuma e a+b=11
Shuma e a+b=5
Funksioni shuma është thirrur 3 herë
```

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

int thirrja=0;

void shuma(int a, int b)
{
    thirrja++;
    cout<<"Shuma e a+b="
    <<a+b<<endl;
}

int main()
{
    shuma(3,4);
    shuma(5,6);
    shuma(-4,9);
    cout<<"Funksioni shuma është
    thirrur " <<thirrja <<" herë\n";
    system("pause");
    return 0;
}
```

Shembull 3

- Të shkruhet programi i cili përmes funksionit shuma llogarit **shumën**, **prodhimin** dhe **herësin** e numrave **a** dhe **b**. Prodhimi dhe herësi të ruhen në variablat globale, prodhimi respektivisht heresi.

```
float shuma(float a, float b)
{
    .
    .
    .
    return a+b;
}
```

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

```
float prodhimi=0.0;
float heresi=0.0;
```

```
float shuma(float a, float b)
{
    prodhimi=a*b;
    heresi=a/b;
    return a+b;
}
```

```
int main()
{
    cout<<"Shuma e a+b= " <<shuma(12,5);
    cout<<"\nProdhimi i a*b= " <<prodhimi;
    cout<<"\nHeresi i a/b= "
    <<heresi<<endl;
    system("pause");
    return 0;
}
```

Thirrja sipas vlerës/referencës

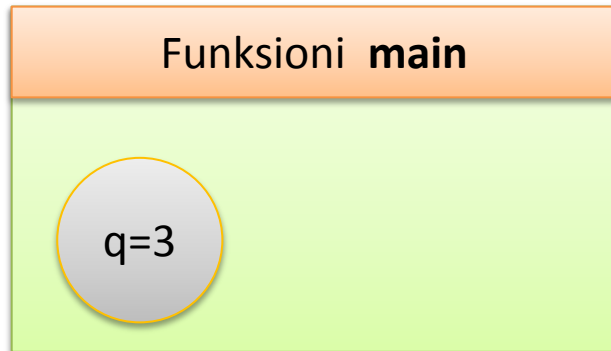
- Deri me tani funksionet i kemi thirrur sipas vlerës.
- Është krijuar kopja e variables dhe ndryshimet nuk janë përcjellur jashtë funksionit

```
//thirrja sipas vlerës
void increment(int a)
{
    a = a + 1;
    cout << "a e inkrementuar " << a << endl;
}
int main()
{
    int q = 3;
    increment(q);
    cout << "q ne main " << q << endl;
}
```

Dalja

```
a e inkrementuar 4
q ne main 3
```

Thirrja sipas vlerës/referencës

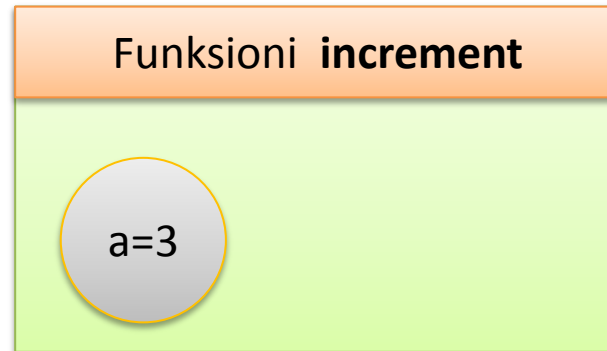
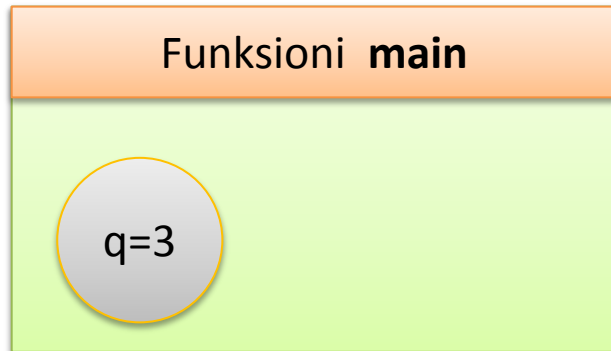


```
//thirrja sipas vlerës
void increment(int a)
{
    a = a + 1;
    cout << "a e inkrementuar " << a << endl;
}
int main()
{
    int q = 3; //KËTU
    increment(q);
    cout << "q ne main " << q << endl;
}
```

