

# Programozási tételek

---

*Jegyzet*

Összeállította: Faludi Anita  
2012.

# Tartalomjegyzék

|  |    |
|--|----|
| Bevezetés .....  | 3  |
| Programozási tételek .....                                       | 4  |
| I. Elemi programozási tételek .....                              | 4  |
| 1. Sorozatszámítás (összegzés) .....                             | 4  |
| 2. Eldöntés .....  | 5  |
| 3. Kiválasztás .....   | 7  |
| 4. Keresés .....   | 8  |
| 5. Megszámolás .....   | 10 |
| 5. Maximumkiválasztás .....                                      | 11 |
| II. Összetett programozási tételek .....                         | 13 |
| 1. Másolás .....   | 13 |
| 2. Kiválogatás .....   | 14 |
| a) Egy sorozathoz egy sorozatot rendelő feladattípusokhoz .....  | 14 |
| a) Egy sorozathoz több sorozatot rendelő feladattípusokhoz ..... | 16 |
| 3. Szétválogatás .....   | 18 |
| 4. Metszet .....   | 20 |
| 5. Egyesítés (unió) .....  | 21 |
| Felhasznált irodalom .....                                       | 24 |

## Bevezetés

Ez a jegyzet elsősorban azoknak a diákoknak készült, akiket tanítok, ezért a jegyzet erőteljesen hiányos. Az olvasó egy percig se gondolja azt, hogy a témakörhöz ennyi információ tartozik. A jegyzetben csak azokat a területeket érintettem, amit szükségesnek ítélttem meg, és amiről úgy gondoltam, hogy megfelelő alapot biztosít a továbbfejlődéshez.

## Programozási tételek

A programozási feladatok különböző feladatosztályokba sorolhatók a feladat jellege szerint. A különböző feladatosztályokhoz tartozó megoldó algoritmus-szabályt programozási tételnek nevezzük.

A feladatosztályokat négy típusra osztjuk:

- egy sorozathoz egy értéket rendelő feladatok,
- egy sorozathoz egy sorozatot rendelő feladatok,
- egy sorozathoz több sorozatot rendelő feladatok,
- több sorozathoz egy sorozatot rendelő feladatok.

## I. Elemi programozási tételek

### 1. Sorozatszámítás (összegzés)

Egy sorozathoz egy értéket rendelő feladattípusokhoz.

*Bemenet:* N, A(), művelet

*Kimenet:* S

*Változók:* N: egész [ megadott sorozat elemeinek száma ]

A: tömb (1..N) elemtípus [ a sorozat elemei ]

S: elemtípus [ végeredmény ]

*Algoritmus:*

```
Sorozatszámítás (N,A,S)
  S:= kezdőérték
  Ciklus i=1-től N-ig
    S:= S művelet A(i)
  Ciklus vége
Eljárás vége
```

**Feladatok:**

1. Adjunk össze N darab szomszédos egész számot!
2. Szorozzunk össze N darab szomszédos egész számot!
3. Határozzuk meg K darab egész szám négyzetének összegét, és szorzatát!
4. Határozzuk meg N darab szám átlagát!
5. N hónapon át törlesztettünk egy kölcsönt, havonkénti fizetéssel. (Havonként különböző összeget is fizethettünk). Adjuk meg az eddig kifizetett összeget!
6. Adott egy tekéző sorozata (melyik fordulóban hány fát ütött). Írjunk programot, amely meghatározza a versenyző összesített eredményét!

## 2. Eldöntés

Egy sorozathoz egy logikai értéket rendelő feladattípusokhoz. A logikai érték „igaz”-at vesz fel, ha a sorozatban létezik adott (T) tulajdonságú elem, és „hamis”-at vesz fel, ha ilyen tulajdonságú nincs a sorozatban.

*Bemenet:* N, A(), T tulajdonság

*Kimenet:* VAN: igaz/hamis

*Változók:* N: egész [ megadott sorozat elemeinek száma ]

A: tömb (1..N) elemtípus [ a sorozat elemei ]

VAN: logikai [ végeredmény ]

*Algoritmus:*

```
Eldöntés (N,A,VAN)
  i:= 1
  Ciklus amíg i≤N és A(i) nem T tulajdonságú
    i:= i+1
  Ciklus vége
  VAN:=(i≤N)
Eljárás vége
```

*Feladatok megoldással:*

1. Egy kosárlabda csapatnak van-e 210 cm-nél nagyobb játékosa?

*Bemenet:* N, A(), T tulajdonság -> **A(i)>210**

*Kimenet:* igaz -> **van ilyen magas játékos**, / hamis -> **nincs ilyen magas játékos**

*Változók:* N: egész [ **csapattagok száma** ]

A: tömb (1..N) **valós** [ **magasságok cm-ben** ]

VAN: logikai [ **van-e 210 cm-nél nagyobb játékos** ]

*Algoritmus:*

```
Eldöntés (N,A,VAN)
  i:= 1
  Ciklus amíg i≤N és A(i)≤210
    i:= i+1
  Ciklus vége
  VAN:=(i≤N)
Eljárás vége
```

2. Írjunk programot, amely egy osztály tanulói év végi osztályzatai ismeretében eldönti, hogy van-e bukott tanulója az osztálynak!

