





## Azonosító: MAX03

Efejezetfeldolgozás után Ön képes lesz:

-  függvényeket definiálni
-  függvényeket deklarálni
-  függvényeket meghívni
-  különböző céllal függvényparamétereket megfelelően alkalmazni

## Feladat




Adjunk meg egy konstans vektorban egész számokat, keressük meg a vektorban a legnagyobb elemet, és az eredményt írjuk ki szabványos outputra.

Írjunk egy olyan „univerzális programot” - **eljárást, függvényt** - amely alkalmazásávalunk megadott, tetszőleges vektor maximuma meghatározására. A maximuma a függvényparaméterlistájában adjuk meg.

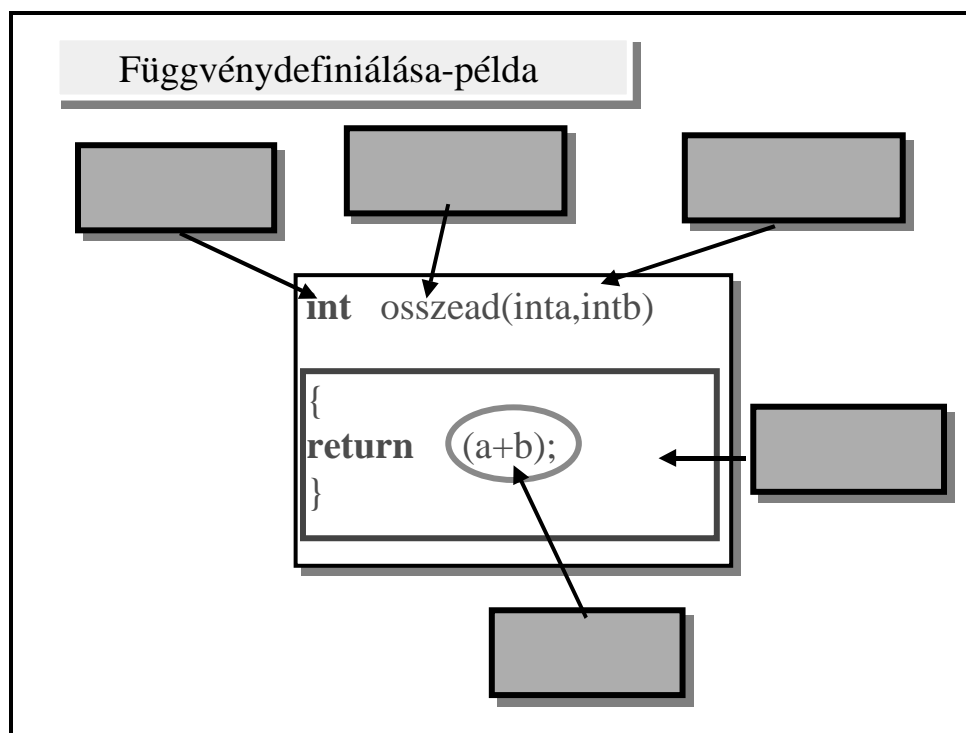
`maxker(v, n, i, max)`

## Megoldás

A feladat megoldása három fő részből áll:


-  adatok előkészítése
-  számítások végzése (absztrakt megoldó program) - **függvény**
-  eredmény megjelenítése

## A függvény



**Aeset**


Függvénydefiniálása	Függvénymeghívása
<pre>intosszead(inta,intb) {   return (a+b); }</pre>	<pre>... inta=1,b=2,c; c=osszead(a,b); ... </pre>

 : .....

.....

**Beset**


Függvénydefiniálása	Függvénymeghívása
<pre>voidosszead(inta,intb,intc) {   c=a+b; }</pre>	<pre>... inta=1,b=2,c=0 ; osszead(a,b,c); ... </pre>

 : .....

.....

**Ceset**


Függvénydefiniálása	Függvénymeghívása
<pre>voidosszead(inta,intb,int &amp;c) {   c=a+b; }</pre>	<pre>... inta=1,b=2,c=0; osszead(a,b,c); ... </pre>

 : .....

.....

