



Belépő a tudás közösségébe

Informatika szakköri segédanyag



Elemi programozási tételek 1.

**Heizlerné Bakonyi Viktória, Horváth Győző, Menyhárt László,
Szlávi Péter, Törley Gábor, Zsakó László**

Szerkesztő: Zsakó László, Abonyi-Tóth Andor

A kiadvány „A felsőoktatásba bekerülést elősegítő készségfejlesztő és kommunikációs programok megvalósítása, valamint az MTMI szakok népszerűsítése a felsőoktatásban” (EFOP-3.4.4-16-2017-006) című pályázat keretében készült 2017-ben.



Eötvös Loránd Tudományegyetem
Informatikai Kar

SZÉCHENYI 2020



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

Feladataink egy jelentős csoportjában egyetlen bemenő sorozat alapján kell meghatároznunk egy értéket eredményként.

1. Sorozatszámítás

Kezdjük a legelső témakört néhány feladattal!

- F1. Egy osztály N db tanuló osztályzatának ismeretében adjuk meg az osztály átlagát!
- F2. Egy M elemű betűsorozat betűit fűzzük össze egyetlen szöveg típusú változóba!
- F3. Készítsünk algoritmust, amely egy autóversenyző körönkénti ideje alapján meghatározza a versenyző egy kör megtételéhez szükséges átlagidejét!
- F4. A Balaton mentén K db madarász végzett megfigyeléseket. Mindegyik megadta, hogy milyen madarakat látott. Készítsünk algoritmust, amely megadja a megfigyelések alapján a Balatonon előforduló madárfajokat!
- F5. Adjuk meg az első N természetes szám szorzatát (N faktoriális)!

Vizsgáljuk meg, mi a közös a fenti öt feladatban! Mindegyikben adatok valamilyen sorozatával kell foglalkoznunk, e sorozathoz kell hozzárendelnünk egyetlen értéket. Ezt az értéket egy, az egész sorozaton értelmezett függvény adja (N szám összege, M betű egymásutánírása, K halmaz uniója, N szám szorzata). Ezt a függvényt azonban felbonthatjuk értékpárokon kiszámított függvények sorozatára (2 „valami” összegére, egymásután írására, uniójára, szorzatára).

Az algoritmus:

Változók:

N :	Egész	[a feldolgozandó sorozat elemei száma]
X :	Tömb ($1..N$:Elemtípus)	[a feldolgozandó sorozat elemei]
F_0 :	Elemtípus	[a művelet nullelem]
S :	Elemtípus	[az eredmény]

Így tehát minden olyan művelet szerepelhet e feladattípusban, amelyre a matematika valamilyen „egységes” jelölést használ: összeg, szorzat, unió, metszet, logikai műveletek, konkatenáció, Mindegyik művelet visszavezethető egy bináris műveletre, s megadható mindegyikhez egy semleges elem (nullelem) is, amelyre egy tetszőleges elemmel és vele elvégzett 2 operandusú művelet az adott elemet adja.

Variációk: $F = \Sigma, \Pi, \cup, \cap, \wedge, \vee, \&$ (konkatenáció).

$F_0 = 0, 1, \{\}, \text{alaphalmaz, Igaz, Hamis, ""}$.

A megoldás a nullelemre, valamint a 2 operandusú műveletre épül, a matematikában is szokásos módszer, az indukció alapján:

- 0 elemre tudjuk az eredményt: F_0 ,
- ha $i-1$ elemre tudjuk az eredményt (F^{i-1}), akkor i elemre (F^i) a következőképpen kell kiszámítani:
 $F^i = f(F^{i-1}, X_i)$.

Az F -re vonatkozó alábbi rekurzív definícióból:

$$F^i := F(X_1, \dots, X_i) := \begin{cases} F_0 & , \text{ ha } i = 0 \\ f(F^{i-1}, X_i) & , \text{ különben} \end{cases}$$

Végezetül az algoritmikus nyelvünkben a feladatot megoldó algoritmus:

```
Sorozatszámítás (N, X, S) :
    S := F0
    Ciklus i=1-től N-ig
        S := f(S, X[i])
    Ciklus vége
Eljárás vége.
```

Alkalmazzuk az általános megoldást a fejezet elején kitűzött egyes feladatokra!