**Bemenet:** N, A(), T tulajdonság -> **A(i)=1**

**Kimenet:** igaz -> **van bukott tanuló**, / hamis -> **nincs bukott tanuló**

**Változók:** N: egész [ **tanulók száma** ]

A: tömb (1..N) **egész** [ **jegyek** ]

VAN: logikai [ **van-e bukott tanuló** ]

**Algoritmus:**

```
Eldöntés (N,A,VAN)
  i:= 1
  Ciklus amíg i≤N és A(i)≠1
    i:= i+1
  Ciklus vége
  VAN:=(i≤N)
Eljárás vége
```

**Feladatok:**

3. Írjunk programot, amely egy osztály nyilvántartásából megállapítja, hogy van-e év vesztes az osztályban!
4. Írjunk programot, amely egy szabászat személyi nyilvántartásából kideríti, hogy dolgozik-e férfi ezen a munkaterületen!
5. Adott egy 12. osztály tanulóinak nyilvántartása, valamint az aznapi dátum. Írjunk programot, amely megállapítja, hogy van-e olyan tanuló, aki nem nagykorú!
6. Írjunk programot, amely egy adott értelmes szövegről eldönti, hogy több szóból áll-e!
7. Írjunk programot, amely egy adott értelmes szövegről eldönti, hogy több mondatból áll-e!
8. Ismert az egymást követő N nyáron át a lehullott átlagos csapadékmennyiség (mm-ben). Ha 30 mm alatti érték van, akkor abban az évben központi öntözési támogatást kapnak a gazdaságok. Kellett-e a vizsgált időszakban ilyen támogatást adni?
9. Ismert N autó fogyasztása. (100 km-ként fogyasztott literben mérve). Döntsük el, hogy minden autó 10 liter alatt fogyasztott-e!
10. Adott a tanulók neve és magassága, névsor szerint rendezve. Döntsük el, hogy a névsor és a magasság szerinti sorrend azonos-e!

### 3. Kiválasztás

Egy sorozathoz egy értéket rendelő feladattípusokhoz.

Adott egy N elemű sorozat és egy, a sorozat elemein értelmezett T tulajdonság. Azt is tudjuk, hogy a sorozatban van legalább egy T tulajdonságú elem (előfeltétel). A feladat ezen elem sorszámának meghatározása.

*Bemenet:* N, A(), T tulajdonság

*Kimenet:* sorszám

*Változók:* N: egész [ megadott sorozat elemeinek száma ]

A: tömb (1..N) elemtípus [ a sorozat elemei ]

SORSZAM: egész [ T tulajdonságú elem sorszáma ]

*Algoritmus:*

```
Kiválasztás (N,A,SORSZAM)
  i:= 1
  Ciklus amíg A(i) nem T tulajdonságú
    i:= i+1
  Ciklus vége
  SORSZAM:= i
Eljárás vége
```

*Feladatok megoldással:*

1. Tekergő Gergő az állomáson éjszakázik. Hányszor ébreszti föl a hangosbemondó, mielőtt végre a várt vonatra szállhat?

*Bemenet:* N, A(), T tulajdonság -> **A(i)=indulási idő**

*Kimenet:* **ébredések**

*Változók:* N: egész [ **indulási időpontok száma** ]

A: tömb (1..N) **valós** [ **időpontok** ]

**EBRED:** egész [ **ébredések száma** ]

*Algoritmus:*

```
Kiválasztás (N,A,EBRED)
  i:= 1
  Ciklus amíg A(i) < indulási idő
    i:= i+1
  Ciklus vége
  EBRED:= i
Eljárás vége
```

### **Feladatok:**

2. Írjon programot, amely egy adott névsorban megmondja, hogy Kiss Pista hányadik sorszámú!
3. Készítsünk programot, amely egy hónapnévvel megadott dátumot átalakít hónapszámmal megadott dátummá!
4. Határozzuk meg, hogy egy adott hónap melyik évszakba esik!
5. Olvassunk be neveket addig, amíg nem írtunk egymás után két egyformát!
6. Lujzi barátja ragaszkodik a „Sport” mozihoz. Adjuk meg, hogy ebben a moziban mikor fogják vetíteni az „X” című filmet!
7. Határozzuk meg, hogy hány nap múlva lesz a következő vasárnap!
8. Addig kötünk fogadásokat, amíg először nem veszítünk. Állapítsuk meg, hányszor kötöttünk fogadást, összesen mennyit nyertünk!

