





# A PROGRAMOZÁS ALAPJAI 1

## 3. előadás

Vitéz András  
egyetemi adjunktus  
BME Híradástechnikai Tanszék  
vitez@nit.bme.hu

2012. február 21.



# Miről lesz ma szó?

- Adatvektor fogalma
- Végeles sorozat (null-terminált sorozat)
- Számok, karakterek
- Elemi vektor-skálár algoritmusok (programozási tételek)
  - Értékek összege/szorzata/átlaga
  - Leszámlálás
  - Értékek minimuma/maximuma
- Tömb: memóriában tárolt vektor
- További elemi vektor-skálár/vektor-vektor algoritmusok
  - Eldöntés
  - Leválogatás
  - Helyben szétválogatás

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



# Adatvektor fogalma

Azonos típusú adatok véges sorozata

A sorrend számít!

Nem kell feltétlenül a memóriában tárolnunk:  
érkezhet sorosan a program bemeneteként.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



# Végeles sorozat (null-terminált sorozat)

Ha beolvassuk, tudnunk kell, hány adat lesz.

**Legalább is a beolvasás végére tudnunk kell!**

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



# Végeles sorozat (null-terminált sorozat)

Itt még nincs végjel!

1. lehetőség

Először beolvassuk a darabszámot, azután ciklusban beolvassuk, és feldolgozzuk az ígért számú adatot.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



# Végeles sorozat (null-terminált sorozat)

2. lehetőség

Nem tudjuk előre a darabszámot, de ciklusban beolvassuk, és feldolgozzuk az adatokat, amíg egy előre megbeszélt (a többi adattal össze nem téveszthető) adatot nem kapunk.

Például pozitív egészek feldolgozása esetén nulla jelezheti az adatok végét.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Számok, karakterek**

Hogy programot tudjunk írni,  
legalább néhány adattípust ismernünk kell.

Most éppen csak hallunk róluk.

Később részletesen megismerjük őket.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Számok, karakterek**

**int**

Előjeles egész számok fixpontos ábrázolása

A C nyelv legfontosabb alaptípusa

Belső ábrázolása általában  
**kettes komplementes kód**

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Számok, karakterek**

**double**

Előjeles valós számok növelt pontosságú  
lebegőpontos ábrázolása

A C nyelv legfontosabb aritmetikai típusa

Belső ábrázolása  
**normalizált mantissza + karakterisztika**

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Számok, karakterek**

**char**

Karakterek (betűk, számjegyek, írásjelek)  
ábrázolása

A C nyelv legkisebb tárhelyet foglaló típusa

Ebben fogjuk mérni a többi típus helyigényét.

Belső ábrázolása  
**egész számként**

Ha aritmetikai kifejezésben szerepel,  
először **int**-té alakul.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Számok, karakterek**

**igazságérték**

Ilyen típusa a C nyelvnek nincs!

Láttuk, hogyan értelmezzük a kifejezéseket  
igazságértékként.

Azok az operátorok, amelyek igazságértéket  
szolgáltatóknak (pl. hasonlító relációk),  
**int** típusú eredményt generálnak:  
**HAMIS** esetben 0-t, **IGAZ** esetben 1-et.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skalár algoritmusok  
(programozási tételek)**

**Vissza a vektorokhoz!**

Az 1. megoldás folyamata

Változók:

- n: az elemek száma
- i: segédváltozó az elemek számlálásához
- a: az éppen feldolgozandó adatelem

```

    graph TD
      Start([Vektor_feldolg_1]) --> BE_n[BE: n]
      BE_n --> i_0[i ← 0]
      i_0 --> Decision{i < n}
      Decision -- N --> End([ ])
      Decision -- I --> BE_a[BE: a]
      BE_a --> feldolgozas[feldolgozás]
      feldolgozas --> n_plus_1[n ← n + 1]
      n_plus_1 --> Decision
  
```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skálár algoritmusok (programozási tételek)**

### Vissza a vektorokhoz!

A 2. megoldás folyamata (null-terminált sorozat)

Változók:  
a: az éppen feldolgozandó adatelem

Nem kellett segédváltozó, de minden adatot kétszer tesztlünk!

