Dokumentáció a 26. feladathoz

**Felhasználói dokumentáció**

**Feladat:**

A meteorológiai intézet az ország N településére adott M napos időjárás előrejelzést, az adott településen az adott napra várt legmagasabb hőmérsékletet. Készíts programot, amely megadja azokat a napokat, amikor mindenhol legalább F fok várható!

**Környezet:**

IBM PC, legalább MS DOS 5.0 operációs rendszer.

**Használat:**

1. A program indítása

A program meteorologia(3).pas és meterologia(3).exe néven található az elküldött mellékletben.

A program indítása a meterologia(3).exe kell elindítani. A meterologias(3).pas a Geany nevű fejlesztői környezetben nyitható meg és ebben fordítás után jön létre a futtatható állomány.

1. A program bemenete

* Ha a programot paraméter nélkül indítjuk, akkor először bekéri, hogy hány település, nap és hőmérsékleti adattal dolgozzon, majd bekéri az adatokat. Egy lehetséges párbeszéd (*vastagon és dőlten* szedtük a felhasználó válaszait):

|  |
| --- |
| Hány település van?  ***3***  Hány naposak az előre jelzések?  ***4***  Mekkora hőmérsékletet keresünk?  ***10***  … |

* Ha a programot paraméterrel indítjuk, akkor a paraméter fájlt feldolgozza, és annak megfelelően számolja ki a keresett településeket. A fájl felépítése a következő:

Első sora: települések száma

Második sora: napok száma

Harmadik sora: keresett hőmérséklet

Többi sor: településen mért hőmérsékleti adatok

|  |
| --- |
| Példa (4 település, 8 nap, legalább 2 fok):  4  8  2  1 2 3 4 5 6 7 8  3 2 4 5 3 4 5 10  3 4 5 6 7 3 11 4 |

**Hiba lehetőség:**

Ha a városok vagy napok számához kisebb, mint 1-et adunk meg, akkor a program kiírja, hogy „Helytelen parameter(ek)! (N>1, M>1)”. Ilyenkor a programot újra kell indítani.

**A program eredménye:**

A program visszaírja az általunk beírt adatokat, majd kiírja azoknak a városoknak a számait, ahol legalább annyi hőmérséklet volt, amennyit megadtunk bemeneti adatként.

**Fejlesztői dokumentáció**

**Feladat:**

A meteorológiai intézet az ország N településére adott M napos időjárás előrejelzést, az adott településen az adott napra várt legmagasabb hőmérsékletet. Készíts programot, amely megadja azokat a napokat, amikor mindenhol legalább F fok várható!

**Specifikáció:**

**Bemenet:**

[Települések száma]

[Napok száma]

[Beolvasott homersekleti adatok]

[Keresendő hőmérséklet]

**Kimenet:**

[Napok száma, ahol legalább F fok volt mindenhol]

**Előfeltétel:**

**Utófeltétel:**

**Környezet:**

IBM PC, legalább MS DOS 5.0 operációs rendszer. Free Pascal (Geany) fejlesztői környezet, crt unit.

**Forráskód:**

A program a meterologia(3).pas néven tálalható.

**Megoldás:**

**Fontos típusok:**

A specifikációban leirt valtozok típusai:

type MatrixType = array of array of integer;

type ArrayType = array of integer;

**Algoritmus:**

|  |
| --- |
| **Program** meterologia;  AdatokBeolvasasa(Matrix, N, M, F);  Kereses(Matrix, N, M, F, Eredmeny, EredmenySzama);  MatrixotKiir(Matrix, N, M);  EredmenytKiir(Eredmeny,EredmenySzama, F);  **Program vege**. |

|  |
| --- |
| **Eljaras** AdatokBeolvasasa (var Matrix : MatrixTipus; var N : egeszek; var M : egeszek; var F : egeszek):  Beolvas(N);  Beolvas(M);  Beolvas(F);  **ciklus** i:=0**-tol** N-1**-ig**  **ciklus** j:=0**-tol** M-1**-ig**  Beolvas(Matrix[i,j]);  **Eljaras vege.**  **Eljaras** Kereses(Matrix : MatrixType; N : egeszek; M: egeszek; F : egeszek; var Eredmeny : Tomb; var EredmenySzama : egeszek):  **Valtozok**: i,j egeszek  talalt : logikai  **ciklus** j:=0**-tol** M-1**-ig**  {  talalt := true;  **ciklus** i:=0**-tol** N-1**-ig**  {  Ha Matrix(i,j) < F  talalt := false  }  **Ha** talalt:  Eredmeny(EredmenySzama) := j + 1  EredmenySzama := EredmenySzama + 1  }  **Eljaras vege.**  **Eljaras** MatrixotKiir(Matrix : MatrixType; N : egeszek; M : egeszek):  **Valtozok**: i,j : egeszek;  **ciklus** i:=0**-tol** N-1**-ig**  **ciklus** j:=0**-tol** M-1**-ig**  Kiir(Matrix(i,j));  **Eljaras vege.**  **Eljaras** EredmenytKiir(Eredmeny : ArrayType; EredmenySzama : egeszek; F : egeszek);  **Valtozok**: i : egeszek;  **Ha** EredmenySzama = 0  Kiir(Nincs ilyen nap);  **Ha** EredmenySzama 0  Kiir(Eredmeny:);  **cikuls** i:=0**-tol** EredmenySzama-1**-ig**  Kiir(Eredmeny(i));  **Eljaras vege.** |