## **4. Keresés**

Egy sorozathoz egy értéket rendelő feladattípusokhoz.

A feladat lényege, hogy a sorozatban egy adott T tulajdonságú elemet kell meghatároznunk (magát az elemet, vagy a sorozatbeli sorszámát szokás kérni). Nem biztos, hogy a sorozatban van ilyen elem.

*Bemenet:* N, A(), T tulajdonság

*Kimenet:* sorszám

*Változók:* N: egész [ megadott sorozat elemeinek száma ]  
A: tömb (1..N) elemtípus [ a sorozat elemei ]  
SORSZAM: egész [ a T tulajdonságú elem sorszáma ]

*Algoritmus:*

```
Keresés (N,A,SORSZAM)  
  i:= 1  
  Ciklus amíg i≤N és A(i) nem T tulajdonságú  
    i:= i+1  
  Ciklus vége  
  VAN:=(i≤N)  
  Ha VAN akkor SORSZAM:=i  
Eljárás vége
```



### Feladatok megoldással:

1. Adott egy vektor. Határozzuk meg egy negatív elemének indexét!

Bemenet: N, A(), T tulajdonság -> **A(i)<0**

Kimenet: **index**

Változók: N: egész [ vektor elemeinek száma ]

A: tömb (1..N) **valós** [ számokat tartalmazó vektor ]

**INDEX**: egész [ hányadik az első negatív szám ]

Algoritmus:

```
Keresés (N,A,INDEX)
  i:= 1
  Ciklus amíg i≤N és A(i) ≥ 0
    i:= i+1
  Ciklus vége
  VAN:=(i≤N)
  Ha VAN akkor INDEX:=i
Eljárás vége
```

2. Egy csomag kártyát összekeverték. Adjuk meg, hogy hol van benne joker (ha van egyáltalán)!

Bemenet: N, A(), T tulajdonság -> **A(i)=joker**

Kimenet: hely

Változók: N: egész [ kártyalapok száma ]

A: tömb (1..N) **szöveg** [ kártyalapok elnevezését tartalmazó tömb ]

**HELY**: egész [ joker helye a sorozatban ]

Algoritmus:

```
Keresés (N,A,HELY)
  i:= 1
  Ciklus amíg i≤N és A(i)≠joker
    i:= i+1
  Ciklus vége
  VAN:=(i≤N)
  Ha VAN akkor HELY:=i
Eljárás vége
```

### **Feladatok:**

3. Egy kosárlabdacsapat nyilvántartása többek között tartalmazza a játékosok nevét és magasságát. Írjunk programot, amely kiírja egy 210 cm-nél magasabb játékos nevét!
4. A vasúti nyilvántartás tartalmazza a Savaria expresszre kiadott helyjegyeket. Írjunk programot, amely meghatároz egy szabad helyet a vonaton!
5. Adott egy táblázat, amelynek elemei karakterek (üres helyek is lehetnek benne). Írjunk programot, amely választ a táblázatban egy véletlen helyet, majd kiválaszt hozzá egy üres szomszédot!
6. Egy házi telefonkönyv a nevek szerint rendezett. Keressünk meg benne egy adott névhez tartozó telefonszámot!
7. Egy csomag kártyát összekeverték.
  - a) Adjuk meg, hogy hol van egymás mellett két azonos színű lap!
  - b) Adjuk meg, hogy hol van egymás mellett két azonos figurájú lap!
  - c) Adjuk meg, hogy hol van benne joker (ha van benne egyáltalán)!
8. Nyelvvizsgán a nyelvtani tesztek pontszámait ülési sorrendben jegyezték fel. Keressünk olyan vizsgázót, aki ugyanannyi pontot kapott, mint a szomszédja! (A tesztlapokat ülésorrendben szedték össze.)

## **5. Megszámolás**

Egy sorozathoz egy értéket rendelő feladattípusokhoz.

Rendelkezésünkre áll egy  $N$  elemű sorozat és egy, a sorozat elemein értelmezett  $T$  tulajdonság. Az a feladat, hogy a  $T$  tulajdonsággal rendelkező elemeket megszámloljuk.