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Informatikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skálár algoritmusok (programozási tételek)**

### Null-terminált sorozat feldolgozása

Szervezzük ügyesebben!

Változók:  
a: az éppen feldolgozandó adatelem

A feldolgozás keretét megszerveztük, de még szükségünk lesz néhány adminisztratív kiegészítő tevékenységre!

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Informatikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skálár algoritmusok (programozási tételek)**

### Null-terminált sorozat feldolgozása

Végleges változat

Változók:  
a: az éppen feldolgozandó adatelem

A különböző feladatok megoldásához csak a színes blokkokat kell megvalósítani.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Informatikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skálár algoritmusok (programozási tételek)**

### Null-terminált sorozat feldolgozása

Struktogram

Változók:  
a: az éppen feldolgozandó adatelem

Most már kódolhatjuk!

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Informatikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skálár algoritmusok (programozási tételek)**

### Null-terminált sorozat feldolgozása

A keretprogram

```
#include <stdio.h>
int main() {
    /* DEKLARÁCIÓK */
    int a;
    scanf("%d", &a);
    /* ELŐKÉSZÍTÉS */
    while (a) {
        /* FELDOLGOZÁS */
        scanf("%d", &a);
    }
    /* VÁLASZ */
}
```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Informatikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skálár algoritmusok (programozási tételek)**

### Értékek összege

Felvesszünk egy segédváltozót. Ebben fogjuk számolni az összeget.

Kezdetben nullára állítjuk.

Minden egyes értéket hozzáadunk.

Végül kiírjuk a kapott eredményt.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Informatikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Elemi vektor-skalár algoritmusok (programozási tételek)

### Értékek összege

Felvezünk egy segédváltozót.  
Ebben fogjuk számolni az összeget.

Kezdetben nullára állítjuk.

Minden egyes értéket hozzáadunk.

Végül kiírjuk a kapott eredményt.

```
#include <stdio.h>
int main() {
    /* DEKLARÁCIÓK */
    int a;
    scanf("%d", &a);
    /* ELŐKÉSZÍTÉS */
    while (a) {
        /* FELDOLGOZÁS */
        scanf("%d", &a);
    }
    /* VÁLASZ */
}
```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Elemi vektor-skalár algoritmusok (programozási tételek)

### Értékek összege

Felvezünk egy segédváltozót.  
Ebben fogjuk számolni az összeget.

Kezdetben nullára állítjuk.

Minden egyes értéket hozzáadunk.

Végül kiírjuk a kapott eredményt.

```
#include <stdio.h>
int main() {
    int sum;
    int a;
    scanf("%d", &a);
    sum=0;
    while (a) {
        sum=sum+a;
        scanf("%d", &a);
    }
    printf("%d\n", sum);
}
```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Elemi vektor-skalár algoritmusok (programozási tételek)

### Értékek szorzata

Felvezünk egy segédváltozót.  
Ebben fogjuk számolni a szorzatot.

Kezdetben egyre állítjuk.

Minden egyes értékkel megszorozzunk.

Végül kiírjuk a kapott eredményt.

```
#include <stdio.h>
int main() {
    /* DEKLARÁCIÓK */
    int a;
    scanf("%d", &a);
    /* ELŐKÉSZÍTÉS */
    while (a) {
        /* FELDOLGOZÁS */
        scanf("%d", &a);
    }
    /* VÁLASZ */
}
```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Elemi vektor-skalár algoritmusok (programozási tételek)

### Értékek szorzata

Felvezünk egy segédváltozót.  
Ebben fogjuk számolni a szorzatot.

Kezdetben egyre állítjuk.

Minden egyes értékkel megszorozzunk.

Végül kiírjuk a kapott eredményt.

```
#include <stdio.h>
int main() {
    /* DEKLARÁCIÓK */
    int a;
    scanf("%d", &a);
    /* ELŐKÉSZÍTÉS */
    while (a) {
        /* FELDOLGOZÁS */
        scanf("%d", &a);
    }
    /* VÁLASZ */
}
```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Elemi vektor-skalár algoritmusok (programozási tételek)

### Értékek szorzata

Felvezünk egy segédváltozót.  
Ebben fogjuk számolni a szorzatot.

