

Flow Academy Mentors / ... / Programozási alapok 

Programozási tételek

**Zsolt Tasnádi**

Utoljára módosított pillanatokkal ezelőtt

Felelős: @Zsolt Tasnádi

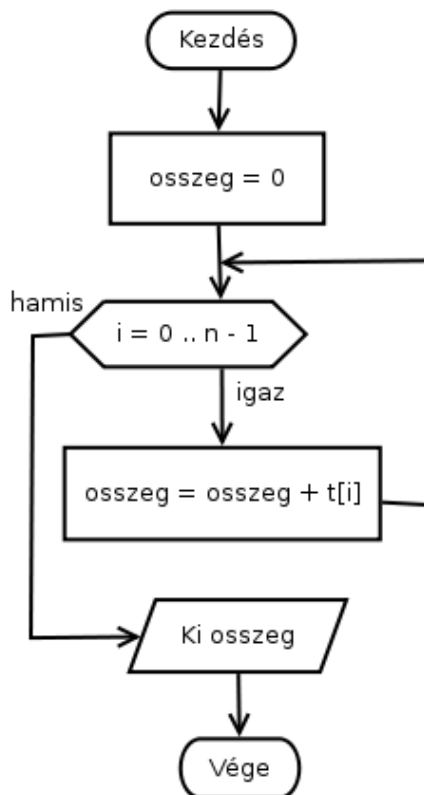
Elméleti órák: 10x1.5

Gyakorlati órák (napközi): 10 nap (4 óra /nap) (Feladatgyűjtemények oldal)

- Összegzés
- Megszámlálás
- Eldöntés
- Kiválasztás
- Keresés
- Másolás
- Kiválogatás
- Szétválogatás
- Metszet
- Unió
- Minimum / maximum érték keresése
- Rendezési algoritmusok
 - Minimum-kiválasztásos rendezés
 - Beszúrásos rendezés
 - Buborékos rendezés (szomszédos elemek cseréje)
 - Cserélő rendezés
- Kapcsolódó anyagok

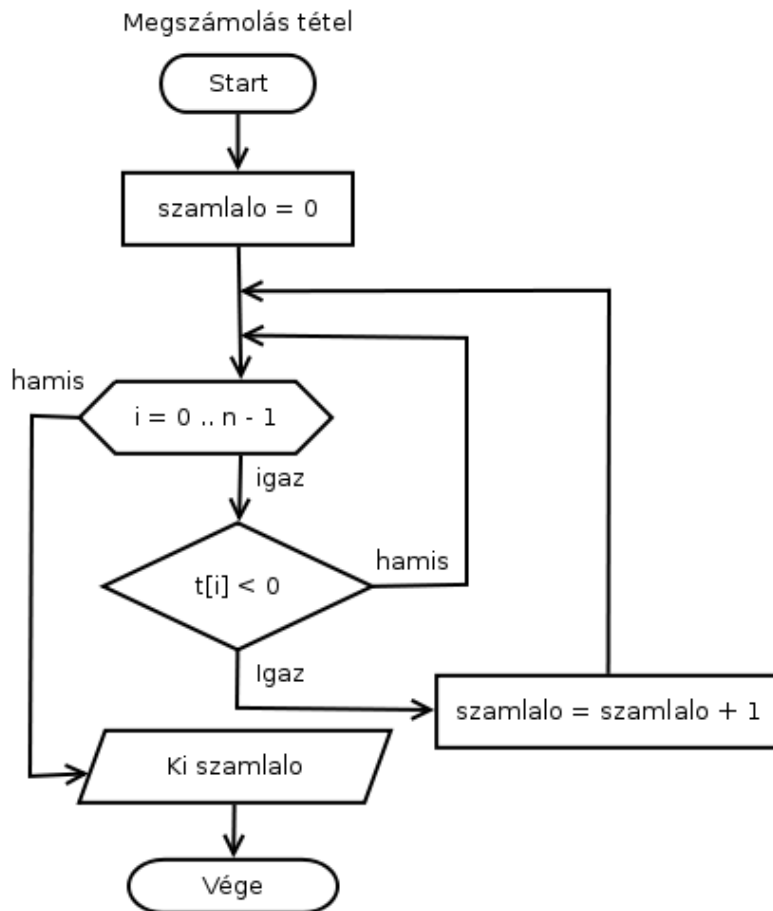
Összegzés

Összegzés tétel



```
1 let összeg = 0;  
2 let tomb = [1, 2, 3, 4, 5, 6, 7, 8, 9];  
3 for(let i = 0; i < tomb.length; i++) {  
4     összeg += tomb[i];  
5 }  
6 console.log(összeg);
```