*Bemenet:*  $N, A(), T$  tulajdonság

*Kimenet:*  $DB$

*Változók:*  $N$ : egész [ megadott sorozat elemeinek száma ]

$A$ : tömb (1.. $N$ ) elemtípus [ a sorozat elemei ]

$DB$ : egész [  $T$  tulajdonságú elemek darabszáma ]

*Algoritmus:*

**Megszámolás ( $N, A, DB$ )**

**$DB := 0$**

**Ciklus  $i=1$ -től  $N$ -ig**

**Ha  $A(i)$   $T$  tulajdonságú akkor  $DB := DB + 1$**

**Ciklus vége**

**Eljárás vége**

### Feladatok megoldással:

1. A testnevelés órai időméréses futóverseny alapján állapítsuk meg, hogy a diákok hány százaléka teljesítette az 5-ös szintet!

Bemenet: N, A(), T tulajdonság -> **A(i) <= 5 (5 perc az ötös szint)**

Kimenet: **DARAB**

Változók: N: egész [ osztály létszáma ]

A: tömb (1..N) **valós** [ futási idők ]

**DARAB**: egész [ ötösök darabszáma ]

Algoritmus:

```
Megszámolás (N, A, DARAB)
  DARAB := 0
  Ciklus i=1-től N-ig
    Ha A(i) <= 5 akkor DARAB := DARAB + 1
  Ciklus vége
Eljárás vége
```

### Feladatok:

2. Határozzuk meg az A(N) vektor negatív elemeinek számát!
3. Számoljuk meg, hogy hányszor fordul elő egy szövegben az „a” névelő!
4. Határozzuk meg az A(N) vektor azon pozitív elemeinek számát, amelyek közvetlenül egy negatív elem után állnak!
5. Adott az A(N) vektor és a K szám. Állapítsuk meg, hogy a vektor elemei között hányszor szerepel a K!
6. Állapítsuk meg, hogy az A(N) vektorban negatív vagy pozitív számból van-e több!
7. Határozzuk meg az S karaktersorozatban a magánhangzók számát!

## 5. Maximumkiválasztás

Egy sorozathoz egy értéket rendelő feladattípusokhoz.

Adott egy N elemű sorozat. A feladat ezen sorozat legnagyobb elemének (illetve feladattól függően legnagyobb értékének) meghatározása. Hasonló feladat – csak a relációt kell megfordítani – a minimumkiválasztás.

Bemenet: N, A()

Kimenet: INDEX, **MAXIMUM** (feladattól függően)

Változók: N: egész [ megadott sorozat elemeinek száma ]

A: tömb (1..N) elemtípus [ a sorozat elemei ]

INDEX: egész [ maximális érték helye a sorozatban ]

**MAXIMUM**: elemtípus [ maximális érték ]

Algoritmus:

```
Maxinex (N,A,INDEX,MAXIMUM)
  INDEX:= 1
  Ciklus i=2-től N-ig
    Ha A(i) > A(INDEX) akkor INDEX:= i
  Ciklus vége
  MAXIMUM := A(INDEX)
Eljárás vége
```

**Feladatok megoldással:**

1. Egy osztály tanulóinak testmagasságát cm-ben kaptuk meg.
  - a) Melyik tanuló a legmagasabb?
  - b) Milyen magas a legmagasabb tanuló?

Bemenet: LETSZAM(), MAGAS()

Kimenet: IND, MAX

Változók: LETSZAM: egész [ osztály létszáma ]

MAGAS: tömb (1..N) **valós** [ magasságok ]

IND: egész [ hányadik helyen van a legmagasabb ]

MAX: valós [ hány cm a legmagasabb tanuló ]

Algoritmus:

```
Maximum (LETSZAM,MAGAS,IND,MAX)
  IND:=1
  Ciklus i=2-től LETSZAM-ig
    Ha MAGAS(i) > MAGAS(IND) akkor IND:=i
  Ciklus vége
  MAX:= MAGAS(IND)
Eljárás vége
```

**Feladatok:**

2. 5 szám közül válasszuk ki a legnagyobbat és a legkisebbet!
3. Határozzuk meg egy N elemű sorozat legkisebb pozitív elemét!
4. Keressük meg az N elemű sorozatnak azt a legnagyobb elemét, amely nagyobb az előtte lévő elemnél, de kisebb az őt követőnél!
5. Egy folyó sodrára merőlegesen H méterenként megmértük a folyó mélységét. Állapítsuk meg, hol a legmeredekebb a meder? Mekkora ez a meredekség?
6. A rádió reggeli időjárás-jelentése alapján állapítsuk meg, hogy az ország melyik városában van a leghidegebb!

## II. Összetett programozási tételek

### 1. Másolás

Egy sorozathoz egy másik sorozatot rendelő feladattípusokhoz.