F1. N szám összege

```
elemek száma: N
osztályzatok: X (N db elem)
F, f, F0      : S, +, 0
```

```
Összegzés (N, X, S) :
    S := 0
    Ciklus i=1-től N-ig
        S := S + X[i]
    Ciklus vége
Eljárás vége.
```

Minta kódok

C++ [cpp/1_Nszamosszeg/feladat.cpp](#)

C# [cs/1_Nszamosszeg/feladat.cs](#)

Java [java/1_Nszamosszeg/feladat.java](#)

Pascal [pas/1_Nszamosszeg/feladat.pas](#)

Python [py/1_Nszamosszeg/feladat.py](#)



2. M betű egymásutánírása

```

elemek száma: M
betűk       : X (M db
elem)
F, f, F0   : &, +, "",

```

```

Egymásután (M, X, Sz) :
  Sz:="" (üres szöveg)
  Ciklus i=1-től M-ig
    Sz:=Sz+X[i]
  Ciklus vége
Eljárás vége.

```

Minta kódok

C++ cpp/2_Mbetuegymasutan/feladat.cpp

C# cs/2_Mbetuegymasutan/feladat.cs

Java java/2_Mbetuegymasutan/feladat.java

Pascal pas/2_Mbetuegymasutan/feladat.pas

Python py/2_Mbetuegymasutan/feladat.py



F4. K halmaz uniója

```

elemek száma : K
megfigyelések: X (N db elem)
F, f, F0   : ∪, ∪, {}

```

```

Unió (K, X, H) :
  H:={} (üres halmaz)
  Ciklus i=1-től K-ig
    H:=H∪X[i]
  Ciklus vége
Eljárás vége.

```

2. Megszámolás

Az előző feladatokban előfordulhatott, hogy több elem is rendelkezik a vizsgált tulajdonsággal. Ekkor érdekes lehet annak megvizsgálása, hogy hány ilyen elem van.

F6. Családok létszámának, illetve jövedelmének alapján állapítsuk meg, hogy hány család él a létminimum alatt!

F7. Egy futóverseny végeredménye határozzuk meg, hogy a versenyzők hány százaléka teljesítette az olimpiai induláshoz szükséges szintet!

F8. Adjuk meg egy szöveg magánhangzóinak számát!

A közös jellemző itt tehát a számlálás. Vegyük észre, hogy a feladatot sorozatszámításként is fel-foghatjuk, amelyben 1-eseket kell összeadnunk, de bizonyos feltételtől függően.

Az algoritmus:**Függvény:**

$T: \text{Elemtípus} \rightarrow \text{Logikai}$

Változók:

N : Egész [a feldolgozandó sorozat elemei száma]
X : Tömb[1..N:Elemtípus] [a feldolgozandó sorozat elemei]
Db: Egész [eredmény - a megfelelő elemek száma]

A feladat sorozatszámítás (sőt összegzés), tehát egy ciklust kell alkalmazni a megoldáshoz. A ciklus belsejében egy unió típusú (χ függvény értéke) adatot kell feldolgozni, ezt egy elágazással tehetjük meg.

Megszámolás (N, X, Db) :

Db:=0

Ciklus i=1-től N-ig

Ha T(X[i]) **akkor** Db:=Db+1

Ciklus vége

Eljárás vége.

Nézzük meg két feladat megoldását!

F16. Létnminimum alattiak száma.

Minden családi létszámhoz megadható az a jövedelem, amely a létminimumhoz kell, a megoldásban ezt használjuk fel.

elemek száma: N
létszámok: L (N db elem)
jövedelmek: J (N db elem)
létminimumok: MIN sorozat
 $T(L-j) : j - \text{Min}(L) \geq 0$

Megszámolás (N, J, L, MIN, Db) :

Db:=0

Ciklus i=1-től N-ig

Ha J[i] ≤ MIN[L[i]]

akkor Db:=Db+1

Ciklus vége

Eljárás vége.