Kezdetben egyre állítjuk.

Minden egyes értékkel megszorozzunk.

Végül kiírjuk a kapott eredményt.

```
#include <stdio.h>
int main() {
    int prod;
    int a;
    scanf("%d", &a);
    prod=1;
    while (a) {
        prod=prod*a;
        scanf("%d", &a);
    }
    printf("%d\n", prod);
}
```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Elemi vektor-skalár algoritmusok (programozási tételek)

### Értékek átlaga

Felvezünk két segédváltozót.  
Egyet az összeg kiszámolásához.  
*Lebegőpontosat, hogy az osztás majd helyes eredményt adjon!*  
Egyet az értékek számlálásához.

Kezdetben mindkettőt nullára állítjuk.

Minden egyes értéket összeadunk,  
és a számlálót eggyel megnöveljük.

Végül kiírjuk a két változó hányadosát.

```
#include <stdio.h>
int main() {
    int sum;
    int count;
    int a;
    sum=0;
    count=0;
    while (a) {
        sum=sum+a;
        count++;
        scanf("%d", &a);
    }
    printf("%f\n", (float)sum/count);
}
```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skalár algoritmusok**  
(programozási tételek)

### Értékek átlaga

Felveszünk két segédváltozót.  
Egyet az összeg kiszámolásához.  
*Lebegőpontosat, hogy az osztás majd helyes eredményt adjon!*  
Egyet az értékek számlálásához.

Kezdetben mindkettőt nullára állítjuk.

Minden egyes értéket összeadunk, és a számlálót eggyel megnöveljük.

Végül kiírjuk a két változó hányadosát.

```
#include <stdio.h>
int main() {
    /* DEKLARÁCIÓK */
    int a;
    scanf("%d", &a);
    /* ELŐKÉSZÍTÉS */
    while(a) {
        /* FELDOLGOZÁS */
        scanf("%d", &a);
    }
    /* VÁLASZ */
}
```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skalár algoritmusok**  
(programozási tételek)

### Értékek átlaga

Felveszünk két segédváltozót.  
Egyet az összeg kiszámolásához.  
*Lebegőpontosat, hogy az osztás majd helyes eredményt adjon!*  
Egyet az értékek számlálásához.

Kezdetben mindkettőt nullára állítjuk.

Minden egyes értéket összeadunk, és a számlálót eggyel megnöveljük.

Végül kiírjuk a két változó hányadosát.

```
#include <stdio.h>
int main() {
    double sum;
    int n;
    int a;
    scanf("%d", &a);
    sum=0;
    n=0;
    while(a) {
        sum=sum+a;
        n=n+1;
        scanf("%d", &a);
    }
    printf("%lf\n", sum/n);
}
```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skalár algoritmusok**  
(programozási tételek)

### Leszámlálás

Hány olyan érték van, amely  
eleget tesz egy adott feltételnek?

Például kétjegyű szám-e?

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skalár algoritmusok**  
(programozási tételek)

### Leszámlálás

Felveszünk egy segédváltozót.  
Ebben fogjuk számlálni a helyes  
értékek előfordulását.

Kezdetben nullára állítjuk.

Minden egyes értéket megvizsgálunk,  
és ha helyes, léptetjük a számlálót.

Végül kiírjuk a kapott eredményt.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skalár algoritmusok**  
(programozási tételek)

### Leszámlálás

Felveszünk egy segédváltozót.  
Ebben fogjuk számlálni a helyes  
értékek előfordulását.

Kezdetben nullára állítjuk.

Minden egyes értéket megvizsgálunk,  
és ha helyes, léptetjük a számlálót.

Végül kiírjuk a kapott eredményt.

```
#include <stdio.h>
int main() {
    /* DEKLARÁCIÓK */
    int a;
    scanf("%d", &a);
    /* ELŐKÉSZÍTÉS */
    while(a) {
        /* FELDOLGOZÁS */
        scanf("%d", &a);
    }
    /* VÁLASZ */
}
```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**Elemi vektor-skalár algoritmusok**  
(programozási tételek)

### Leszámlálás

Felveszünk egy segédváltozót.  
Ebben fogjuk számlálni a helyes  
értékek előfordulását.

