

# INFORMATIKA LABORATÓRIUM I.

Programozási tételek

# Összegzés tétele

Adott egy  $N$  elemű sorozat, számoljuk ki a sorozat elemeinek összegét. A sorozatot egy  $A$  nevű vektorban tároljuk el!

**PROGRAM** osszegzes

$S := 0$

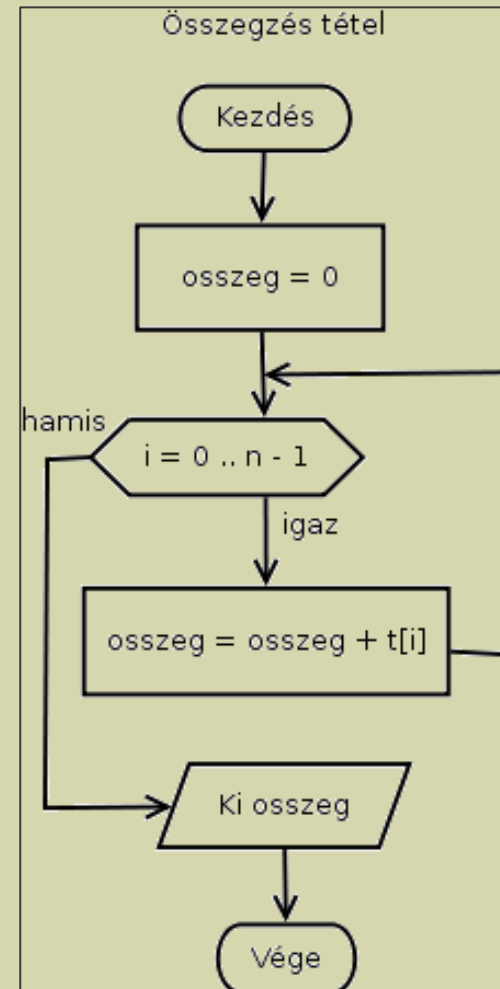
**CIKLUS**  $I := 1$ -TŐL  $N$ -IG  $1$ -VEL

$S := S + a[i]$

**CIKLUS VÉGE**

Ki:  $s$

**PROGRAM VÉGE**



# Összegzés tétele

```
//Összegzés tétele
#include <stdio.h>

main()
{
    int tomb[5] = {8, 9, 3, 5, 4};
    int n = 5;
    printf("Összegzés\n");
    int osszeg = 0;

    for (int i = 0; i < 5; i++)
        osszeg = osszeg + tomb[i];

    printf("%d\n", osszeg);
}
```

# Megszámlálás tétele

Adott egy  $N$  elemű sorozat. Számoljuk meg, a sorozat hány elemén értelmezhető a  $T$  tulajdonság!

PROGRAM megszamlal

$s := 0$

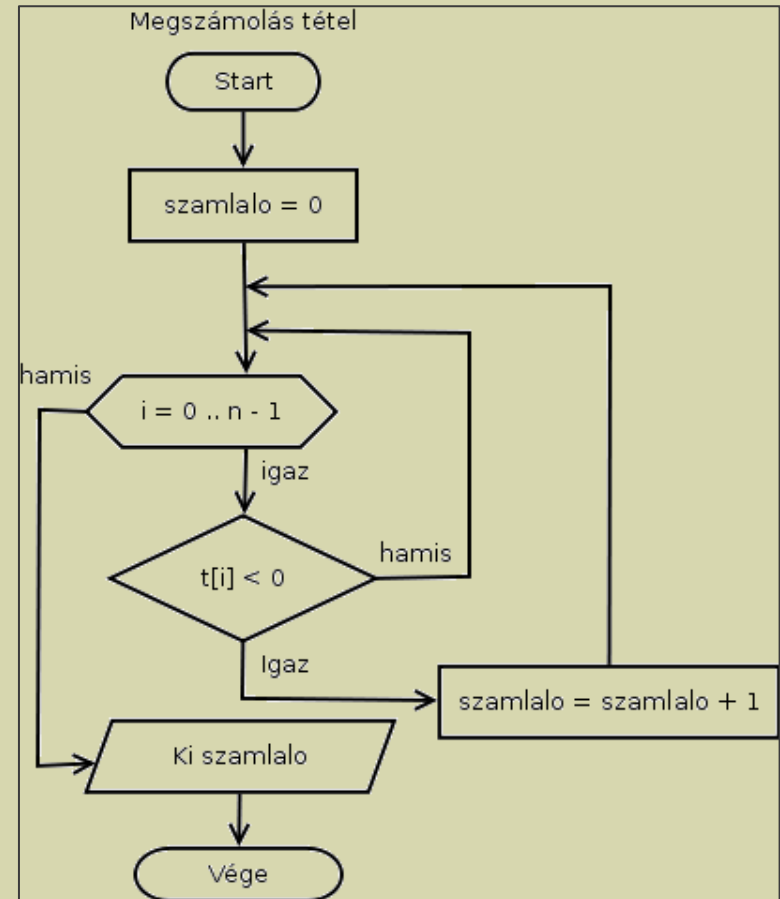
CIKLUS  $i := 1$ -TŐL  $N$ -IG 1-VEL

HA  $a[i]$   $T$  tulajdonságú AKKOR  $s := s + 1$

ELÁGAZÁS VÉGE

KI: 'Darabszám: ',  $s$

PROGRAM VÉGE



# Megszámlálás tétele

```
//megszámlálás
#include <stdio.h>

main()
{
    int tomb[] = {8, 9, 3, 5, 4, 2, 7};

    int szamol=0;
    int i;
    for(i=0; i<7;i++)
        if(tomb[i] < 5)
            szamol++;
    printf("5-nél kisebb számok: %d\n", szamol);
}
```

# Eldöntés tétele

Adott egy  $N$  elemű sorozat és egy a sorozat elemein értelmezett  $T$  tulajdonság. Határozzuk meg, hogy van-e a sorozatban legalább egy  $T$  tulajdonsággal rendelkező elem!

**PROGRAM** eldontes

$i := 1$

**CIKLUS AMÍG**  $i \leq n$  ÉS  $a[i]$  nem  $T$  tulajdonságú

$i := i + 1$

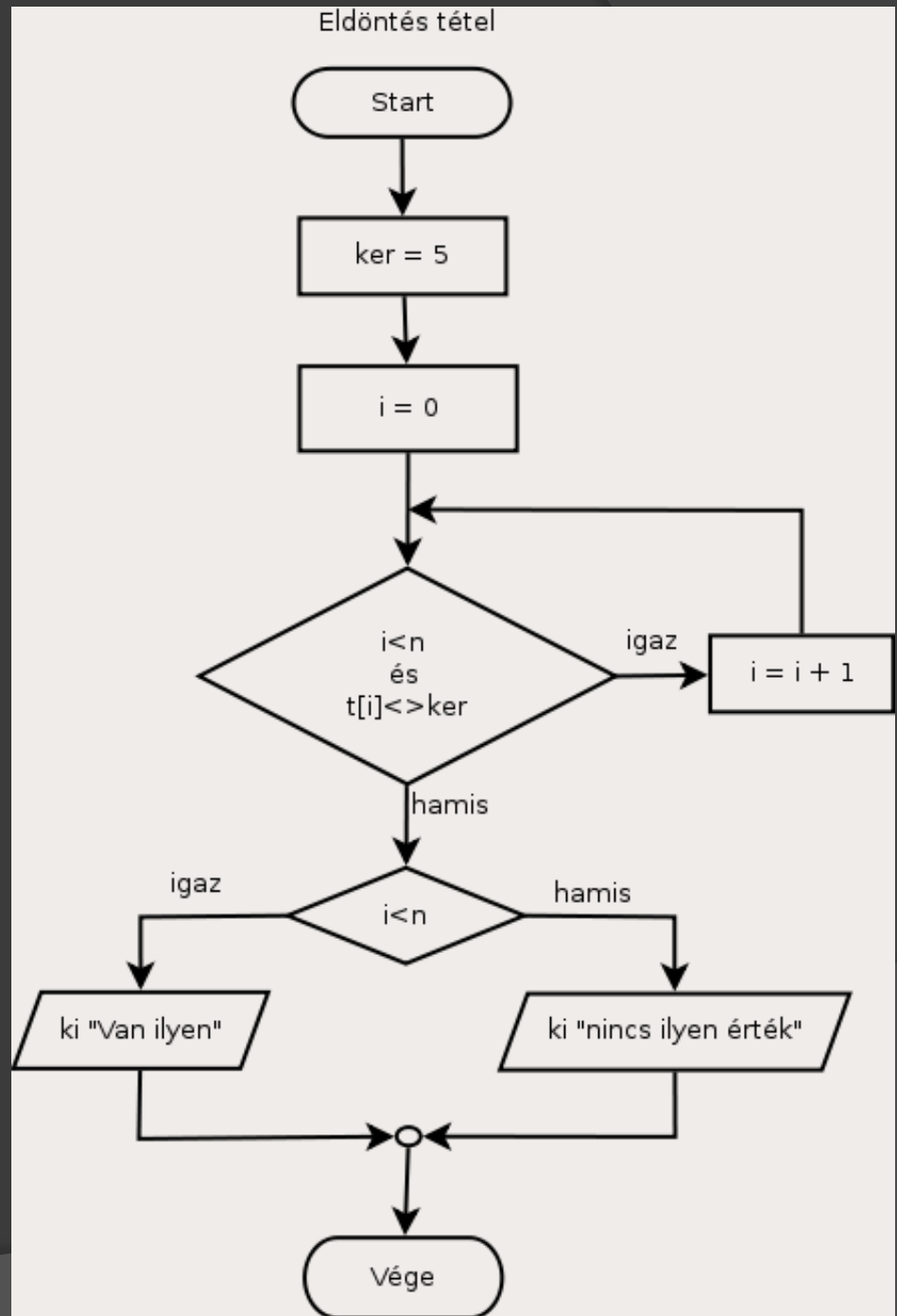
**CIKLUS VÉGE**

**HA**  $i \leq N$  **AKKOR** Ki: 'Volt ilyen elem!'

**KÜLÖNBEN** Ki: 'Nem volt ilyen elem!'

**PROGRAM VÉGE**

# Eldöntés tétele



# Eldöntés tétele

```
//eldöntés
#include <stdio.h>

main()
{
    int tomb[] = {8, 9, 3, 5, 4, 2, 7};
    int n=7;
    int ker=5;
    int i=0;
    while(i<n && tomb[i] != ker)
        i++;
    if(i<n)
        printf("Van ilyen\n");
    else
        printf("Nem található\n");
}
```



# Kiválasztás tétele

Adott egy  $N$  elemű sorozat és a sorozat elemein értelmezett  $T$  tulajdonság. A sorozat legalább 1  $T$  tulajdonságú elemmel rendelkezik. Határozzuk meg az első ilyen  $T$  tulajdonságú elem sorszámát!

**PROGRAM** kiválasztás

**CIKLUS AMÍG**  $a[i]$  nem  $T$  tulajdonságú

$i := i + 1$

**CIKLUS VÉGE**

    Ki:  $i$ , '. elem volt!'

**PROGRAM VÉGE**

# Kiválasztás tétele

```
//kiválasztás
#include <stdio.h>

main()
{
    int tomb[] = {8, 9, 3, 5, 4, 2, 7};
    int n=7;
    int ker=5;
    int i=0;
    while(i<n && tomb[i] != ker)
        i++;

    printf("A keresett elem helye: %d\n", i);
}
```

# Lineáris keresés tétele

Adott egy  $N$  elemű sorozat és a sorozat elemein értelmezett  $T$  tulajdonság. Határozzuk meg, hogy van-e  $T$  tulajdonságú elem, és ha van, akkor adjuk meg annak a sorszámát!

