



Azonosító:MAX05

EfejezetfeldolgozásutánÖnképeslesz:

-  Összetettadatszerkezeteket(struktúrákat)definiálniéshasználni
-  Függvénybenstruktúrátvisszatérésiértékkéntmegadni.

Feladat

Adjunkmeg egy konstansvektorban egész számokat, keressük meg a v vektor valamely maximális elemét, és az eredményt írjuk ki szabványos outputra.




Írjunk egy olyan „univerzális programot” - **eljárást, függvényt** - amely a kalmasegyáltalunk megadott, tetszőleges vektor maximális elemének meghatározására. A maximális elemet a függvény visszatérési értékeként adjuk meg.

maxelem = maxker(v, n),

ahol maxelem egy összetett adat, mely tartalmazza a maximális elem ind. exetési értékét.


Megoldás

A feladat megoldása három fő részből áll:

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

Astruktúra

Struktúra deklarálása	Struktúra definíció + hivatkozás
<pre>struct személy; { string neve; int kora; };</pre>	<pre>. . . main() { személy valaki; . . . valaki.kora valaki.neve . . . } // main</pre>

 :

.....

.....

.....

.....

.....

.....

.....

.....

.....

Struktúra, mint visszatérési érték

Struktúra deklarálása

```
struct maxelem;
{
    int index;
    int ertekek;
};
```



.....

.....

.....



.....

.....

.....

Függvénydeklarálása

```
maxelem maxker(const int[], const int);
```



.....

.....



.....

.....

Függvénymeghívása

```
. . .main . . . ;
{
    . . .
    maxelem elem;
    elem= maxker(v,n);
    . . .
};
```



.....

.....

.....

.....



.....

.....

Függvénydefiniálása

```
maxelem maxker(const int v[ ], const int n);
{
    maxelem m;
    . . . m.index . . .
    . . . m.ertekek . . .

    return m;
};
```



.....

.....

.....

.....

.....



.....

.....

.....

.....

Struktúra, mint paraméter

Struktúra deklarálása

```
struct szemely;
{
    string neve;
    int kora;
};
```



.....

.....

.....



.....

.....

Függvénydeklarálása

értékszerinti paraméterátadás esetén	címszerinti paraméterátadás esetén
<code>void fv(szemely);</code>	<code>void fv(szemely *);</code>



.....

.....

Függvény meghívása

értékszerinti paraméterátadás esetén	címszerinti paraméterátadás esetén
<pre>. . . main ... ; { szemely valaki; . . . fv(valaki) . . . };</pre>	<pre>. . . main . . . ; { szemely valaki; . . . fv(&valaki) . . . };</pre>



.....

.....

.....

Függvény definiálása

értékszerinti paraméterátadás esetén	címszerinti paraméterátadás esetén
<pre>void fv(szemely sz); { sz.kora=21; . . . };</pre>	<pre>void fv(szemely *p); { p->kora=21; . . . };</pre>



.....


.....

.....

.....

MegoldóprogramC++ -ban

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


:

.....

.....

```
struct maxelem
{
int index;
int ertek;
};
```



:

.....


.....

.....

.....

```
maxelem maxker(const 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";
}
```

:

.....

.....

¹Lásd: max01

```
//Maximumkeresés
```

```
maxelem elem;
```

```
elem=maxker(v,n);
```

☺

☺



.....

.....

.....

.....

```
//Eredmény megjelenítése
```

```
cout << "A vektor egyik legnagyobb eleme: " << elem.ertek << ".";
```

```
cout << "\n" << "Ez a vektor " << (elem.index+1) << ". eleme. \n";
```

```
cin >> barmi;
```

```
return 0;
```

```
}//main
```



.....

.....

.....

.....

.....

.....

.....

```
//Függvények definíciója
```

```
maxelem maxker(const int v[], const int n)
```

```
{
```

```
    int k=0;
```

```
    maxelem m;
```

```
    m.index=0;
```

```
    m.ertek=v[0];
```

```
    while(k!=(n-1)){
```

```
        if (v[k+1] >= m.ertek){
```

```
            m.index=k+1;
```

```
            m.ertek=v[k+1];
```

```
        }
```

```
        k=k+1;
```

```
    }
```

```
    return m;
```

```
}//maxker
```

☺

☺

☺

☺

☺

☺

☺



.....

.....

.....

.....

.....

Ateljesmegoldóprogram

```
2/*****  
/* Feladat: Maximumkeresés konstansként megadott vektorban */  
/* maxker függvénnnyel, mely struktúrában visszaadja      */  
/* a maxelem indexét és értékét.                          */  
/* Program neve: Max05.cpp                                */  
/*****  
  
#include<iostream>  
using namespace std;  
  
struct maxelem 😊  
{  
int index;  
int ertek;  
};  
  
maxelem maxker(const int[], const 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 😊  
    maxelem elem; 😊  
    elem=maxker(v,n); 😊  
  
    //Eredmény megjelenítése  
    cout << "A vektor egyik legnagyobb eleme: " << elem.ertek << ".";  
    cout << "\n" << "Ez a vektor " << (elem.index+1) << ". eleme. \n";  
    cin >> barmi;  
    return 0;  
} //main
```

²Lásd: max01

³Lásd: max01

//Függvények definíciója

maxelem maxker(const int v[], const int n)



{

int k=0;

maxelem m;



m.index=0;



m.ertek=v[0];



while(k!=(n-1)){

if (v[k+1] >= m.ertek){



m.index=k+1;



m.ertek=v[k+1];



}

k=k+1;

}

return m;



//maxker



Saját jegyzeteim