Minta kódok

C++ [cpp/3_Letminimum/feladat.cpp](#)

C# [cs/3_Letminimum/feladat.cs](#)

Java [java/3_Letminimum/feladat.java](#)

Pascal [pas/3_Letminimum/feladat.pas](#)

Python [py/3_Letminimum/feladat.py](#)



F6. Olimpiai indulási szintet hány százalék teljesítette?

A feladat megoldása egy megszámlálás, majd az eredményből és a résztvevők számából százalékot számolunk.

```
elemek száma: N
idők: Id (N db elem)
T(id) : id ≤ Szint
```

```
Megszámlálás (N, Id, Száz) :
  Db := 0
  Ciklus i = 1-től N-ig
    Ha Id[i] ≤ Szint akkor Db := Db + 1
  Ciklus vége
  Száz := Kerekít(100 * Db / N)
Eljárás vége.
```

Minta kódok

C++ [cpp/4 Olimpia/feladat.cpp](#)

C# [cs/4 Olimpia/feladat.cs](#)

Java [java/4 Olimpia/feladat.java](#)

Pascal [pas/4 Olimpia/feladat.pas](#)

Python [py/4 Olimpia/feladat.py](#)



3. Maximumkiválasztás

A sorozatszámítás egy újabb speciális esetével ismerkedünk meg a következő feladattípusban, először néhány feladaton keresztül.

F9. Egy kórházban megmérték minden beteg lázát, adjuk meg, hogy ki a leglázasabb!

F10. Egy család havi bevételei és kiadásai alapján adjuk meg, hogy melyik hónapban tudtak a legtöbbet megtakarítani!

F11. Egy osztály tanulói nevei alapján adjuk meg a névsorban legelső tanulót!

Közös jellemzőjük e feladatoknak, hogy minden esetben egy sorozat elemei közül kell kiválasztani a legnagyobbat (illetve a legkisebbet). Itt is meggondolandó – mint a *kiválasztás*nál –, hogy elem értéket vagy pedig elem sorszámot várunk eredményként. Az ottani gondolatmenethez hasonlóan most is a sorszám meghatározását választjuk.

Az algoritmus:**Változók:**

N : Egész [a feldolgozandó sorozat elemei száma]
 X : Tömb(1..N:Elem típus) [a feldolgozandó sorozat elemei]
 Max: Egész [a maximális értékű elem sorszáma]

A sorozatszámításnál használjuk **F** függvényként a $\text{MAX}(A_1, A_2, \dots, A_N)$ függvényt! Ehhez az **f** függvényt a következőképpen használjuk:

$$f(x, y) := \max(x, y).$$

Legyen ehhez $F_0 := A_1$! Alakítsuk át úgy a sorozatszámítás megoldását, hogy ne értéket, hanem sorszámot kapjunk eredményül! (Annyi észrevenni valónk van csak az algoritmizáláskor, hogy most a sorozat a 2. elemmel kezdődik.)

Maximumkiválasztás (N, X, Max) :

```
Max:=1
Ciklus i=2-től N-ig
  Ha X[Max]<X[i] akkor Max:=i
Ciklus vége
Eljárás vége.
```

Sok esetben túlságosan sok időbe kerül a sorozat egy elemének meghatározása. Ekkor olyan megoldást készíthetünk, amely a maximális értéket határozza meg, vagy pedig a sorszámot is és az értéket is.

Maximumkiválasztás (N, X, Max, Maxért) :

```
Max:=1; Maxért:=X[1]
Ciklus i=2-től N-ig
  Ha Maxért<X[i] akkor Max:=i; Maxért:=X[i]
Ciklus vége
Eljárás vége.
```

Ha a feladat minimumkiválasztás, akkor pedig nincs más teendőnk, mint az elágazás feltételében szereplő < reláció átírása >-ra. Erre példa az egyik kitűzött feladat megoldása.

