

LEGMAGASABB HŐMÉRSÉKLET

Feladat:

Egymást követő napokon megmértük a napi hőmérsékletet, és ezeket az értékeket egy vektorban tároljuk. Melyik volt a legmagasabb hőmérséklet azon napok között, amikor fagyponthoz alatti hőmérsékletet regisztráltunk?

Specifikáció:

$$A = \text{vect}(\mathbf{R}) \times \mathbf{R} \times \mathbf{Z} \times \mathbf{L} \mid \times \mathbf{Z}$$

$$h \quad \max i \quad l \quad k$$

$$B = \text{vect}(\mathbf{R})$$

$$h'$$

$$Q = (h=h')$$

$$R = (Q \wedge l = (\exists j \in [h.lob..h.hib] : h_j < 0) \wedge l \Rightarrow (i \in [h.lob..h.hib] \wedge h_i < 0 \wedge \max = h_i \wedge \forall j \in [h.lob..h.hib] : h_j < 0 \Rightarrow h_j \leq h_i))$$

Visszavezetés: Feltételes maximumkeresés

Tétel	Feladat
m	$h.lob$
n	$h.hib$
$B(i)$	$h.i < 0$

Szigorítás	Megadjuk az értéket is.
Plussz feltétel	Nincs.
Plussz paraméter	h vektor.

Megoldás:

$k, l := h.lob - 1, \text{hamis}$			
$k \neq h.hib$			
$h_{k+1} \geq 0$	$(h_{k+1} < 0) \wedge \neg l$	$(h_{k+1} < 0) \wedge l$	
SKIP	$l, i, \max :=$ $igaz, k + 1, h_{k+1}$	$h_{k+1} \geq \max$	$h_{k+1} \leq \max$
		$i, \max := k + 1, h_{k+1}$	SKIP
$k := k + 1$			

Tétel:

A stuktogramm formában megadott program megoldása a feladatnak.

Bizonyítás:

A feltételes maximumkeresés tételből következik.

C++ Forráskód:

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
int main()
{
    //Alapváltozók
    string ragain;
```

```
string choice;
string fname;
const int maxn=365;
int n;
float* h;
float max=0;
int i=0;
bool l=false;
//Interface
do{
    cout << "ADATOK BEOLVASASA" << endl;
    do{
        cout << "Fajlból [F]" << endl;
        cout << "Billentyűről [B]" << endl;
        cin >> choice;}
    while((choice!="F")&&(choice!="f")&&(choice!="B")&&(choice!="b"));
    //Betöltés fajlból...
    if((choice=="F")||(choice=="f")){
        cout << "Fajlnev: ";
        cin >> fname;
        ifstream inp(fname.c_str());
        if(inp.fail()){
            cout << "A megadott fajlt nem talalom!" << endl;
            exit(1);}
        inp >> n;
        cout << "Napok száma: " << n << endl;
        if(n<1){
            cout << "Hibas adat!" << endl;
            exit(1);}
        h = new float[n];
        for(int l=0;l<n;++l){
            inp >> h[l];
            cout << (l+1) << ". napi homerseklet: " << h[l] << endl;}
    }else{
        //Beolvasas billentyurol...
        cout << "Napok száma (1-" << maxn << "): ";
        cin >> n;
        if(n<1){
            cout << "Legalabb 1 nap kell!" << endl;
            exit(1);}
        if(n>maxn){
            cout << "Legfeljebb " << maxn << " nap dolgozható fel!" << endl;
            exit(1);}
        h = new float[n];
        for(int j=0;j<n;++j){
            cout << (j+1) << ". napi homerseklet: ";
            cin >> h[j];}}
    //Felteteles maximumkereses
    for(int k=0;k<n;++k){
        if((h[k]<0)&&(l==false)){
            l=true;
            i=k;
            max=h[k];}
        if((h[k]<0)&&(l==true)){
            if(h[k]>=max){
                i=k;
                max=h[k];}}}
    //Eredmeny kiiratas
    if(l==true){
        cout << "Legmagasabb, fagypon alatti homerseklet napja: " << (i+1) << "." << endl;
        cout << "Legmagasabb, fagypon alatti homerseklet: " << max << " fok." << endl;
    }else{
        cout << "Nem volt fagypon alatti homerseklet." << endl;}
    delete[] h;
    cout << "Futtassam ujra (I/N)? ";
    cin >> ragain;
    while((ragain!="N")&&(ragain!="n"));
    return 0;
}
```

Tesztelési terv:

Tesztesek a feladat alapján (fekete doboz tesztelés):

- *Nem létező fájl kipróbálása;*
- *Létező fájl kipróbálása*

- *Billentyűről olvasás kipróbálása;*
- *Fagyypont feletti ($h[x]>0$) értékek kipróbálása;*
- *Fagyypont ($h[x]=0$) érték kipróbálása;*
- *Fagyypont alatti ($h[x]<0$) értékek kipróbálása;*
- *Újrafuttatás kipróbálása;*

Tesztesek a kód alapján (fehér doboz tesztelés):

- *Nem létező fájl kipróbálása;*
- *Létező fájl kipróbálása;*
- *Billentyűről olvasás kipróbálása;*
- *Nulla ($n=0$) érték megadása tömbdeklaráció előtt;*
- *Negatív ($n<0$) értékek megadása tömbdeklaráció előtt;*
- *Pozitív ($n>0$) értékek megadása tömbdeklaráció előtt;*
- *Fagyypont feletti ($h[j]>0$) értékek kipróbálása;*
- *Fagyypont ($h[j]=0$) érték kipróbálása;*
- *Fagyypont ($h[j]<0$) értékek kipróbálása;*
- *Újrafuttatás kipróbálása.*