**Pascal kod:**

|  |
| --- |
| **program** Meterlogia;  **uses** crt;  **type**  MatrixType = **array** of **array** of **integer**;  ArrayType = **array** of **integer**;  **function** HelyesekAParameterek(N : **integer**; M : **integer** ) : **boolean**;  **begin**  HelyesekAParameterek := **true**;  **if** N < 1 **then**  **begin**  HelyesekAParameterek := **false**;  **end**;  **if** M < 1 **then**  **begin**  HelyesekAParameterek := **false**;  **end**;  **end**;  **procedure** AdatokBeolvasasaBillentyuzetrol(**var** Matrix : MatrixType; **var** N : **integer**; **var** M : **integer**; **var** F : **integer**);  **var** i,j : **integer**;  **begin**  **Write**('Hany telepules van?: ');  **ReadLn**(N);  **Write**('Hany naposak az elorejelzesek?: ');  **ReadLn**(M);  **Write**('Mekkora homersekletet keresunk?: ');  **ReadLn**(F);  **if** HelyesekAParameterek(N,M) = **true** then  **begin**  **SetLength**(Matrix, N, M);  **for** i:=0 **to** N-1 **do**  **begin**  **for** j:=0 **to** M-1 **do**  **begin**  **Write**(i+1,'-dik varos ',j+1,' napi homerseklete: ');  **ReadLn**(Matrix[i,j]);  **end**;  **end**;  **end**  **end**;  **procedure** AdatokBeolvasasaFajlbol(**var** Matrix : MatrixType; **var** N : **integer**; var M : **integer**; **var** F : **integer**; FajlNev : **string**);  **var** Fajl : **text**;  **var** i,j : **integer**;  **begin**  **Assign**(Fajl, FajlNev);  **{$I-}**  **Reset**(Fajl);  **{$I+}**  **if** ioresult <> 0 **then**  **begin**  **WriteLn**(FajlNev, ' nevu fajlt nem sikerult megnyitni!');  **halt**;  **end**;  **ReadLn**(Fajl,N);  **ReadLn**(Fajl,M);  **ReadLn**(Fajl,F);  **if** HelyesekAParameterek(N,M) = **true** **then**  **begin**  **SetLength**(Matrix, N, M);  i := 0;  **while** **not** eof(Fajl) **do**  **begin**  j := 0;  **while** **not** eoln(Fajl) **do**  **begin**  **Read**(Fajl, Matrix[i,j]);  j := j + 1;  **end**;  **ReadLn**(Fajl);  i := i + 1;  **end**;  **end**;  **Close**(Fajl);  **end**;  **procedure** Kereses(Matrix : MatrixType; N : **integer**; M: **integer**; F : **integer**; **var** Eredmeny : ArrayType; **var** EredmenySzama : **integer**);  **var** i,j : **integer**;  **var** talalt : **boolean**;  **begin**  **for** j:=0 **to** M-1 **do**  **begin**  talalt := **true**;  **for** i:=0 **to** N-1 **do**  **begin**  **if** Matrix[i,j] < F **then**  **begin**  talalt := **false**;  **end**;  **end**;  **if** talalt = **true** **then**  **begin**  Eredmeny[EredmenySzama] := j + 1;  EredmenySzama := EredmenySzama + 1;  **end**;  **end**;  **end**;  **procedure** MatrixotKiir(Matrix : MatrixType; N : **integer**; M : **integer** );  **var** i,j : **integer**;  **begin**  **for** i:=0 **to** N-1 **do**  **begin**  **for** j:=0 **to** M-1 **do**  **begin**  **Write**(Matrix[i,j], ' ');  **end**;  **WriteLn**();  **end**;  **end**;  **procedure** EredmenytKiir(Eredmeny : ArrayType; EredmenySzama : **integer**; F : **integer**);  **var** i : **integer**;  **begin**  **if** EredmenySzama = 0 **then**  **begin**  **Write**('Egyik napon sem volt legalabb ', F, ' fok minden varosban!');  **end**  **else**  **begin**  **Write**('A kovetkezo nap(okon) volt legalabb ',F, ' fok minden varosban: ');  **for** i:=0 **to** EredmenySzama-1 **do**  **begin**  **Write**(Eredmeny[i],' ');  **end**;  **end**;  **end**;  **var** Matrix : MatrixType;  **var** Eredmeny : ArrayType;  **var** N : **integer**;  **var** M : **integer**;  **var** F : **integer**;  **var** FajlNev : **string**;  **begin**  ClrScr();  **if** ParamCount = 1 **then**  **begin**  FajlNev := ParamStr(1);  **WriteLn**('Beolvasas a(z) ', FajlNev, ' fajlbol...');  AdatokBeolvasasaFajlbol(Matrix, N, M, F, FajlNev);  **end**  **else**  **begin**  AdatokBeolvasasaBillentyuzetrol(Matrix, N, M, F);  **end**;  **if** HelyesekAParameterek(N,M) = **true** **then**  **begin**  SetLength(Eredmeny,M);  Kereses(Matrix, N, M, F, Eredmeny, EredmenySzama);  MatrixotKiir(Matrix, N, M);  EredmenytKiir(Eredmeny,EredmenySzama, F);  **end**  **else**  **begin**  **WriteLn**('Helytelen parameter(ek)! (N>1, M>1)');  **end**;  **Readln**;  **end**. |