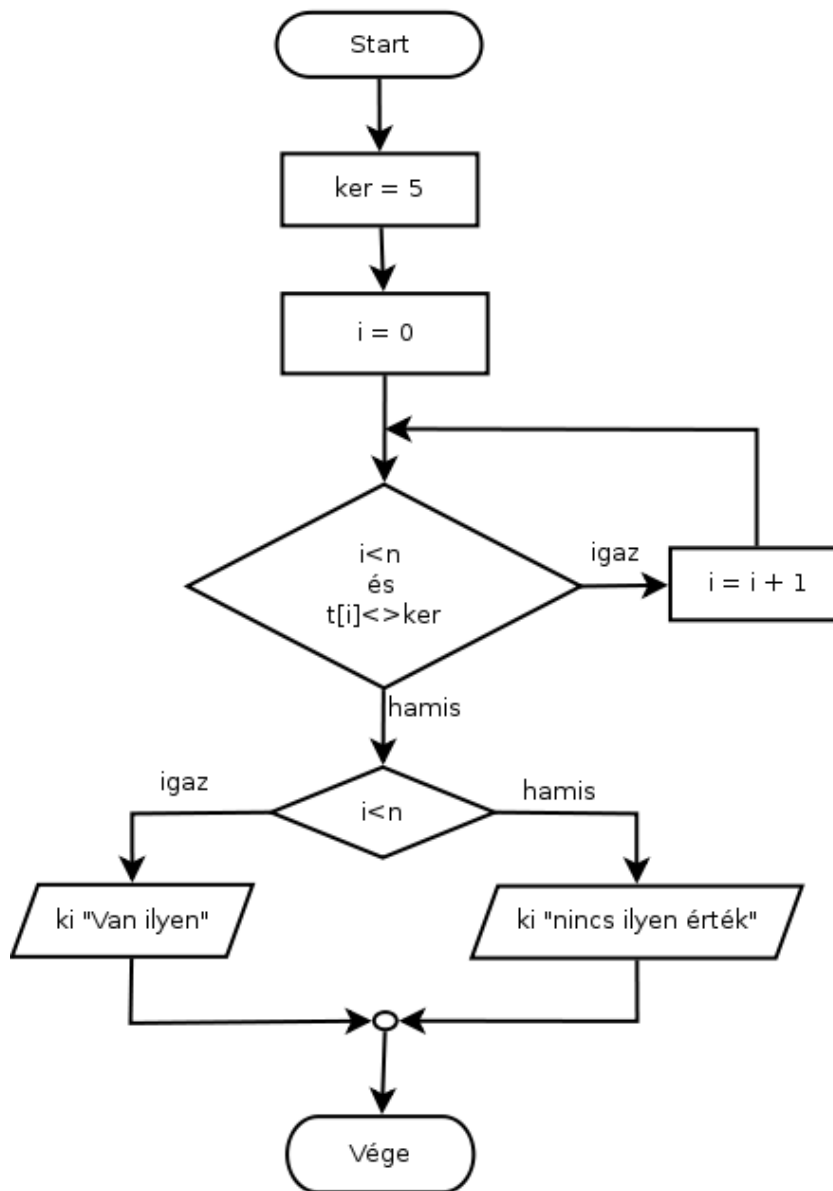
Megszámlálás



```
1 let szamlalo = 0;
2 let tomb = [1, 2, 3, 4, 5, 6, 7, 8, 9];
3 for(let i = 0; i < tomb.length; i++) {
4     if(tomb[i] > 3) {
5         szamlalo += 1;
6     }
7 }
8 console.log(szamlalo);
```