Adott egy N elemű sorozat. A feladat ezen sorozat lemásolása, s közben egy (elemre vonatkozó) átalakítást lehet végezni. Az eredmény mindig ugyanannyi elemszámú.

*Bemenet:* N, A()

*Kimenet:* B()

*Változók:* N: egész [ megadott sorozat elemeinek száma ]

A: tömb (1..N) elemtípus [ a sorozat elemei ]

B: tömb (1..N) elemtípus [ az új sorozat elemei ]

*Algoritmus:*

```
Másolás (N,A,B)
  Ciklus i=1-től N-ig
    B(i) := A(i) // lehetséges művelet végzése A(i)-vel
  Ciklus vége
Eljárás vége
```

*Feladatok megoldással:*

1. Egy névben cseréljük ki az összes kisbetűt nagyra!

*Bemenet:* SHOSSZ, nevkis()

*Kimenet:* nevnagy()

*Változók:* SHOSSZ: egész [ karakterek száma ]

nevkis: tömb (1..SHOSSZ) karakter [ kisbetűs név ]

nevnagy: tömb (1..SHOSSZ) karakter [ nagybetűs név ]

*Algoritmus:*

```
Másolás (SHOSSZ, nevkis, nevnagy)
  Ciklus amíg i=1-től SHOSSZ-ig
    nevnagy(i) := nevkis(i) nagybetűvel
  Ciklus vége
Eljárás vége
```

2. Egy születési évekből álló sorozat minden eleméről állapítsuk meg, hogy az illető idős ( $\geq 60$ ), középkorú ( $\geq 30$  és  $< 60$ ), vagy fiatal ( $< 30$ )!

Bemenet: DB, evszamok()

Kimenet: jellemzés()

Változók: DB: egész [ arakterek száma ]

evszamok: tömb (1..DB) egész [ születési évszámok ]

jellemzes: tömb (1..DB) szöveg [ szöveges jellemzések a korról ]

Algoritmus:

```

Másolás (DB, evszamok, jellemzes)
  Ciklus amíg i=1-től DB-ig
    Ha evszamok(i) >= 60 akkor
      jellemzes(i) := "idős"
    különben
      Ha evszamok(i) >= 30 akkor
        jellemzes(i) := "közép korú"
      különben
        jellemzes(i) := "fiatal"
    Elágazás vége
  Ciklus vége
Eljárás vége

```

**Feladatok:**

3. Egy szövegben cseréljük ki az összes magánhangzót „e” betűre!
4. Egy osztály tanulóinak átlageredménye alapján határozzuk meg, hogy a bizonyítványba milyen szöveg kerüljön (bukott tanuló nincs)!

## 2. Kiválogatás

### a) Egy sorozathoz egy sorozatot rendelő feladattípusokhoz.

Rendelkezésünkre áll egy N elemű sorozat és egy, a sorozat elemein értelmezett T tulajdonság. Az a feladat, hogy a T tulajdonsággal rendelkező elemeket meghatározzuk egy másik sorozatban.

Bemenet: N, A(), T tulajdonság

Kimenet: DB, B()

Változók: N: egész [ megadott sorozat elemeinek száma ]

A: tömb (1..N) elemtípus [ a sorozat elemei ]

DB: egész [ T tulajdonságú elemek darabszáma ]

B: tömb (1..N) elemtípus [ T tulajdonságú elemek ]

Algoritmus:

```
Kiválogatás (N,A,DB,B)
  DB:= 0
  Ciklus i=1-től N-ig
    Ha A(i) T tulajdonságú akkor
      DB:=DB+1
      B(DB) :=A(i)
  Ciklus vége
Eljárás vége
```

**Feladat megoldással:**

1. Határozzuk meg, hogy egy adott névsorban kiknek kezdődik „B” betűvel a neve!

Bemenet: NevDB, nevek(), T tulajdonság -> nevek(i) első betűje „B”

Kimenet: DARAB, Bnevek()

Változók: NevDB: egész [ nevek száma ]  
nevek: tömb (1..NevDB) szöveg [ nevek tömbje ]  
DARAB: egész [ „B” betűs nevek száma ]  
Bnevek: tömb (1..NevDB) szöveg [ „B” betűs nevek tömbje ]

Algoritmus:

```
Kiválogatás (NevDB, nevek, DARAB, Bnevek)
  DARAB:=0
  Ciklus i=1-től N-ig
    Ha nevek(i) első betűje ="B" akkor
      DARAB:=DARAB+1
      Bnevek(DARAB) :=nevek(i)
  Ciklus vége
Eljárás vége
```

**Feladatok:**

2. Határozzuk meg az A(N) vektor negatív elemeit!
3. Határozzuk meg az A(N) vektor azon pozitív elemeinek számát, amelyek közvetlenül egy negatív elem után állnak!