Kezdetben nullára állítjuk.

Minden egyes értéket megvizsgálunk,  
és ha helyes, léptetjük a számlálót.

Végül kiírjuk a kapott eredményt.

```
#include <stdio.h>
int main() {
    int n;
    int a;
    scanf("%d", &a);
    n=0;
    while(a) {
        if(a>9 && a<100)
            n=n+1;
        scanf("%d", &a);
    }
    printf("%d\n", n);
}
```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Elemi vektor-skalár algoritmusok (programozási tételek)

### Értékek minimuma

Felveszünk egy segédváltozót. Ebben fogjuk tárolni az eddigi legkisebb értéket.

Kezdetben az első ismert adatot jegyezzük meg pillanatnyi minimumként.

Minden egyes értékkel összehasonlítjuk, és ha az új a kisebb, azt jegyezzük meg.

Végül kiírjuk a kapott eredményt.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Elemi vektor-skalár algoritmusok (programozási tételek)

### Értékek minimuma

```
#include <stdio.h>
int main() {
    /* DEKLARÁCIÓK */
    int a;
    scanf("%d", &a);
    /* ELŐKÉSZÍTÉS */
    while(a) {
        /* FELDOLGOZÁS */
        scanf("%d", &a);
    }
    /* VÁLASZ */
}
```

Felveszünk egy segédváltozót. Ebben fogjuk tárolni az eddigi legkisebb értéket.

Kezdetben az első ismert adatot jegyezzük meg pillanatnyi minimumként.

Minden egyes értékkel összehasonlítjuk, és ha az új a kisebb, azt jegyezzük meg.

Végül kiírjuk a kapott eredményt.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Elemi vektor-skalár algoritmusok (programozási tételek)

### Értékek minimuma

```
#include <stdio.h>
int main() {
    int min;
    int a;
    scanf("%d", &a);
    min=a;
    while(a) {
        if (a<min)
            min=a;
        scanf("%d", &a);
    }
    printf("%d\n", min);
}
```

Felveszünk egy segédváltozót. Ebben fogjuk tárolni az eddigi legkisebb értéket.

Kezdetben az első ismert adatot jegyezzük meg pillanatnyi minimumként.

Minden egyes értékkel összehasonlítjuk, és ha az új a kisebb, azt jegyezzük meg.

Végül kiírjuk a kapott eredményt.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Elemi vektor-skalár algoritmusok (programozási tételek)

### Értékek maximuma

Felveszünk egy segédváltozót. Ebben fogjuk tárolni az eddigi legnagyobb értéket.

Kezdetben az első ismert adatot jegyezzük meg pillanatnyi maximumként.

Minden egyes értékkel összehasonlítjuk, és ha az új a nagyobb, azt jegyezzük meg.

Végül kiírjuk a kapott eredményt.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Elemi vektor-skalár algoritmusok (programozási tételek)

### Értékek maximuma

```
#include <stdio.h>
int main() {
    /* DEKLARÁCIÓK */
    int a;
    scanf("%d", &a);
    /* ELŐKÉSZÍTÉS */
    while(a) {
        /* FELDOLGOZÁS */
        scanf("%d", &a);
    }
    /* VÁLASZ */
}
```

Felveszünk egy segédváltozót. Ebben fogjuk tárolni az eddigi legnagyobb értéket.

Kezdetben az első ismert adatot jegyezzük meg pillanatnyi maximumként.

Minden egyes értékkel összehasonlítjuk, és ha az új a nagyobb, azt jegyezzük meg.

Végül kiírjuk a kapott eredményt.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Elemi vektor-skalár algoritmusok (programozási tételek)

### Értékek maximuma

```
#include <stdio.h>
int main() {
    int max;
    int a;
    scanf("%d", &a);
    max=a;
    while(a) {
        if (a>max)
            max=a;
        scanf("%d", &a);
    }
    printf("%d\n", max);
}
```

Felveszünk egy segédváltozót. Ebben fogjuk tárolni az eddigi legnagyobb értéket.