Dalja

a e inkrementuar 4
q ne main 3

Thirrja sipas vlerës/referencës

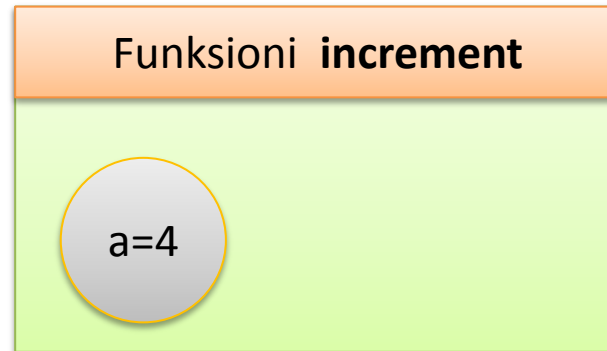
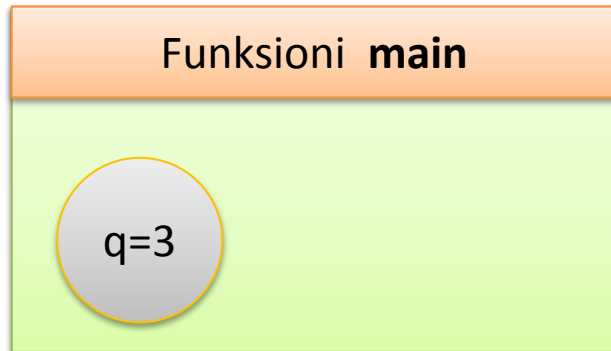


```
//thirrja sipas vlerës
void increment(int a) //KËTU
{
    a = a + 1;
    cout << "a e inkrementuar " << a << endl;
}
int main()
{
    int q = 3;
    increment(q);
    cout << "q ne main " << q << endl;
}
```

Dalja

a e inkrementuar 4
q ne main 3

Thirrja sipas vlerës/referencës



```
//thirrja sipas vlerës
void increment(int a)
{
    a = a + 1; //KËTU
    cout << "a e inkrementuar " << a << endl;
}
int main()
{
    int q = 3;
    increment(q);
    cout << "q ne main " << q << endl;
}
```

Dalja

a e inkrementuar 4
q ne main 3

Thirrja sipas vlerës/referencës

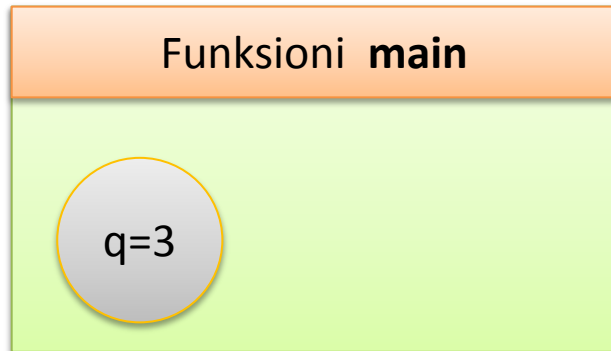
- Nëse dëshirojmë të modifikojmë variablen origjinale në vend se të bëhet kopje, funksioni thirret sipas referencës (**int &a** në vend të **int a**)

```
//thirrja sipas referencës
void increment(int &a)
{
    a = a + 1;
    cout << "a e inkrementuar " << a << endl;
}
int main()
{
    int q = 3;
    increment(q);
    cout << "q ne main " << q << endl;
}
```

Dalja

```
a e inkrementuar 4
q ne main 4
```


Thirrja sipas vlerës/referencës

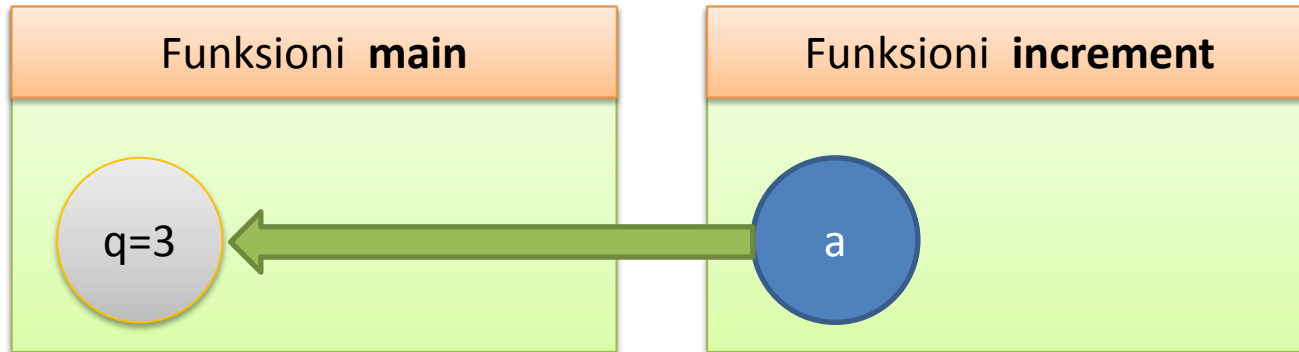


```
//thirrja sipas referencës
void increment(int &a)
{
    a = a + 1;
    cout << "a e inkrementuar " << a << endl;
}
int main()
{
    int q = 3; //KËTU
    increment(q);
    cout << "q ne main " << q << endl;
}
```