F11. A névsorban legelső tanuló.

```
elemek száma: N
tanulók neve: X (N db elem)
```

Minimum (N, X, Min, Név) :

```
Min:=1; Név:=X[1]
Ciklus i=2-től N-ig
  Ha Név>X[i]
    akkor Min:=i
    Név:=X[i]
Ciklus vége
Eljárás vége.
```

Minta kódok

C++	cpp/5_Nevsorban/feladat.cpp
C#	cs/5_Nevsorban/feladat.cs
Java	java/5_Nevsorban/feladat.java
Pascal	pas/5_Nevsorban/feladat.pas
Python	py/5_Nevsorban/feladat.py



Feladatok programozási tételekre a Nemes Tihamér OITV-ről és az Informatika OKTV-ről

1. feladat

Egy iskola tanáraitól tudjuk, hogy mikor milyen órát tartanak. A tanárokat, a tantárgyakat, a hét napjait, a napokon belüli órákat sorszámukkal azonosítjuk. Készíts programot, amely megadja:

- A. minden napra az aznap órát tartó tanárok számát;
- B. azt a tantárgyat, amit a legtöbb tanár tanít;
- C. azt a tanárt, akinek a legtöbb lyukasórája van (lyukasóra: aznap előtte is van órája valamikor és utána is van órája valamikor);
- D. az adott T tanárt egész héten helyettesíteni tudó tanárt.

A standard bemenet első sorában a tanárok száma ($1 \leq N \leq 100$), a tantárgyak száma ($1 \leq M \leq 100$) és egy tanár sorszáma van ($1 \leq T \leq N$), egy-egy szóközzel elválasztva. A következő sorok mindegyikében 4 egész szám van, egy-egy szóközzel elválasztva: tanár sorszáma, tanított tantárgy sorszáma, nap (1 és 5 közötti egész szám), óra (0 és 8 közötti egész szám). Például 3 7 2 0 azt jelenti, hogy a harmadik tanár a hetedik tantárgyat a hét második napján a nulladik órában tanítja.

A standard kimenetre négy sort kell írni! Az első sorba az A, a másodikba a B, a harmadikba a C, a negyedikbe pedig a D részfeladat eredményét. Ha több megoldás van, akkor az elsőket kell kiírni. Ha nincs megoldás (C és D részfeladatban), akkor -1-et kell kiírni! Az első sorban 5 szám szerepeljen, egy-egy szóközzel elválasztva!

Példa:

bemenet	kimenet
3 4 1	2 3 1 0 0
1 1 1 6	2
1 1 2 2	1
1 2 1 3	3
2 1 2 2	
2 2 3 1	
3 4 1 2	
3 2 1 4	
3 3 2 1	

A példában szereplő 3 tanár órarendje:

1. tanár	1. nap	2. nap	3. nap	2. tanár	1. nap	2. nap	3. nap	3. tanár	1. nap	2. nap	3. nap
0. óra				0. óra			T2	0. óra			
1. óra				1. óra				1. óra		T3	
2. óra		T1		2. óra		T1		2. óra	T4		
3. óra	T2			3. óra				3. óra			
4. óra				4. óra				4. óra	T2		
5. óra				5. óra				5. óra			
6. óra	T1			6. óra				6. óra			

A. MEGSZÁMOLÁSI TÉTEL!

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Megszámolási tétel
Feladat	29. Órát tartó tanárok száma naponként

B. NEM ELEMÍ TÉTELES FELADAT!

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Tételek összeépítése
Feladat	53. Legtöbb tanár által tanított tantárgy

C. NEM ELEMÍ TÉTELES FELADAT!

Minden tanárra minden napra számoljuk meg, hogy az adott tanár aznap hány órát tart (DB), válasszuk ki az aznapi legelső (E) és legutolsó (U) óráját. A lyukasórak száma ekkor $U-E+1-DB$, ha $DB > 0$.