Kezdetben az első ismert adatot jegyezzük meg pillanatnyi maximumként.

Minden egyes értékkel összehasonlítjuk, és ha az új a nagyobb, azt jegyezzük meg.

Végül kiírjuk a kapott eredményt.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Tömb: memóriában tárolt vektor

**Tömb**  
(adatvektor)

- lineáris adatszerkezet
- azonos típusú adatok a memóriában egymás után tárolva
- az elemek elérése indexeléssel lehetséges

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Tömb: memóriában tárolt vektor

Deklarációja:

`<elemtípus> <tömb azonosító>[<elemszám>]`

Például

`double d[8];`  
d azonosítójú, 8 elemű double tömb

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Tömb: memóriában tárolt vektor

Az első tömbelem indexe 0  
Egy N elemű tömb utolsó indexe tehát N-1

Például

`int a[5]` tárolása

|       |       |       |       |       |
|-------|-------|-------|-------|-------|
| $a_0$ | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
|-------|-------|-------|-------|-------|

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## Tömb: memóriában tárolt vektor

Tömbelem kiválasztása (indexelés)

`<tömb azonosító>[<index>]`

Például

$a_2$  hivatkozása `a[2]` alakban történhet

Tömbelemmel mindaz megtehető,  
ami különálló változóként megtehető volna

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## További elemi vektor-skalár vektor-vektor algoritmusok

### Keretprogramot váltunk

Következő példáinknál vagy bemeneti, vagy kimeneti, vagy mindkét irányban szükség lesz az egész vektor egyidejű tárolására, ezért tömböt illetve tömböket fogunk használni.

Formáját tekintve programrészleteket fogunk írni.

A keretprogramban először feltöltjük a tömböt.

A programrészlet eredményét a program meg nem írt utolsó részében használjuk fel.

A példa kedvéért mindenhol 100 elemű tömbbel dolgozunk, amiben int típusú adatokat tárolunk.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



## További elemi vektor-skalár vektor-vektor algoritmusok

### Keretprogramot váltunk

Egy n-elemű tömb bejárására szokásosan for ciklust szervezünk, amelynek magjában végezzük el az adatalem feldolgozását.

`for (i=0; i<n; i++) { /* FELDOLGOZÁS */ }`

Ugyanígy fogunk beolvasni is:

`for (i=0; i<100; i++) scanf("%d", &a[i]);`

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**További elemi vektor-skalár vektor-vektor algoritmusok**

### Keretprogramot váltunk

A feldolgozandó tömb neve legyen **a**

Ha az eredmény egy egész számként értelmezhető, töltsük egy **v** nevű változóba (válasz)

Ha az eredmény egy másik tömb, annak neve legyen **b**

A programrészletet **blokk** formájában írjuk meg, hogy szükség esetén segédváltozókat deklarálhassunk

Az eredeti szerkezetből elmaradhat a „válasz” tevékenység

A programozás alapjai 1 3. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék Budapesti Műszaki és Gazdaságtudományi Egyetem

**További elemi vektor-skalár vektor-vektor algoritmusok**

### Keretprogramot váltunk

```
#include <stdio.h>
int a[100], b[100], n=100, v;
int main() {
    int i; for(i=0; i<100; i++) scanf("%d", &a[i]);
    {
        /* ide kerül a kódrészlet */
    }
    {
        /* itt történik a további feldolgozás */
    }
    return 0;
}
```

A programozás alapjai 1 3. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék Budapesti Műszaki és Gazdaságtudományi Egyetem

**További elemi vektor-skalár vektor-vektor algoritmusok**

### Eldöntés

Igaz-e, hogy egy vektor minden eleme rendelkezik egy adott tulajdonsággal?

Igaz-e, hogy egy vektor egyik eleme sem rendelkezik egy adott tulajdonsággal?

Igaz-e, hogy egy vektornak van olyan eleme, amely rendelkezik egy adott tulajdonsággal?

Igaz-e, hogy egy vektornak van olyan eleme, amely nem rendelkezik egy adott tulajdonsággal?