Dalja

```
a e inkrementuar 4
q ne main 4
```

Thirrja sipas vlerës/referencës

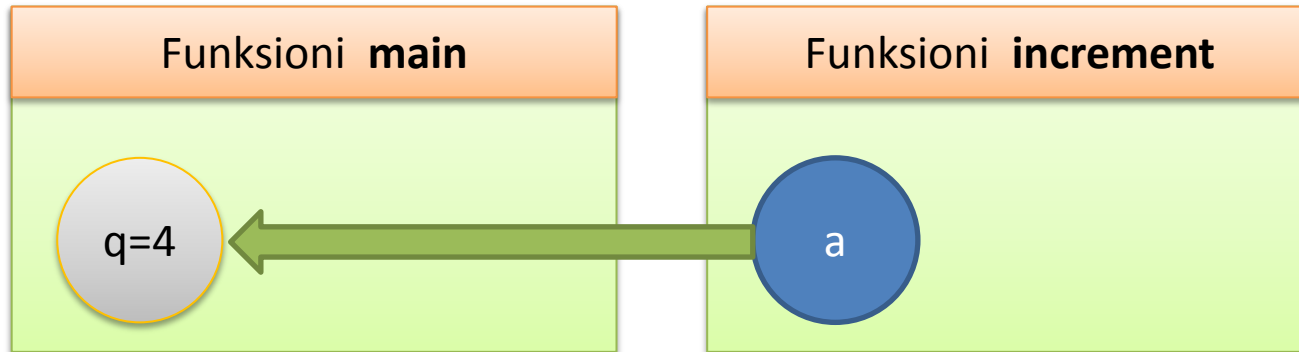


```
//thirrja sipas referencës
void increment(int &a) //KËTU
{
    a = a + 1;
    cout << "a e inkrementuar " << a << endl;
}
int main()
{
    int q = 3;
    increment(q);
    cout << "q ne main " << q << endl;
}
```

Dalja

a e inkrementuar 4
q ne main 4

Thirrja sipas vlerës/referencës



```
//thirrja sipas referencës
void increment(int &a)
{
    a = a + 1; //KËTU
    cout << "a e inkrementuar " << a << endl;
}
int main()
{
    int q = 3;
    increment(q);
    cout << "q ne main " << q << endl;
}
```

Dalja

```
a e inkrementuar 4
q ne main 4
```

Shembull 4

- Të shkruhet programi për shkëmbimin e vlerave të dy variablave **q** dhe **r** duke shfrytëzuar funksionin **shkembe**. Prototipi i funksionit është.

```
void shkembe(int &a, int &b)
```

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

void shkembe(int &a, int &b)
{
    int t = a;
    a = b;
    b = t;
}

int main()
{
    int q = 3, r = 5;
    cout<< "Vlerat para shkembimit \nq
    = " << q
    << "\nr = " << r << endl << endl;
    shkembe(q, r);
    cout << "Vlerat pas shkembimit\n"
    << "q = " << q
    << "\nr = " << r << endl;
    system("pause");
    return 0;
}
```

Fushat numerike në funksione

- Fusha numerike si parametër i funksionit

```
int fusha[10];  
funksioni(fusha, 10);
```

Fusha numerike

Madhësia e fushës

Emri i fushës pa kllapa

- Prototipi i funksionit