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Tételek összeépítése
Feladat	44. Legkevesebb lyukasórájú tanár

Beolvasás:

```
DB[] := [0, ..., 0] : E[] := [9, ..., 9] : U[] := [-1, ..., -1]
```

```
Ciklus amíg nem vége(f)
```

```
  Olvas(f, i, j, k, o); DB[i, k] := DB[i, k] + 1
```

```
  Ha o < E[i, k] akkor E[i, k] := o
```

```
  Ha o > U[i, k] akkor U[i, k] := o
```

```
Ciklus vége
```

```
Eljárás vége.
```

C:

```
Max := 0; Maxért := -1
```

```
Ciklus i = 1-től n-ig
```

```
  Lyukas[i] := 0
```

```
  Ciklus j = 1-től 5-ig
```

```
    Ha Db[i, j] > 1
```

```
      akkor Lyukas[i] := Lyukas[i] + U[i, j] - E[i, j] + 1 - Db[i, j]
```

```
  Ciklus vége
```

```
  Ha Lyukas[i] > Maxért akkor Maxért := Lyukas[i]; Max := i
```

```
Ciklus vége
```

```
Eljárás vége.
```

D. NEM ELEMİ TÉTELES FELADAT!

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Tételek összeépítése
Feladat	4. Egész héten helyettesíteni tudó tanár

2. feladat

Egy vizsgabizottságban egy nap feljegyezték, hogy az egyes vizsgázók mikor vizsgáztak (egyszerre egy vizsgázó lehet), mindenkiről 4 adatot tudunk: kezdőóra, kezdőperc, végóra, végperc. A vizsgázók adatait idő szerinti sorrendben kapjuk.

Készíts programot, amely beolvassa a vizsgázók számát ($1 \leq N \leq 100$), az egyes vizsgázók vizsgájának kezdetét ($V[i,1]$) és végét ($V[i,2]$), majd megadja

- A. annak a 60 perces időszaknak a kezdetét, amikor a legtöbb vizsgázó végzett a vizsgájával (közülük az első pontosan ebben a percben végezzen);
- B. a leghosszabb vizsgaszünet hosszát – óra, perc (amikor 2 vizsgázó között senki sem volt a vizsgabizottságnál);
- C. a leghosszabb időtartamot, amikor a vizsgabizottság nem tarthatott szünetet!

Példa

Bemenet:

N=6
 8 20 8 30
 8 50 9 0
 9 50 10 10
 10 10 10 30
 10 30 10 55
 11 55 12 10

Kimenet:

10 10 (10 óra 10 perctől)
 1 0 (1 óra, 0 perc)
 1 5 (1 óra, 5 perc)

Az első fontos gondolat: beolvasáskor azonnal mindent számoljunk át percekbe, azzal könnyebb lesz dolgozni! Eredmény kiírásnál pedig számoljuk vissza órákra és percekre!

A. NEM ELEMI TÉTEL;

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	
Téma	
Feladat	

B. vizsgakezdetek és az előző vizsga végek különbségének maximuma;

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	
Téma	
Feladat	

C. az utolsó szünet végétől az adott vizsga végéig terjedő időtartamok maximuma!

B:

Idő:=0

Ciklus i=1-től N-1-ig**Ha** V[i+1,1]-V[i,2]>idő **akkor** Idő:=V[i+1,1]-V[i,2]**Ciklus vége****Eljárás vége.**

C:

Idő:=-1; K:=1

Ciklus i:=2-től n-ig**Ha** V[i,1]≠V[i-1,2] **akkor****Ha** V[i-1,2]-V[k,1]>idő **akkor** Idő:=V[i-1,2]-V[k,1]

K:=i

Elágazás vége**Ciklus vége****Ha** V[n,2]-V[k,1]>j **akkor** Idő:=V[n,2]-V[k,1]**Eljárás vége.**

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	
Téma	
Feladat	

3. feladat

Ismerjük N emberről, hogy ki szülője kinek ($1 \leq N \leq 100$). Az embereket az 1 és N közötti sorszámmal azonosítjuk.