A programozás alapjai 1 3. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék Budapesti Műszaki és Gazdaságtudományi Egyetem

**További elemi vektor-skalár vektor-vektor algoritmusok**

### Eldöntés (ész nélkül) Mind kétjegyű-e

Felveszünk egy segédváltozót. Ebben fogjuk előállítani a választ.

Kezdetben IGAZ értéket állítunk be.

Ha valamelyik elemre nem teljesül a feltétel, HAMIS értékre váltunk.

A válasz már v-ben van.

A programozás alapjai 1 3. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék Budapesti Műszaki és Gazdaságtudományi Egyetem

**További elemi vektor-skalár vektor-vektor algoritmusok**

### Eldöntés (ész nélkül) Mind kétjegyű-e

Használjuk a v segédváltozót. Ebben fogjuk előállítani a választ.

Kezdetben IGAZ értéket állítunk be.

Ha valamelyik elemre nem teljesül a feltétel, HAMIS értékre váltunk.

A válasz már v-ben van.

```
{
    /* DEKLARÁCIÓK */
    int i;
    /* ELŐKÉSZÍTÉS */
    for (i=0; i<n; i++)
        /* FELDOLGOZÁS */
    /* VÁLASZ */
}
```

A programozás alapjai 1 3. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék Budapesti Műszaki és Gazdaságtudományi Egyetem

**További elemi vektor-skalár vektor-vektor algoritmusok**

### Eldöntés (ész nélkül) Mind kétjegyű-e

Használjuk a v segédváltozót. Ebben fogjuk előállítani a választ.

Kezdetben IGAZ értéket állítunk be.

Ha valamelyik elemre nem teljesül a feltétel, HAMIS értékre váltunk.

A válasz már v-ben van.

```
{
    int i;
    v=1;
    for (i=0; i<n; i++)
        if (a[i]<10 || a[i]>99)
            v=0;
}
```

A programozás alapjai 1 3. előadás © Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék Budapesti Műszaki és Gazdaságtudományi Egyetem





További elemi vektor-skálár  
vektor-vektor algoritmusok

Eldöntés (ésszel)

Mind kétjegyű-e

A v változót csak a végén állítjuk be.

Ha valamelyik elemre nem teljesül a feltétel, azonnal kilépünk a ciklusból.

Ha elérünk a végére, IGAZ értéket állítunk be, egyébként HAMIS

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



További elemi vektor-skálár  
vektor-vektor algoritmusok

Eldöntés (ésszel)

Mind kétjegyű-e

A v változót csak a végén állítjuk be.

Ha valamelyik elemre nem teljesül a feltétel, azonnal kilépünk a ciklusból.

Ha elérünk a végére, IGAZ értéket állítunk be, egyébként HAMIS

```

{
    int i;
    for (i=0; i<n; i++)
        /* FELDOLGOZÁS */
        /* FELTÉTELES KIUGRÁS */
        /* VÁLASZ */
}

```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



További elemi vektor-skálár  
vektor-vektor algoritmusok

Eldöntés (ésszel)

Mind kétjegyű-e

A v változót csak a végén állítjuk be.

Ha valamelyik elemre nem teljesül a feltétel, azonnal kilépünk a ciklusból.

Ha elérünk a végére, IGAZ értéket állítunk be, egyébként HAMIS

```

{
    int i;
    for (i=0; i<n; i++)
        if (a[i]<10 || a[i]>99)
            break;
    v=(i==n);
}

```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



További elemi vektor-skálár  
vektor-vektor algoritmusok

Leválogatás

Gyűjtsük egy másik vektorba azokat az elemeket, amelyek rendelkeznek egy adott tulajdonsággal!

Függvényértékként adjuk vissza, hogy hány elemet válogattunk ki, és tettünk a másik vektorba!

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



További elemi vektor-skálár  
vektor-vektor algoritmusok

Leválogatás

A kétjegyűeket

Felvesszünk egy segédváltozót. Ez mutatja a következő szabad helyet, egyúttal az elemszámot.

Kezdetben nullára állítjuk.