```
void funksioni(int f[], int madhesia)
```

- Emri i parametrut është opsional

`int f[]` mund të shkruhet `int []`

`int madhesia` mund të jetë vetëm `int`

Shembull 5

- Të krijohet vektori $A(m)$ duke shfrytëzuar funksionin `krijo_vektor`. Anëtarët e vektorit të mbushen sipas shprehjes së $2*i+1$. Më pas të shtypen anëtarët e vektorit të krijuar.

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

const int m=5
void krijo_vektor(int v[], int m)
{
    for (int i = 0; i < m; i++)
        v[i] =2*i+1;
}

int main()
{
    int A[m];
    krijo_vektor(A,m);
    cout<<"Vektori i krijuar: "
    <<"A["<<m<<"] = {";
    for (int i=0;i<m;i++)
    {
        cout<<A[i];
        if (i<m-1)
            cout<<", ";
    }
    cout << "}\n";
    return 0;
}
```

Shembull 6

- Të krijohet matrica $A(m,n)$ duke shfrytëzuar funksionin `krijo_matrice`. Anëtarët e matrices të mbushen sipas shprehjes së mëposhtme dhe më pas të shtypen anëtarët e matricës së krijuar.

$$A[i][j] = \begin{cases} i+j & i < j \\ 0 & i = j \\ i-j & i > j \end{cases}$$

```
#include <iostream>
using namespace std;
const int m=5, n=6;

void krijo_matrice(int A[m][n])
{
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++)
        {
            if(i<j)
                A[i][j] =i+j;
            else if (i==j)
                A[i][j] =0;
            else
                A[i][j] =i-j;
        }
}

void printo_matrice(int M[m][n])
{
    cout<<"Matrica e krijuar\n";
    for (int i=0;i<m;i++)
    {
        for (int j=0;j<n;j++)
        {
            cout << M[i][j];
            cout.width(5);
        }
        cout << "\n";
    }
}

int main()
{
    int A[m][n];
    krijo_matrice(A);
    printo_matrice(A);
    system("pause");
    return 0;
}
```

Detyra: 13.1

- Të shkruhet programi për numërimin e numrave tek dhe çift nga m deri në n duke shfrytëzuar funksionin `tek_cift`.

```
void tek_cift(int prej, int deri)
```

```
Numra tek prej 100 deri 1000 jane 451  
Numra cift prej 100 deri 1000 jane 450
```


Detyra: 13.2

- Të shkruhet programi për shkëmbimin e vlerave të dy variablave **q** dhe **r** duke shfrytëzuar funksionin **shkembe_vlerat**, por duke mos përdorur variabla ndihmëse.

```
Vlerat para shkembimit
q = 3
r = 5

Vlerat pas shkembimit
q = 5
r = 3
```

Detyra: 13.3

- Të shkruhet funksioni `minimumi` për gjetjen e anëtarit të vektorit me vlerë minimale.

```
#include <iostream>
using namespace std;
const int m=8;
```

Funksioni `minimumi`

```
int main()
{
    int a[m] = {3,7,4,-6,2,-3,9,1};
    cout<< "Vlera minimale e vektorit: "
    <<minimumi(a, m) << endl;
    system("pause");
    return 0;
}
```

Detyra: 13.4

- Të shkruhet programi për mbushjen e vektorit me numra të rastit nga 1 deri në 100. Vektori të sortohet dhe më pas të shtypen anëtarët e vektorit të pasortuar dhe të sortuar. Mbushja, sortimi dhe shtypja të realizohen duke shfrytëzuar funksionet.

```
Vektori : 1 A[8] = <99, 97, 43, 76, 37, 59, 67, 60>  
Vektori : 2 A[8] = <37, 43, 59, 60, 67, 76, 97, 99>
```

Detyra: 13.5

- Të shkruhet programi për mbushjen e matricës me numra të rastit nga 1 deri në 100. Më pas të gjendet anëtari më i madh në secilin rresht. Matrica të shtypet duke shfrytëzuar funksionet.

```
31 31 55 20 85 40 34 43
57 4 69 35 64 20 97 3
0 26 43 94 13 57 9 92
97 4 22 64 72 31 54 51
78 0 65 9 50 65 39 10
75 65 4 48 29 74 7 71
31 19 92 65 58 57 30 88
32 34 81 52 82 68 56 73
Anëtari më i madh në rreshtin 1 = 85
Anëtari më i madh në rreshtin 2 = 97
Anëtari më i madh në rreshtin 3 = 94
Anëtari më i madh në rreshtin 4 = 97
Anëtari më i madh në rreshtin 5 = 78
Anëtari më i madh në rreshtin 6 = 75
Anëtari më i madh në rreshtin 7 = 92
Anëtari më i madh në rreshtin 8 = 82
```



Pyetje ?