Eldöntés

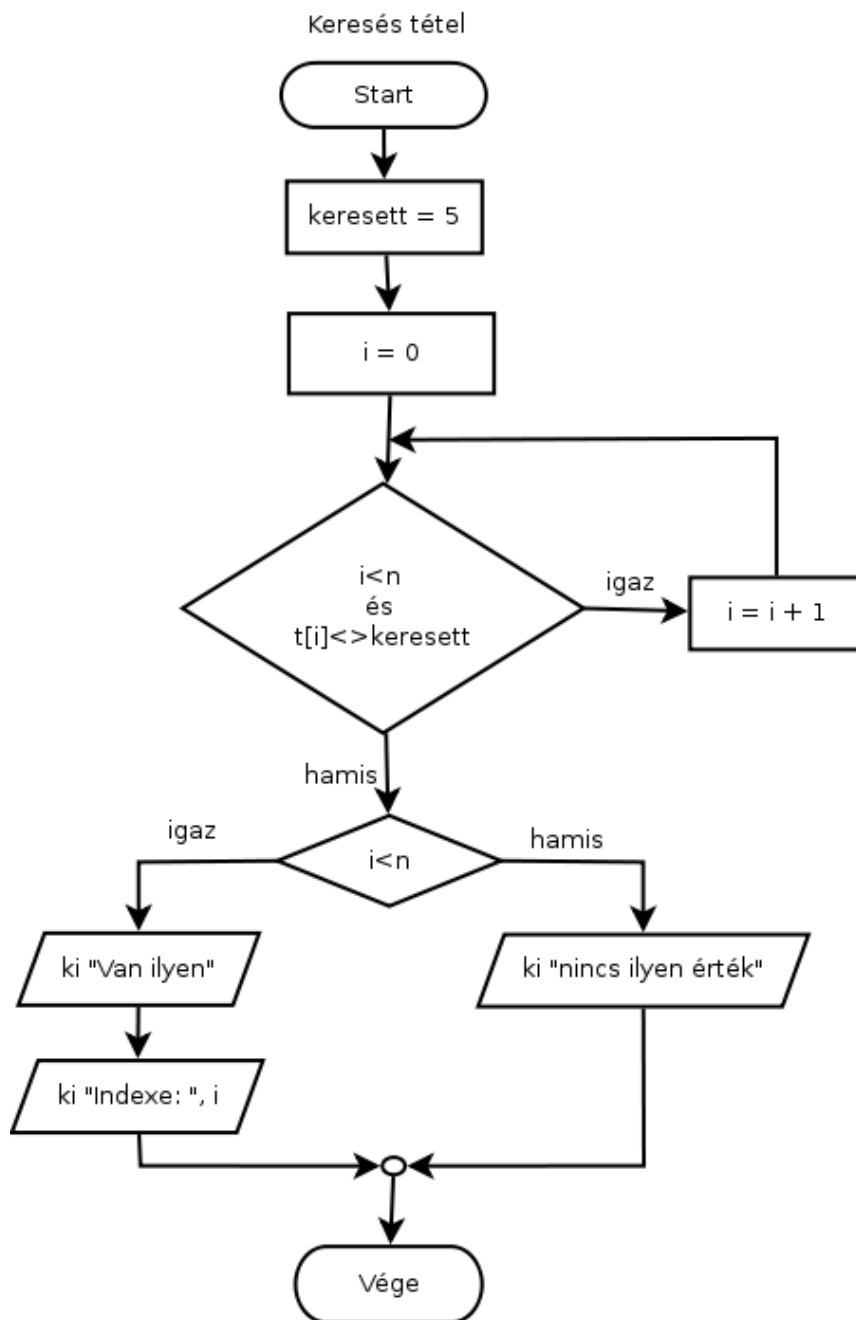
Eldöntés tétel



```

1  let van = false;
2  let tomb = [1, 2, 3, 4, 5, 6, 7, 8, 9];
3  for(let i = 0; i < tomb.length; i++) {
4      if(tomb[i] > 3) {
5          van = true;
6      }
7  }
8  console.log(van);
  