### a) Egy sorozathoz több sorozatot rendelő feladattípusokhoz.

Rendelkezésünkre áll egy  $N$  elemű sorozat és egy, a sorozat elemein értelmezett  $T_1, T_2, \dots, T_x$  tulajdonságok. Az a feladat, hogy a különböző tulajdonságok szerint hozzunk létre új sorozatokat.

*Bemenet:*  $N, A(), T_1, T_2, \dots, T_x$  ( $x$  darab tulajdonság)

*Kimenet:*  $DB_1, DB_2, \dots, DB_x, TOMB_1(), TOMB_2(), \dots, TOMB_x()$

*Változók:*

|            |                           |  |
|------------|---------------------------|--|
| $N$ :      | egész                     | [ megadott sorozat elemeinek száma ]     |
| $A$ :      | tömb (1.. $N$ ) elemtípus | [ a megadott sorozat elemei ]            |
| $DB_1$ :   | egész                     | [ $T_1$ tulajdonságú elemek darabszáma ] |
| $DB_2$ :   | egész                     | [ $T_2$ tulajdonságú elemek darabszáma ] |
| ...        |                           |  |
| $DB_x$ :   | egész                     | [ $T_x$ tulajdonságú elemek darabszáma ] |
| $TOMB_1$ : | tömb (1.. $N$ ) elemtípus | [ $T_1$ tulajdonságú elemek ]            |
| $TOMB_2$ : | tömb (1.. $N$ ) elemtípus | [ $T_2$ tulajdonságú elemek ]            |
| ...        |                           |  |
| $TOMB_x$ : | tömb (1.. $N$ ) elemtípus | [ $T_x$ tulajdonságú elemek ]            |

*Algoritmus:*

```
Kiválogatás (N,A,DB1,DB2,...,DBx,TOMB1,TOMB2,...,TOMBx)
DB1:= 0; DB2:= 0; ... ; DBx:= 0;
Ciklus i=1-től N-ig
    A(i) T1 tulajdonsága esetén
        DB1:=DB1+1;
        TOMB1(DB1):=A(i);
    A(i) T2 tulajdonsága esetén
        DB2:=DB2+1;
        TOMB2(DB2):=A(i);
    ...
    A(i) Tx tulajdonsága esetén
        DBx:=DBx+1;
        TOMBx(DBx):=A(i);
Ciklus vége
Eljárás vége
```



### Feladat megoldással:

1. Külön adjuk meg egy névsor „A” , „B” és „C” betűvel kezdődő neveit!

Bemenet: NevDB, nevek(), T1 tulajdonság -> nevek(i) első betűje „A”

T2 tulajdonság -> nevek(i) első betűje „B”

T3 tulajdonság -> nevek(i) első betűje „C”

Kimenet: aDB, aNevek(), bDB, bNevek(), cDB, cNevek(),

Változók: NevDB: egész [ nevek száma ]  
nevek: tömb (1..N) szöveg [ nevek tömbje ]  
aDB: egész [ „A” betűs nevek száma ]  
bDB: egész [ „B” betűs nevek száma ]  
cDB: egész [ „C” betűs nevek száma ]  
aNevek: tömb (1..N) szöveg [ „A” betűs nevek tömbje ]  
bNevek: tömb (1..N) szöveg [ „B” betűs nevek tömbje ]  
cNevek: tömb (1..N) szöveg [ „C” betűs nevek tömbje ]

### Algoritmus:

```
Kiválogatás (NevDB, nevek, aDB, bDB, cDB, aNevek, bNevek, cNevek)
  aDB:=0 bDB:=0 cDB:=0
  Ciklus i=1-től N-ig
    nevek(i) első betűje ="A" esetén
      aDB:=aDB+1
      aNevek(aDB) :=nevek(i)
    nevek(i) első betűje ="B" esetén
      bDB:=bDB+1
      bNevek(bDB) :=nevek(i)
    nevek(i) első betűje ="C" esetén
      cDB:=cDB+1
      cNevek(cDB) :=nevek(i)
  Ciklus vége
Eljárás vége
```

### Feladatok:

2. Határozzuk meg az A(N) vektor negatív elemeit!
3. Határozzuk meg az A(N) vektor azon pozitív elemeinek számát, amelyek közvetlenül egy negatív elem után állnak!

### 3. Szétválogatás

Egy sorozathoz két sorozatot rendelő feladattípusokhoz.