**Teszteles:**

N = Telepulesek szama

M= Napok szama

F= Keresendő hőmérséklet

1. Standard bemenet

* N=-1, M=1, F=1

Helytelen parameter(ek)! (N>1, M>1)

* N=1, M=-1, F=1

Helytelen parameter(ek)! (N>1, M>1)

* N=0, M=1, F=1

Helytelen parameter(ek)! (N>1, M>1)

* N=1, M=0, F=1

Helytelen parameter(ek)! (N>1, M>1)

* N=1, M=1, F=1

Hany telepules van?: 1

Hany naposak az elorejelzesek?: 1

Mekkora homersekletet keresunk?: 1

1-dik varos 1 napi homerseklete: 1

A kovetkezo nap(okon) volt legalabb 1 fok minden varosban: 1

* N=3, M=4, F=2

Hany telepules van?: 3

Hany naposak az elorejelzesek?: 4

Mekkora homersekletet keresunk?: 2

1-dik varos 1 napi homerseklete: 1

1-dik varos 2 napi homerseklete: 3

1-dik varos 3 napi homerseklete: 4

1-dik varos 4 napi homerseklete: 5

2-dik varos 1 napi homerseklete: 3

2-dik varos 2 napi homerseklete: 2

2-dik varos 3 napi homerseklete: 3

2-dik varos 4 napi homerseklete: 5

3-dik varos 1 napi homerseklete: 6

3-dik varos 2 napi homerseklete: 3

3-dik varos 3 napi homerseklete: 3

3-dik varos 4 napi homerseklete: 2

1 3 4 5

3 2 3 5

6 3 3 2

A kovetkezo nap(okon) volt legalabb 2 fok minden varosban: 2 3 4

1. Paraméter fájl

* Teszt 1

Üres fájl!

Helytelen parameter(ek)! (N>1, M>1)

* Teszt 2

Hiányzó fájl.

hianyzo.txt nevu fajlt nem sikerult megnyitni!

* Teszt 3

-1

4

5

Helytelen parameter(ek)! (N>1, M>1)

* Teszt 4

1

-1

20

Helytelen parameter(ek)! (N>1, M>1)

* Teszt 5

0

4

5

Helytelen parameter(ek)! (N>1, M>1)

* Teszt 6

1

0

20

Helytelen parameter(ek)! (N>1, M>1)

* Teszt 7

4

8

5

1 2 5 6 -5 6 7 8

3 2 5 5 5 4 5 10

3 4 5 6 5 3 11 4

5 2 5 -1 5 6 7 1

A kovetkezo nap(okon) volt legalabb 5 fok minden varosban: 3 7

* Teszt 8

6

20

10

15 23 35 26 -5 26 27 23 31 33 34 12 35 16 17 32 31 21 4 31

25 12 25 11 15 14 15 10 11 24 12 1 4 4 12 17 24 2 1 -9

30 4 5 6 5 3 11 4 34 12 -9 0 1 23 32 1 32 4 12 -9

18 2 5 -1 5 6 7 1 23 15 19 2 -8 2 31 25 17 19 31 22

20 18 14 2 30 23 15 15 17 17 8 -9 13 5 23 12 22 21 31 31

23 23 4 25 25 26 30 10 22 17 13 19 24 26 -2 -2 1 3 41 19

A kovetkezo nap(okon) volt legalabb 10 fok minden varosban: 1 9 10

**Fejlesztési lehetőségek:**

A mátrix elemei a természetes számok helyett lehetnének valós számok.

**Készítő:**

Takács Róbert