```

Kiválasztás

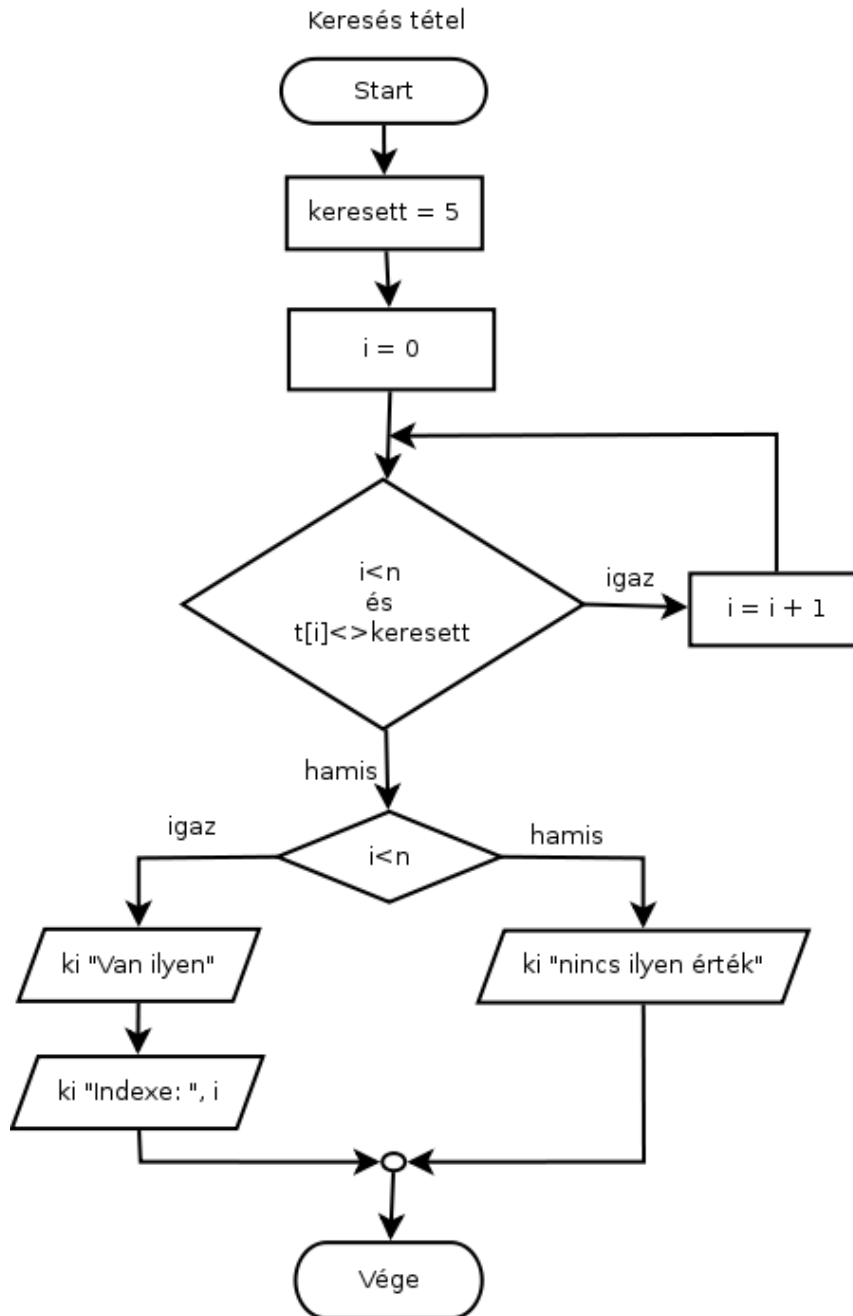


```

1  let i = 0;
2  let keresett = 7;
3  let tomb = [1, 2, 3, 4, 5, 6, 7, 8, 9];
4  while(i < tomb.length && tomb[i] != keresett) {
5      i += 1;
6  }
7  if(i < n) {
8      console.log("van ilyen, a ", i+1, ". elem");
9  } else {
10     console.log("nincs ilyen");
11 }

```

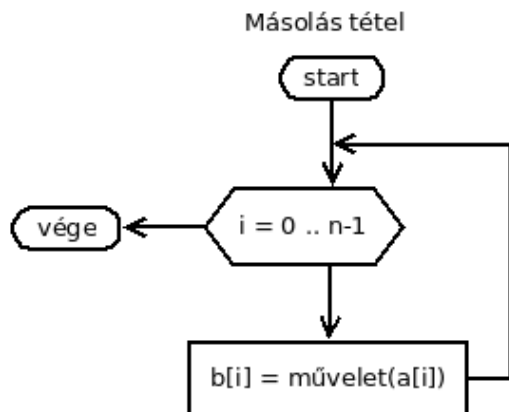
Keresés



```

1  let i = 0;
2  let keresett = 7;
3  let tomb = [1, 2, 3, 4, 5, 6, 7, 8, 9];
4  while(i < tomb.length && tomb[i] != keresett) {
5      i += 1;
6  }
7  if(i < n) {
8      console.log("van ilyen, a ", i+1, ". elem");
9  } else {
10     console.log("nincs ilyen");
11 }
  