Ha valamelyik elemre teljesül a feltétel, beírjuk a következő helyre a másik vektorba, és léptetjük a számlálót.

A válasz a leválogatott elemek száma.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem



További elemi vektor-skálár  
vektor-vektor algoritmusok

Leválogatás

A kétjegyűeket

Felvesszünk egy segédváltozót. Ez mutatja a következő szabad helyet, egyúttal az elemszámot.

Kezdetben nullára állítjuk.

Ha valamelyik elemre teljesül a feltétel, beírjuk a következő helyre a másik vektorba, és léptetjük a számlálót.

A válasz a leválogatott elemek száma.

```

{
    /* DEKLARÁCIÓK */
    int i;
    /* ELŐKÉSZÍTÉS */
    for (i=0; i<n; i++)
        /* FELDOLGOZÁS */
        /* VÁLASZ */
}

```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**További elemi vektor-skálár vektor-vektor algoritmusok**

### Leválogatás

A kétjegyűeket

Felvesszünk egy segédváltozót. Ez mutatja a következő szabad helyet, egyúttal az elemszámot.

Kezdetben nullára állítjuk.

Ha valamelyik elemre teljesül a feltétel, beírjuk a következő helyre a másik vektorba, és léptetjük a számlálót.

A válasz a leválogatott elemek száma.

```

{
    int j;
    int i;
    j=0;
    for (i=0; i<n; i++)
        if (a[i]>9 && a[i]<100) {
            b[j]=a[i];
            j=j+1;
        }
}

```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**További elemi vektor-skálár vektor-vektor algoritmusok**

### Helyben szétválogatás

Gyűjtsük a vektor elejére azokat az elemeket, amelyek rendelkeznek egy adott tulajdonsággal!

A tulajdonsággal nem rendelkező elemek kerüljenek a vektor végére!

A csoportokon belül a sorrend nem számít.

Adjuk vissza az első csoport elemszámát!

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**További elemi vektor-skálár vektor-vektor algoritmusok**

### Helyben szétválogatás

Kétjegyűeket előre

Felvesszünk egy segédváltozót. Ez mutatja, hogy hol kezdődnek a „rossz” elemek a vektor végén.

Kezdetben n-re állítjuk.

*Túl az adatokon, hiszen még egyet sem találtunk!*

Ha valamelyik elemre teljesül a feltétel, átlépjük, egyébként csökkentjük a számlálót, és az elemet hátra cseréljük.

A válasz az első csoport elemszáma.

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**További elemi vektor-skálár vektor-vektor algoritmusok**

### Helyben szétválogatás

Kétjegyűeket előre

Felvesszünk egy segédváltozót. Ez mutatja, hogy hol kezdődnek a „rossz” elemek a vektor végén.

Kezdetben n-re állítjuk.

*Túl az adatokon, hiszen még egyet sem találtunk!*

Ha valamelyik elemre teljesül a feltétel, átlépjük, egyébként csökkentjük a számlálót, és az elemet hátra cseréljük.

A válasz az első csoport elemszáma.

```

{
    /* DEKLARÁCIÓK */
    int i;
    /* ELŐKÉSZÍTÉS */
    for (i=0; i<j;)

    /* FELDOLGOZÁS */

    /* VÁLASZ */
}

```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem

**További elemi vektor-skálár vektor-vektor algoritmusok**

### Helyben szétválogatás

Kétjegyűeket előre

Felvesszünk egy segédváltozót. Ez mutatja, hogy hol kezdődnek a „rossz” elemek a vektor végén.

Kezdetben n-re állítjuk.

*Túl az adatokon, hiszen még egyet sem találtunk!*

Ha valamelyik elemre teljesül a feltétel, átlépjük, egyébként csökkentjük a számlálót, és az elemet hátra cseréljük.

A válasz az első csoport elemszáma.

```

{
    int j;
    int i;
    j=n;
    for (i=0; i<j;)
        if (a[i]>9 && a[i]<100)
            i++;
        else {
            int k=a[i]; j--;
            a[i]=a[j]; a[j]=k;
        }
}

```

A programozás alapjai 1  
3. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem