

9. Típusalgoritmusok (programozási tételek)

Összegezés (sorozatszámítás)

Feladat: Adott az n elemű v vektor. Határozzuk meg a vektor elemeinek összegét!

Algoritmus:

```
s := 0
ciklus i := 0-tól (n-1)-ig
    s := s + v[i]
ciklus vége
ki: s
```

Program:

```
int s = 0;
for (int i = 0; i < n; i++)
    s = s + v[i];
Console.WriteLine("A vektor elemeinek összege= {0}", s);
```

Megszámolás

Feladat: Adott az n elemű v vektor. Számoljuk meg, hány adott tulajdonságú eleme van!

Algoritmus:

```
db := 0
ciklus i := 0-tól (n-1)-ig
    ha v[i] adott tulajdonságú akkor
        db := db + 1
    elágazás vége
ciklus vége
ki: db
```

Program:

```
int db = 0;
for (int i = 0; i < n; i++)
    if (v[i] > 0)
        db++;
Console.WriteLine("A vektornak {0} db pozitív eleme van.", db);
```

Maximum-kiválasztás

Feladat: Adott az n elemű v vektor. Határozzuk meg a legnagyobb elemét, és egy indexet, ahol azt fel is veszi!

Algoritmus:

```
max := v[0]
maxh := 0
ciklus i := 1-től (n-1)-ig
    ha v[i] > max akkor
        max := v[i]
        maxh := i
    elágazás vége
ciklus vége
ki: max, maxh
```

Program:

```
int max = v[0];
int maxh = 0;
for (int i=1; i<n; i++)
    if (v[i] > max)
    {
        max = v[i];
        maxh = i;
    }
Console.WriteLine("A vektor maximuma {0}, amit az {1}. helyen felvesz.", max, maxh);
```

Keresés (lineáris keresés)

Feladat: Adott az n elemű v vektor. Döntsük el, hogy van-e adott tulajdonságú eleme, és ha van, írjuk egy ilyen elem indexét!

Algoritmus:

```
index := 0
ciklus amíg (index < n) és (v[index] nem adott tulajdonságú)
    index := index + 1
ciklus vége
ha index<n akkor
    ki: „Van ilyen:”, index
egyébként
    ki: „Nincs ilyen”
```

Program:

```
int index = 0;
while (index < n && v[index] <= 0)
    index++;
if (index < n)
    Console.WriteLine("Van pozitív elem, például a {0}.", index);
else
    Console.WriteLine("Nincs ilyen elem");
```

Rendezés

Feladat: Adott az n elemű v vektor! Rendezzük növekvő sorba az elemeit!

Rendezés egyszerű cserével

Algoritmus:

```
ciklus i := 0-tól (n-1)-ig
    ciklus j := 0-tól (n-i-2)-ig
        ha v[j] > v[j+1] akkor
            b := v[j]
            v[j] := v[j+1]
            v[j+1] := b
        elágazás vége
    ciklus vége
ciklus vége
```

Program:

```
static void Buborék()
{
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n - i - 1; j++)
            if (v[j] > v[j + 1])
            {
                int b = v[j];
                v[j] = v[j + 1];
                v[j + 1] = b;
            }
}
```

Rendezés maximum-kiválasztással

Algoritmus:

```
ciklus i := (n-1)-től 0-ig
    max := v[0]
    maxh := 0
    ciklus j := 0-tól i-ig
        Ha v[j] > max akkor
            max := v[j]
            maxh := j
        elágazás vége
    ciklus vége
    v[maxh] := v[i]
    v[i] := max
ciklus vége
```

Program:

```
static void Maxrend()
{
    for (int i = n - 1; i >= 0; i--)
    {
        int max = v[0];
        int maxh = 0;
        for (int j = 0; j <= i; j++)
            if (v[j] > max)
            {
                max = v[j];
                maxh = j;
            }
        v[maxh] = v[i];
        v[i] = max;
    }
}
```