```

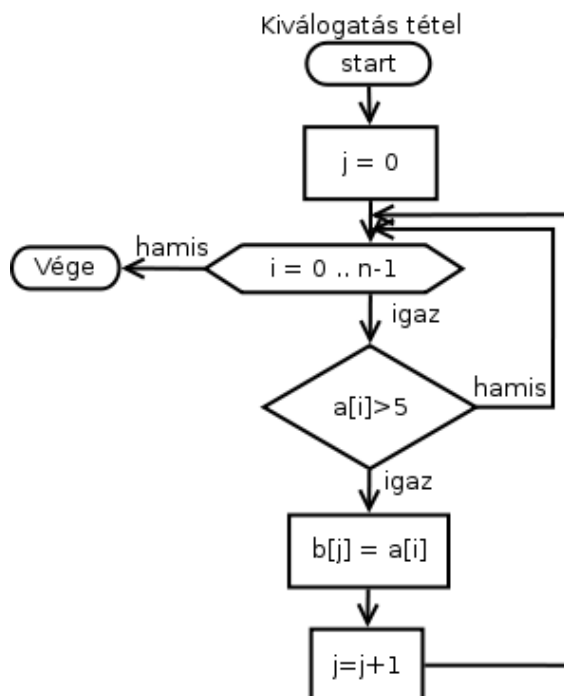
Másolás



```

1  let i;
2  let innenTomb = [1, 2, 3, 4, 5, 6, 7, 8, 9];
3  let ideTomb = [];
4  for(i = 0; i < innenTomb.length; i++) {
5    ideTomb[i] = innenTomb * 2;
6  }
7  console.log(ideTomb);
  
```

Kiválogatás



```

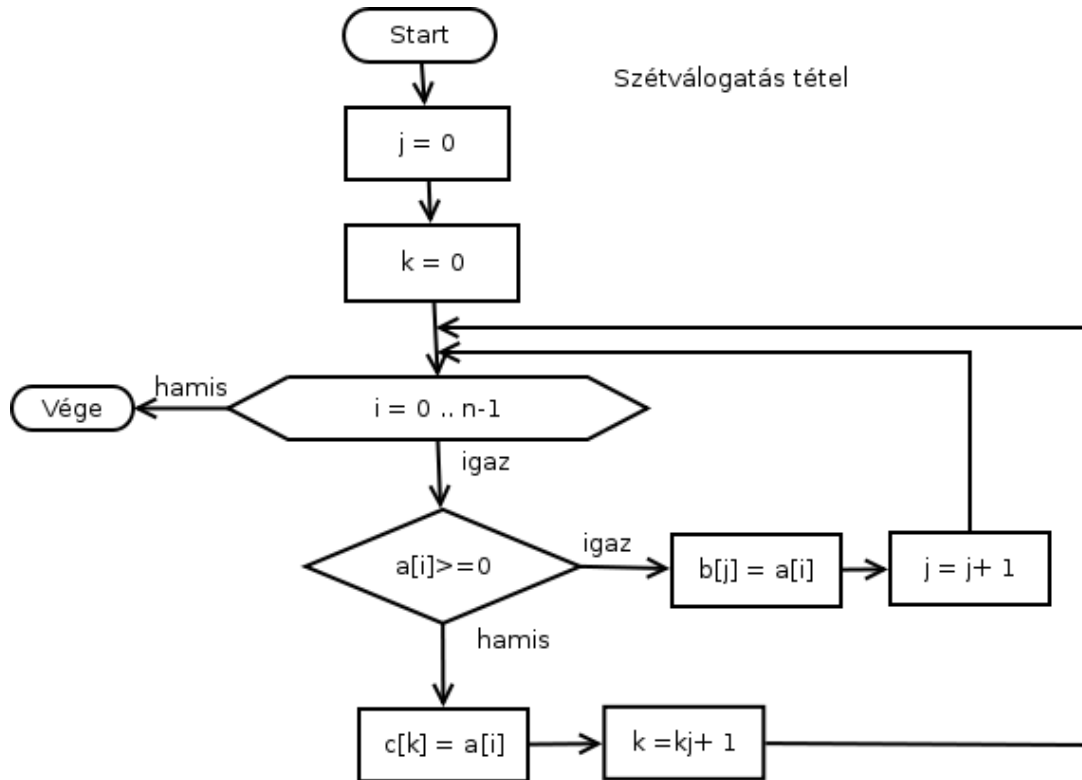
1  let i = 0;
2  let j = 0;
3  let innenTomb = [1, 2, 3, 4, 5, 6, 7, 8, 9];
4  let ideTomb = [];
5  for(i; i < innenTomb.length; i++) {
6    if((innenTomb[i] % 2) == 0) {
7      ideTomb[j] = innenTomb[i];
8    }
  