Rendelkezésünkre áll egy  $N$  elemű sorozat és egy, a sorozat elemein értelmezett  $T$  tulajdonság. Az a feladat, hogy a  $T$  tulajdonsággal rendelkező és a  $T$  tulajdonsággal nem rendelkező elemeket meghatározzuk egy-egy külön sorozatban.

*Bemenet:*  $N, A(), T$  tulajdonság

*Kimenet:*  $DB1, B(), DB2, C()$

*Változók:*

|         |                           |  |
|---------|---------------------------|--|
| $N$ :   | egész                     | [ megadott sorozat elemeinek száma ]       |
| $A$ :   | tömb (1.. $N$ ) elemtípus | [ a sorozat elemei ]                       |
| $DB1$ : | egész                     | [ $T$ tulajdonságú elemek darabszáma ]     |
| $B$ :   | tömb (1.. $N$ ) elemtípus | [ $T$ tulajdonságú elemek ]                |
| $DB2$ : | egész                     | [ nem $T$ tulajdonságú elemek darabszáma ] |
| $C$ :   | tömb (1.. $N$ ) elemtípus | [ nem $T$ tulajdonságú elemek ]            |

*Algoritmus:*

```
Szétválogatás (N,A,DB1,B,DB2,C)
DB1:= 0
DB2:= 0
Ciklus i=1-től N-ig
    Ha A(i) T tulajdonságú akkor
        DB1:=DB1+1
        B(DB1) :=A(i)
    Különben
        DB2:=DB2+1
        C(DB2) :=A(i)
Ciklus vége
Eljárás vége
```

### Feladat megoldással:

1. Adottak egy sorozat elemei, válogassuk szét a pozitív és negatív számokat!  
(feltételezzük, hogy nincs 0 értékű elem)

Bemenet: DB, **szamok()**, T tulajdonság -> **szamok(i) < 0**

Kimenet: **NDB, neg(), PDB, poz()**

Változók: DB: egész [ számok darabszáma ]  
szamok: tömb (1..DB) **valós** [ számok tömbje ]  
NDB: egész [ negatív számok darabszáma ]  
neg: tömb (1..DB) **valós** [ negatív számok tömbje ]  
PDB: egész [ pozitív számok darabszáma ]  
poz: tömb (1..DB) **valós** [ pozitív számok tömbje ]

### Algoritmus:

```
Szétválogatás (DB, szamok, NDB, neg, PDB, poz)
  NDB:=0
  PDB:=0
  Ciklus i=1-től N-ig
    Ha szamok(i) < 0 akkor
      NDB:=NDB+1
      neg(NDB) :=szamok(i)
    Különben
      PDB:=PDB+1
      poz(PDB) :=szamok(i)
  Ciklus vége
Eljárás vége
```

### Feladatok:

2. Egy tömbben keresztnevek szerepelnek. Válogassuk szét őket fiú- és lánynévre azzal a nagyvonalú közelítéssel, hogy a lánynevek utolsó betűi magánhangzók szoktak lenni.
3. Gyűjtsük ki külön tömbökbe a férfiak illetve nők személyi számait!

#### 4. Metszet

Több sorozathoz egy sorozatot rendelő feladatokhoz.

Rendelkezésünkre áll két sorozat. Az a feladat, hogy egy harmadik sorozatba azokat az elemeket válasszuk ki, amelyek mindkét sorozatban megtalálhatók.

*Bemenet:* N, A(), M, B()

*Kimenet:* DB, C()

*Változók:* N, M: egész [ megadott sorozatok elemeinek száma ]  
A, B: tömb (1..N) elemtípus [ a sorozatok elemei ] //EF: N>M  
DB: egész [ metszet sorozat elemeinek darabszáma ]  
C: tömb (1..N) elemtípus [ metszet sorozat elemei ]

*Algoritmus:*

```
Metszet (N,A,M,B,DB,C)
  DB:= 0
  Ciklus i=1-től N-ig
    j:=1
    Ciklus amíg i<=M és A(i)<>B(j)
      j:=j+1
    Ciklus vége
    Ha j<=M akkor
      DB:=DB+1
      C(DB):=A(i)
  Ciklus vége
Eljárás vége
```

**Feladat megoldással:**

1. Matematika és fizika tantárgyból az iskola versenyt rendez. Adottak mindkét verseny indulói külön listában. Adjuk meg, hogy kik azok, akik mindkét versenyen indulnak!