**Deset**

Függvénydefiniálása	Függvénymeghívása
<pre>voidtorol(intv[],int n) {   for(int j=0;j!=n;j++)     v[j]=0; }</pre>	<pre>... intv [3]; torol(v,3); ... </pre>

 : .....

.....

.....

**Definíció →Hivatkozás**

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

```
void maxker(const int v[], const int n, int& i, int& max)
{
    . . . . .
}
```

```
int main(){
//Adatok előkészítése és megjelenítése
. . .
//Maximumkeresés
...
. . . maxker(v,n,i,max) . . .

//Eredmény megjelenítése

return 0;
}
```



.....

.....

.....

**Deklaráció→Hivatkozás →Definíció**

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

```
void maxker(const int, const int, int&, int& );
```

```
int main(){
//Adatok előkészítése és megjelenítése
. . .
//Maximumkeresés
...
. . . maxker(v,n,i,max) . . .
```

```
//Eredmény megjelenítése
. . .
return 0;
}
```

```
void maxker(const int v[], const int n, int& i, int& max)
{
    . . . . .
}
```



.....

.....

.....


## Megoldóprogram C++ -ban

```

/*****
/* Feladat: Maximumkeresés konstansként megadott vektorban          */
/* Maxker eljárással                                                */
/* Program neve: Max03.cpp                                          */
*****/

#include<iostream>
using namespace std;

```


: .....

.....

.....

```
void maxker(const int[], const int, int&, int& );
```



: .....

.....


.....

```

1int main()
{
    //Adatok előkészítése és megjelenítése
    char barmi;
    const int v[]={4,7,0,9,6,7,9,4};
    const int n=sizeof(v)/sizeof(v[0]);

    cout << "A vektor elemei: ";
    for (int j=0; j!=n; j++){
        cout << v[j];
        if (j != (n-1))
            cout << ", ";
        else
            cout << ".\n";
    }
}

```

: .....

.....


.....

```

//Maximumkeresés
int i,max;
maxker(v,n,i,max);

```



: .....

.....

.....

<sup>1</sup>Lásd: Max01

<sup>2</sup>//Eredmény megjelenítése

```
cout << "A vektor egyik legnagyobb eleme: " << v[i] << ".";
cout << "\n" << "Ez a vektor " << (i+1) << ". eleme. \n";
cin >> barmi;
return 0;
```

```
}// main
```



.....

.....

.....

```
void maxker(const int v[], const int n, int& i, int& max)
```



```
{
    int k=0;
    i=0;
    max=v[0];
    while(k!=(n-1)){
        if (v[k+1] >= max){
            i=k+1;
            max=v[k+1];
        }
        k=k+1;
    }
}
```

```
//maxker
```



.....

.....

.....

.....

.....

.....



Saját jegyzeteim



<sup>2</sup>Lásd: **Ma x01**

## Ateljes megoldóprogram

```

/*****
/* Feladat: Maximumkeresés konstansként megadott vektorban          */
/* Maxker eljárással                                              */
/* Program neve: Max03.cpp                                       */
*****/

#include<iostream>
using namespace std;

void maxker(const int[], const int, int&, int& );

3int main()
{
    //Adatok előkészítése és megjelenítése
    char barmi;
    const int v[]={4,7,0,9,6,7,9,4};
    const int n=sizeof(v)/sizeof(v[0]);

    cout << "A vektor elemei: ";
    for (int j=0; j!=n; j++){
        cout << v[j];
        if (j != (n-1))
            cout << ", ";
        else
            cout << ".\n";
    }

    //Maximumkeresés
    int i,max;
    maxker(v,n,i,max);

4//Eredmény megjelenítése
    cout << "A vektor egyik legnagyobb eleme: " << v[i] << ".";
    cout << "\n" << "Ez a vektor " << (i+1) << ". eleme. \n";
    cin >> barmi;
    return 0;

} // main

void maxker(const int v[], const int n, int& i, int& max)
{
    int k=0;
    i=0;
    max=v[0];
    while(k!=(n-1)){
        if (v[k+1] >= max){
            i=k+1;
            max=v[k+1];
        }
        k=k+1;
    }
} //maxker

```

<sup>3</sup>Lásd: **Max01**

<sup>4</sup>Lásd: **Max01**