```

```

9         j += 1;
10    }
11 }
    console.log(ideTomb);

```

Szétválogatás



```

1  let i = 0;
2  let j = 0;
3  let k = 0;
4  let innenTomb = [1, 2, 3, 4, 5, 6, 7, 8, 9];
5  let ideTomb1 = [];
6  let ideTomb2 = [];
7  for(i; i < innenTomb.length; i++) {
8      if((innenTomb[i] % 2) == 0) {
9          ideTomb1[j] = innenTomb[i];
10         j += 1;
11     } else {
12         ideTomb2[k] = innenTomb[i];
13         k += 1;
14     }
15 }
16 console.log(ideTomb1);
17 console.log(ideTomb2);

```

Metszet


```

graph TD
    Start([Start]) --> K0[k = 0]
    K0 --> ILoop((i = 0 .. n - 1))
    ILoop -- hamis --> Vége([vége])
    ILoop -- igaz --> J0[j = 0]
    J0 --> JCond{j < m és  
b[j] <> a[i]}
    JCond -- igaz --> JInc[j = j + 1]
    JInc --> JCond
    JCond -- hamis --> JCond2{j < m}
    JCond2 -- hamis --> ILoop
    JCond2 -- igaz --> C[k = a[i]]
    C --> KInc[k = k + 1]
    KInc --> ILoop

```

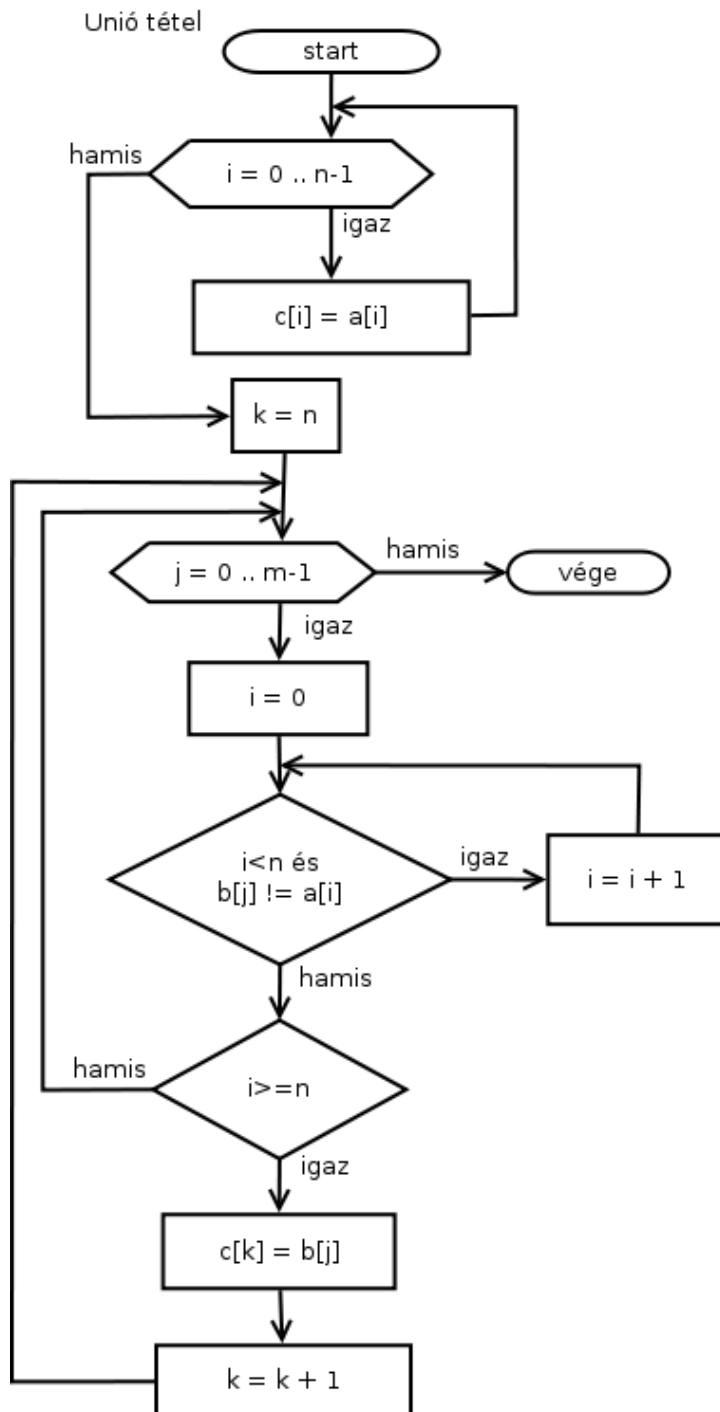
<https://priceless.atlassian.net/wiki/spaces/FM/pages/5848527/Programoz+si+t+telek>

```

14     if(j < forrasTomb2.length) {
15         metszetTomb[k] = forrasTomb1[i];
16         k += 1;
17     }
18 }
    console.log(metszetTomb)

```

Únió



```

1     'use strict';
2
3

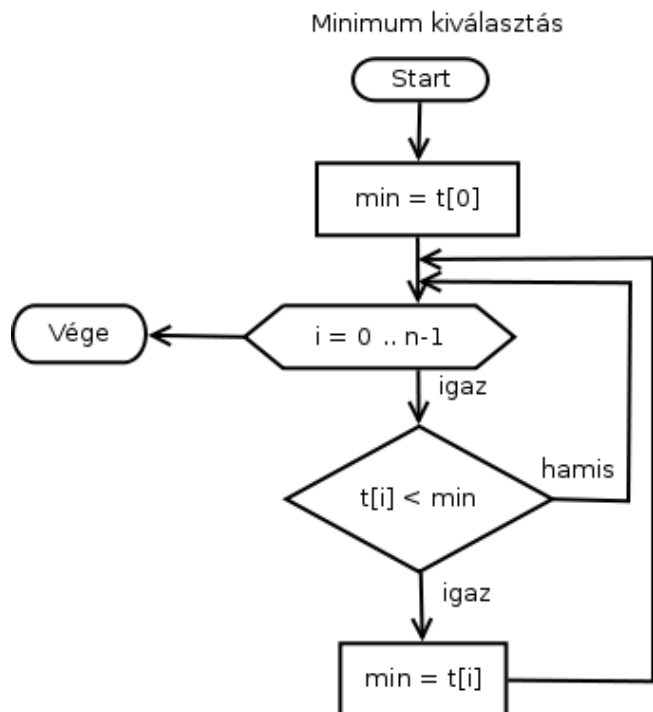
```

```

4  let i = 0;
5  let j;
6  let k = 0;
7  let forrasTomb1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 0];
8  let forrasTomb2 = [1, 2, 3, 5, 6, 8, 9, 12, 13];
9  let unioTomb = [];
10
11  for(i; i < forrasTomb1.length; i++) {
12      unioTomb[i] = forrasTomb1[i];
13  }
14
15  k = forrasTomb1.length;
16
17  for(j; j < forrasTomb2.length; i++) {
18      while(i < forrasTomb1.length && forrasTomb2[j] != forrasTomb1[i]) {
19          i += 1;
20      }
21      if(i >= n) {
22          unioTomb[k] = forrasTomb2[j];
23          k += 1;
24      }
25  }
26
27  console.log(unioTomb)

```

Minimum / maximum érték keresése



```

1  let tomb = [1, 2, 3, 4, 5, 6];
2  let maximum = tomb[0];
3  for(let i = 0; i < tomb.length, i++;) {
4

```

```
5     if(tomb[i] > maximum) {  
6         maximum = tomb[i]  
7     }  
8 }  
    console.log(maximum)
```