*Bemenet:* mDB, mindulok(), fDB, findulok()

*Kimenet:* DB, mindketto()

*Változók:* mDB: egész [ matekon indulók száma ]  
fDB: egész [ fizikán idulók száma ]  
mindulok: tömb (1..mDB) **szöveg** [ matekon indulók nevei ]  
findulok: tömb (1..fDB) **szöveg** [ fizikán indulók nevei ]  
DB: egész [ pozitív számok darabszáma ]  
mindketto: tömb (1..mDB) **szöveg** [ mindkét versenyen indulók nevei ]

Algoritmus:

```
Metszet (mDB, mindulok, fDB, findulok, DB, mindketto)
  DB:=0
  Ciklus i=1-től mDB-ig
    j:=1
    Ciklus amíg j<=fDB és mindulok(i)<>findulok(j)
      j:=j+1
    Ciklus vége
    Ha j<=fDB akkor
      DB:=DB+1
      mindketto(DB)=mindulok(i)
  Ciklus vége
Eljárás vége
```

**Feladatok:**

2. Adjuk meg két természetes szám közös osztóit!
3. Négy ember szabad estéi ismeretében adjuk meg, hogy mikor tudnak találkozni!
4. Adot két étterem kínálata. Melyek azok az ételek, amelyek mindkét étteremben rendelhetők?
5. Adott két szerző műveinek listája. Van-e közös könyvük? Ha igen, melyek ezek?
6. Nyáron és télen is végeztünk madármegfigyeléseket a Balatonon. Adjuk meg a nem költöző madarakat!
7. Egy lóversenyen ugyanazoknak a lovaknak két fordulóban kell helytállniuk. Adjuk meg azokat a lovakat, amelyek mindkét fordulóban kiestek!

## 5. Egyesítés (unió)

Több sorozathoz egy sorozatot rendelő feladatokhoz.

Rendelkezésünkre áll két sorozat. Az a feladat, hogy egy harmadik sorozatba azokat az elemeket válasszuk ki, amelyek legalább az egyik sorozatban megtalálhatók.

*Bemenet:* N, A(), M, B()

*Kimenet:* DB, C()

*Változók:* N, M: egész [ megadott sorozatok elemeinek száma ]  
A, B: tömb (1..N) elemtípus [ a sorozatok elemei ] //EF: N>M  
DB: egész [ egyesített sorozat elemeinek darabszáma ]  
C: tömb (1..N+M) elemtípus [ egyesített sorozat elemei ]

Algoritmus:

```
Egyesítés (N,A,M,B,DB,C)
C:= A; DB:= N;
Ciklus j=1-től M-ig
    i:=1;
    Ciklus amíg i<=N és A(i)<>B(j)
        i:=i+1;
    Ciklus vége
    Ha j>M akkor
        DB:=DB+1; C(DB):=B(j);
    Ciklus vége
Eljárás vége
```

**Feladat megoldással:**

1. Az iskolából két csapat jutott tovább a kerületi „mindentudó” versenyre. Állítson össze egy listát arról, hogy az iskola milyen tantárgyakból ért el helyezést!

Bemenet: aDB, Acsapat(), bDB, Bcsapat()

Kimenet: iDB, icsapat()

Változók: aDB: egész [ A csapat helyezéseinek száma ]  
bDB: egész [ B csapat helyezéseinek száma ]  
Acsapat: tömb (1..aDB) **szöveg** [ A csapat helyezéses tantárgyai ]  
Bcsapat: tömb (1..bDB) **szöveg** [ B csapat helyezéses tantárgyai ]  
iDB: egész [ Iskola helyezéseinek száma ]  
icsapat: tömb (1..aDB+bDB) **szöveg** [ Iskola helyezéses tantárgyai ]

Algoritmus:

```
Egyesítés (aDB,Acsapat,bDB,Bcsapat,iDB,icsapat)
icsapat:=Acsapat; iDB:=aDB;
Ciklus j=1-től bDB-ig
    i:=1;
    Ciklus amíg i<=aDB és Acsapat(i)<>Bcsapat(j)
        i:=i+1;
    Ciklus vége
    Ha i>aDB akkor
        iDB:=iDB+1; icsapat(iDB)=Bcsapat(j);
    Ciklus vége
Eljárás vége
```

**Feladatok:**

2. Adot két étterem kínálata. Melyek azok az ételek, amelyek legalább az egyik étteremben rendelhetők?
3. Egy lóversenyen ugyanazoknak a lovaknak két fordulóban kell helytállniuk. Adjuk meg azokat a lovakat, amelyek valamelyik fordulóban kiestek!
4. Egy nagy utazó minden évben feljegyzi uticéljait. Az elmúlt két év feljegyzése alapján adjuk meg, hogy mely országokban járt!

## Felhasznált irodalom

μlógia 19.

Szlávi Péter – Zsakó László: Módszeres programozás: Programozási tétel

ELTE, Budapest, 1999.

Programozási feladatok I. – összeállította: Zsakó László

Kossuth Kiadó, 1997.