Készíts programot, amely megadja, hogy

- A. kinek van a legtöbb gyereke;
- B. kinek van a legtöbb unokája;
- C. hány embernek nincs unokája!

A bemenet első sorában az N értéke szerepel, a következő N-1 sor pedig egy-egy szülő és gyerek sorszámtartalmazza.

Példa:

Bemenet:

Kimenet: Magyarázat:

N=12

szülő:	gyerek:	A: 4	5,6,7,10 sorszámu
1	2	B: 1	4,8,9,11,12 sorszámu
1	3	C: 10	csak 1-nek és 2-nek van
2	4		
2	8		
2	9		
4	5		
4	6		
4	7		
4	10		
3	11		
3	12		

- A. aki szülőként szerepel, annak a gyerekszámát növeljük eggyel – ha mindenkinek ismert a gyerekszáma, akkor azokra kell egy maximum;
- B. aki szülőként szerepel és a gyereke valahányszor szülő, akkor az unokák számát növeljük annyiival – ha mindenkinek ismert az unokaszáma, akkor azokra kell egy maximum;
- C. a B részben kiszámolt unokaszámok között hány 0 van!

A:

```

Gy[] := [0,...,0]
Ciklus i=1-től m-ig
    Gy[Szülő[i]] := Gy[Szülő[i]] + 1
Ciklus vége
Mgy:=1
Ciklus i=2-től n-ig
    Ha Gy[i]>Gy[Mgy] akkor Mgy:=i
Ciklus vége
Eljárás vége.

```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Középhaladó
Téma	Szülők
Feladat	4. A legtöbb gyerek

```

B:
  U[] := [0, ..., 0]
  Ciklus i=1-től M-ig
    Ciklus j=1-től M-ig
      Ha Gyerek[i]=Szülő[j] akkor U[Szülő[i]] := U[Szülő[i]] + 1
    Ciklus vége
  Ciklus vége
  Mu:=1
  Ciklus i=2-től n-ig
    Ha U[i]>U[Mu] akkor Mu:=i
  Ciklus vége
Eljárás vége.

```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Gráfok, elemi feladatok
Feladat	7. Ember

```

C:
  Db:=0
  Ciklus i=1-től n-ig
    Ha U[i]=0 akkor Db:=Db+1
  Ciklus vége
Eljárás vége.

```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Gráfok, elemi feladatok
Feladat	7. Ember

4. feladat

Egy állatkertben ismerjük a bejárható útvonalakat. A bejárat a 0 sorszámú pont. Az egyes állatokat az 1 és N közötti sorszámukkal azonosítjuk ($1 \leq N \leq 100$), az utakat pedig két olyan állat sorszámával (A,B), amelyek ketrece között vezetnek.

Készíts programot, amely az állatkerti utak ismeretében megadja, hogy hány olyan állat van, amelyik zsákutca végén található, valamint azt, hogy melyik állathoz vezet a legtöbb út (ha több is van, bármelyik megadható)!

Példa:

Bemenet:

Állatok száma: 5

Utak száma: 7

1. út: 0 1

2. út: 1 4

3. út: 3 1

4. út: 3 5

5. út: 2 0

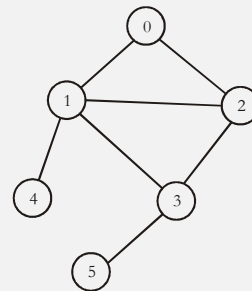
6. út: 2 3

7. út: 1 2

Kimenet:

Állatok száma zsákutca végén: 2

Legtöbb út: 1



Első lépésként minden állatra számoljuk meg, hogy oda hány út vezet (D)! Vigyázat, a 0. „állat” nem állat, hanem a bejárat!

A. ahova 1 út vezet, azok a zsákutcák, ezek számát kell megadni;

B. ahova a legtöbb út vezet, azt kell megadni!

A-B:

```
D[] := [0, ..., 0]
```

```
Ciklus i=1-től M-ig
```

```
  Ha A[i]>0 akkor D[A[i]] := D[A[i]] + 1
```

```
  Ha B[i]>0 akkor D[B[i]] := D[B[i]] + 1
```

```
Ciklus vége
```

```
Db:=0; Max:=1
```

```
Ciklus i=1-től N-ig
```

```
  Ha D[i]=1 akkor Db:=Db+1
```

```
  Ha D[i]>D[Max] akkor Max:=i
```

```
Ciklus vége
```

```
Eljárás vége.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Gráfok, elemi feladatok
Feladat	1. Állatkert

5. feladat

Ismerjük N településre az M napos időjárás előrejelzést, ezek alapján keressük a legmelegebb települést.

Készíts programot, amely megadja négyféle értelmezés szerint a legmelegebb települést:

- A. a legmelegebb település az, amelyre az előrejelzések maximuma a legnagyobb;
 B. a legmelegebb település az, amelyre az előrejelzések átlaga a legnagyobb;
 C. a legmelegebb település az, amelyben a leghosszabb időszakon belül várható folyamatosan K fok feletti hőmérséklet;
 D. a legmelegebb település az, amelyre a legtöbb napon fordul elő, hogy a várt hőmérséklet nagyobb minden más arra a napra előre jelzett hőmérsékletnél.

A standard bemenet első sorában a települések száma ($1 \leq N \leq 1000$), a napok száma ($1 \leq M \leq 1000$) és a hőmérséklet korlát van ($20 \leq K \leq 50$), egy-egy szóközzel elválasztva. A következő N sor mindegyikében M egész szám van, egy-egy szóközzel elválasztva: az i-edik település j-edik napra várt hőmérséklete.

A standard kimenet négy sorába egy-egy település sorszámát kell írni! Az első sorba az A, a másodikba a B, a harmadikba a C, a negyedikbe pedig a D szempont szerinti legmelegebb települést. Ha több megoldás van, bármelyik megadható. Ha nincs megoldás (C és D részfeladatban), akkor -1-et kell kiírni!

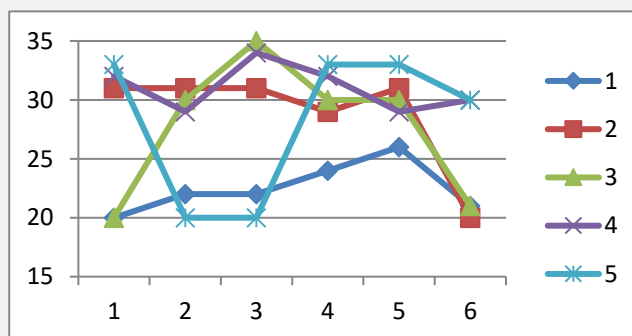
Példa:

bemenet

```
5 6 30
20 22 22 24 26 21
31 31 31 29 31 20
20 30 35 30 30 21
32 29 34 32 29 30
33 20 20 33 33 30
```

kimenet

```
3
4
2
5
```



- A. egy mátrix azon sorát kell megtalálni, ahol a legnagyobb érték szerepel;
 B. egy mátrix azon sorát kell megtalálni, ahol a sorösszeg a lehető legnagyobb;
 C. egy mátrix azon sorát kell megtalálni, ahol a lehető leghosszabb az az intervallum, amikor folyamatosan K fok feletti hőmérséklet;
 D. NEM ELEMI TÉTEL

A:

```
Max:=-Maxint
Ciklus i=1-től N-ig
  Ciklus j=1-től M-ig
    Ha H[i,j]>Max akkor A:=i; Max:=H[i,j]
  Ciklus vége
Ciklus vége
Eljárás vége.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Középhaladó
Téma	Időjárás
Feladat	9. Legmelegebb települések

B:

```

Max:=-Maxint
Ciklus i=1-től N-ig
    s:=0
    Ciklus j=1-től M-ig
        S:=S+H[i,j]
    Ciklus vége
    Ha S>Max akkor B:=i; Max:=s
Ciklus vége
Eljárás vége.

```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Középhaladó
Téma	Időjárás
Feladat	2. Átlagosan legmelegebb település

C:

```

Max:=-1; C:=-1
Ciklus i=1-től N-ig
    H[i,0]:=-Maxint; H[i,M+1]:=-Maxint; Ck:=0; Cv:=-1
    Ciklus j=1-től M-ig
        Ha H[i,j]>K és H[i,j-1]≤K akkor C1:=j
        Ha H[i,j]>K és H[i,j+1]≤K
            akkor Ha j-C1>Cv-Ck akkor Ck:=C1; Cv:=j
    Ciklus vége
    Ha Cv-Ck>Max akkor C:=i; Max:=Cv-Ck
Ciklus vége
Eljárás vége.

```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Tételek összeépítése
Feladat	13. Folyamatos meleg az előrejelzésben

D feladat

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Kezdő
Téma	Tételek összeépítése
Feladat	14. Folyamatosan legmelegebb település

6. feladat

Egy kieséses versenyben ismerjük a csapatok mérkőzéseit: ki kit győzött le.

Írj programot, amely megadja:

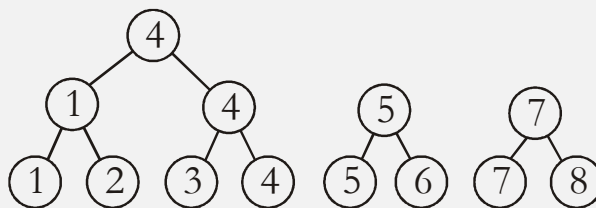
- A. a még versenyben levőket;
- B. azokat a csapatokat, amelyek legalább egyszer győztek, de már kiestek;
- C. a legtöbb csapatot közvetlenül vagy közvetve legyőző csapatot!

A standard bemenet első sorában a csapatok száma ($2 \leq N \leq 1000$) és a mérkőzések száma van ($1 \leq M < N$), egy szóközzel elválasztva. A következő M sor mindegyikében két csapat A és B sorszáma van ($1 \leq A \neq B \leq N$), ami azt jelenti, hogy az A -edik csapat legyőzte a B -edik csapatot.

A standard kimenet első sorába a még versenyben levő csapatok darabszámát, majd a sorszámát kell írni (egy-egy szóközzel elválasztva, növekvő sorrendben), a második sorba azok darabszámát, majd a sorszámát, amelyek úgy estek ki, hogy legalább egyszer győztek (egy-egy szóközzel elválasztva, növekvő sorrendben), a harmadik sorba pedig azt a csapatot, amely a legtöbb más csapatot győzte le közvetve vagy közvetlenül! Ha több megoldás van, bármelyik kiírható.

Példa:

bemenet	kimenet
8 5	3 4 5 7
1 2	1 1
4 3	4
4 1	
7 8	
5 6	



Először számoljuk meg mindenkire, hogy hányszor győzött (GY), illetve hányszor veszített (V)!
Utána a következőket kapjuk:

- A. azok vannak még versenyben, akik nem veszítettek;
- B. azokat a csapatokat, amelyek legalább egyszer győztek, de már kiestek, azaz veszítettek is;
- C. NEM ELEMI TÉTEL

Előkészítés:

```
Ciklus i=1-től m-ig
  Gy[A[i]]:=Gy[A[i]]+1; V[B[i]]:=V[B[i]]+1
Ciklus vége
```

Eljárás vége.

A. feladat:

```
A:=0
Ciklus i=1-től N-ig
  Ha V[i]=0 akkor A:=A+1; Ta[A]:=i
Ciklus vége
```

Eljárás vége.

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Rekurzív adatszerkezetek
Feladat	30. Kieséses verseny

B. feladat:

```
B:=0
Ciklus i=1-től N-ig
  Ha Gy[i]*V[i]>0 akkor B:=B+1; Tb[B]:=i
Ciklus vége
```

Eljárás vége.

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Rekurzív adatszerkezetek
Feladat	30. Kieséses verseny

C. feladat

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Rekurzív adatszerkezetek
Feladat	30. Kieséses verseny