Rendezési algoritmusok

Minimum-kiválasztásos rendezés

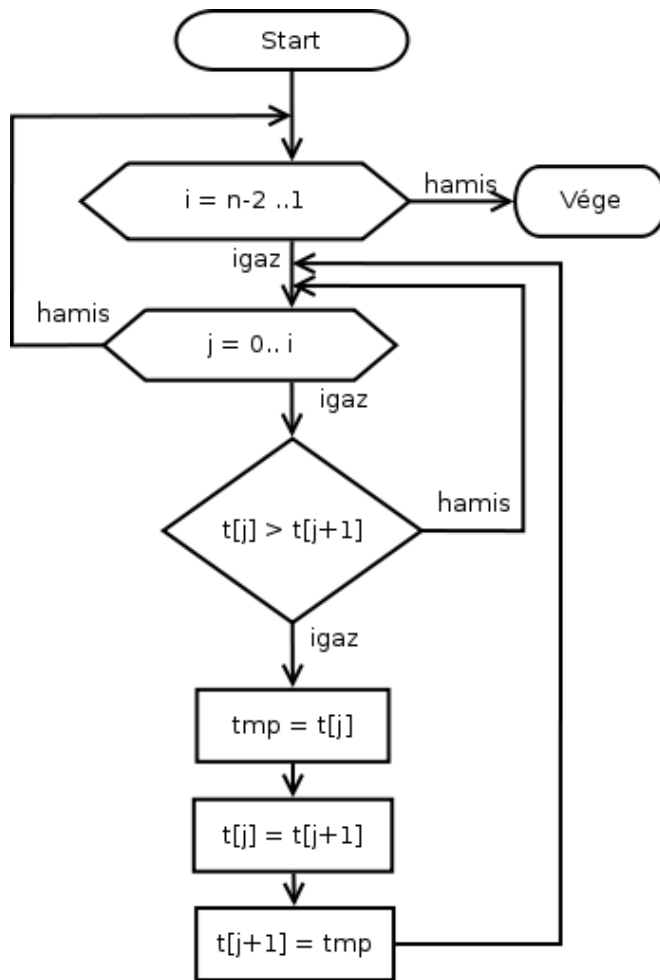
```
'use strict'  
let tomb = [ 4, 5, 2, 13, 8, 9 ]  
let len = tomb.length  
let min = 0  
  
for (let i=0; i < len; i++){  
    min = i  
    for (let j=i+1; j < len; j++){  
        if (tomb[j] < tomb[min]){  
            min = j  
        }  
    }  
    if (i != min){  
        let temp = tomb[i]  
        tomb[i] = tomb[min]  
        tomb[min] = temp  
    }  
}  
  
console.log(tomb)
```

Beszúrásos rendezés

```
'use strict'  
let tomb = [ 9, 8, 7, 6, 5, 4, 3, 2, 1 ]  
let n = tomb.length  
  
for (let i=1; i < n; i++){  
    let temp = tomb[i]  
    let j = i-1  
    while(j>=0 && tomb[j]>temp) {  
        tomb[j+1] = tomb[j]  
        j = j - 1  
    }  
    tomb[j+1] = temp  
}
```

```
console.log(tomb)
```

Buborékos rendezés (szomszédos elemek cseréje)



```

'use strict'
let tomb = [ 4, 5, 2, 13, 8, 9 ]
let n = tomb.length

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

console.log(tomb)

```

Cserélő rendezés

```
'use strict'
let tomb = [ 4, 5, 2, 13, 8, 9 ]
let n = tomb.length
for(let i = 1; i<n-1; i++) {
    for(let j = i+1; j<n; j++) {
        if(tomb[i] > tomb[j]) {
            let temp = tomb[i]
            tomb[i] = tomb[j]
            tomb[j] = temp
        }
    }
}
console.log(tomb)
```


Kapcsolódó anyagok

[http://szit.hu/doku.php?](http://szit.hu/doku.php?id=oktatas:programoz%C3%A1s:programoz%C3%A1s_t%C3%A9telek:mondatszer%C5%B1_le%C3%ADr%C3%A1s)

[id=oktatas:programoz%C3%A1s:programoz%C3%A1s_t%C3%A9telek:mondatszer%C5%B1_le%C3%ADr%C3%A1s](http://szit.hu/doku.php?id=oktatas:programoz%C3%A1s:programoz%C3%A1s_t%C3%A9telek:mondatszer%C5%B1_le%C3%ADr%C3%A1s)

<http://users.nik.uni-obuda.hu/sergyan/Programozas1Jegyzet.pdf>

http://progalap.elte.hu/downloads/seged/eTananyag/lecke16_lap1.html

 **Tetszik** Legyél az első, aki kedveli!

Címkék nélkül 