**PROGRAM** linker

$i := 1$

**CIKLUS AMÍG**  $i \leq N$  és  $a[i]$  nem  $T$  tulajdonságú

$i := i + 1$

**CIKLUS VÉGE**

$\text{van} := i \leq n$

**HA** van **AKKOR**  $\text{sorsz} := i$

**ELÁGAZÁS VÉGE**

**PROGRAM VÉGE**

# Lineáris keresés tétele

```
//lineáris keresés
#include <stdio.h>

main()
{
    int tomb[] = {8, 9, 3, 5, 4, 2, 7};
    int n=7;
    int ker=5;
    int i=0;
    while(i<n && tomb[i] != ker)
        i++;
    if(i<n)
    {
        printf("Van ilyen\n");
        printf("A helye: %d\n", i);
    }
    else
        printf("Nem található\n");
}
```

# Kiválogatás tétele

Adott egy  $N$  elemű sorozat és az ezen a sorozaton értelmezett  $T$  tulajdonság. Határozzuk meg a sorozat összes  $T$  tulajdonsággal rendelkező elemét! A sorozatból kiválasztott elemeket egy  $B$  nevű tömbben helyezzük el!

**PROGRAM** kivalogatas

$j:=0$

**CIKLUS**  $i:=1$ -TŐL  $N$ -IG 1-VEL

**HA**  $a[i]$   $T$  tulajdonságú **AKKOR**  $j:=j+1$

$b[j]:=a[i]$

**ELÁGAZÁS VÉGE**

**CIKLUS VÉGE**

**PROGRAM VÉGE**

# Kiválogatás tétele

```
//kiválogatás tétele
#include <stdio.h>

main()
{
    int tomb[] = {8, 9, 3, 5, 4, 2, 7};
    int b[7];

    int i, n=7;
    int j=0;
    for(i=0; i<n; i++)
        if(tomb[i] < 5)
        {
            b[j] = tomb[i];
            j++;
        }
    int m = j;
    for(i=0; i<m; i++)
        printf("%d ", b[i]);
    printf("\n");
}
```

# Széválogatás tétele

Adott egy  $N$  elemű sorozat. Ennek egy kiválasztott eleme  $x$ . Adjuk meg azt az algoritmust, amelynek eredménye az lesz, hogy az  $x$  előtt a nála kisebb elemek, utána pedig a nála nagyobb elemek lesznek!

**PROGRAM** szetvalogatas

$x := a[1]$

$E := 1$

$U := N$

**CIKLUS AMÍG**  $e < u$

**CIKLUS AMÍG**  $(E < U) \text{ ÉS } (a[U] \geq X)$

$U := U - 1$

**CIKLUS VÉGE**

**HA**  $(E < U)$  **AKKOR**  $a[E] := a[U]$

$E := E + 1$

**ELÁGAZÁS VÉGE**

**CIKLUS AMÍG**  $(E < U) \text{ ÉS } (a[E] \leq X)$

$E := E + 1$

**CIKLUS VÉGE**

**HA**  $(E < U)$  **AKKOR**  $a[U] := a[E]$

$U := U - 1$

**ELÁGAZÁS VÉGE**

**CIKLUS VÉGE**

$a[E] := X$

**PROGRAM VÉGE**

# Széválogatás tétele

```
#include <stdio.h>
main()
{
    int tomb[] = {8, 9, 3, 5, 4, 2, 7};
    int b[7], c[7];

    int i, n=7;
    int j=0, k=0;
    for(i=0; i<n; i++)
        if(tomb[i] < 5)
        {
            b[j] = tomb[i];
            j++;
        }
        else
        {
            c[k] = tomb[i];
            k++;
        }
    int m = j, l = k;
    for(i=0; i<m; i++)
        printf("%d ", b[i]);
    printf("\n");
    for(i=0; i<l; i++)
        printf("%d ", c[i]);
    printf("\n");
}
```



# Rendezések: buborék

Adott egy  $N$  elemű sorozat, rendezzük a sorozat elemeit növekvő majd csökkenő sorrendbe!

**PROGRAM** bubrend\_novekvo

**CIKLUS**

$i := 2$ -TŐL  $N$ -IG 1-VEL

**CIKLUS**

$j := N$ -TŐL  $i$ -IG  $-1$ -VEL

**HA**  $a[j-1] > a[j]$  **AKKOR**  $x := j$

$j := j - 1$

$j-1 := x$

**ELÁGAZÁS VÉGE**

**CIKLUS VÉGE**

**CIKLUS VÉGE**

**PROGRAM VÉGE**

# Rendezések: buborék

```
/* Buborék rendezés */
#include <stdio.h>

int main()
{
    int tomb[] = {3, 8, 2, 4, 5, 1, 6};
    int n = 7; // A tömb elemeinek száma
    int i, j, tmp;

    for(i= n-1; i>0; i--)
        for(j=0; j<i; j++)
            if(tomb[j] > tomb[j+1])
            {
                tmp = tomb[j];
                tomb[j] = tomb[j+1];
                tomb[j+1] = tmp;
            }

    for(i=0; i<n; i++)
        printf("%d ", tomb[i]);
    printf("\n");
}
```

# Rendezések: buborék

```
/* Buborék rendezés, ha a cserét figyeljük */
#include <stdio.h>

int main()
{
    int tomb[] = {3, 8, 2, 4, 5, 1, 6};
    int n = 7; // A tömb elemeinek száma
    int i, j, tmp, voltcsere=1;

    for(i= n-1; i>0 && voltcsere; i--)
    {
        voltcsere = 0;
        for(j=0; j<i; j++)
            if(tomb[j] > tomb[j+1])
            {
                tmp = tomb[j];
                tomb[j] = tomb[j+1];
                tomb[j+1] = tmp;
                voltcsere = 1;
            }
    }

    for(i=0; i<n; i++)
        printf("%d ", tomb[i]);
    printf("\n");
}
```

# Rendezések: buborék

```
/* Buborék rendezés, ha az utolsó csere helyét is figyeljük */
#include <stdio.h>

int main()
{
    int tomb[] = {3, 8, 2, 4, 5, 1, 6};
    int n = 7; // A tömb elemeinek száma
    int i, j, tmp, utolsocsere;

    for(i= n-1; i>0 ; i = utolsocsere)
    {
        utolsocsere = 0;
        for(j=0; j<i; j++)
            if(tomb[j] > tomb[j+1])
            {
                tmp = tomb[j];
                tomb[j] = tomb[j+1];
                tomb[j+1] = tmp;
                utolsocsere = j;
            }
    }

    for(i=0; i<n; i++)
        printf("%d ", tomb[i]);
    printf("\n");
}
```

# Gyakorlás

Programozzuk le az alábbi tételt!

## MAXIMUM-KIVÁLASZTÁS TÉTELE

Adott egy  $N$  elemű sorozat, határozzuk meg a sorozat legnagyobb elemét és annak sorszámát!

**PROGRAM** maxkiv

max:=a[1]

maxsorsz:=1

**CIKLUS** i:=2-TŐL N-IG 1-VEL

HA  $a[i] > \text{max}$  **AKKOR** max:=a[i]

maxsorsz:=i

**ELÁGAZÁS VÉGE**

**CIKLUS VÉGE**

KI: max,maxsorsz

**PROGRAM VÉGE**

***Megjegyzés:** A program több maximális érték esetén csak az első legnagyobb elem sorszámát jegyzi meg!*

# Gyakorlás

Programozzuk le az alábbi tételt!

## INDEXVEKTOROS RENDEZÉS A MINIMUM-KIVÁLASZTÁS MÓDSZERÉVEL

Adott egy  $N$  elemű sorozat. Rendezzük az elemeket úgy, hogy nem a sorozat elemeit cserélgetjük, hanem a sorozat elemeinek indexét helyezzük el megfelelő sorrendben egy másik tömbben (indexvektorban)!

**PROGRAM** indexvektoros\_rendezes

$j:=1$ ;

$\text{min}:=1$ ;

$x:=-\infty$

**CIKLUS AMÍG**  $j \leq n$

$\text{min}:=+\infty$

**CIKLUS**  $i:=1$ -TŐL  $N$ -IG 1-VEL

**HA**  $(a[i] < \text{min})$  ÉS  $(a[i] > x)$  **AKKOR**  $\text{min}:=a[i]$

**ELÁGAZÁS VÉGE**

**CIKLUS VÉGE**

**CIKLUS**  $k:=1$ -TŐL  $N$ -IG 1-VEL

**HA**  $a[k]=\text{min}$  **AKKOR**  $b[j]:=k$

$x:=a[b[j]]$

$j:=j+1$

**ELÁGAZÁS VÉGE**

**CIKLUS VÉGE**

**CIKLUS VÉGE**

**PROGRAM VÉGE**