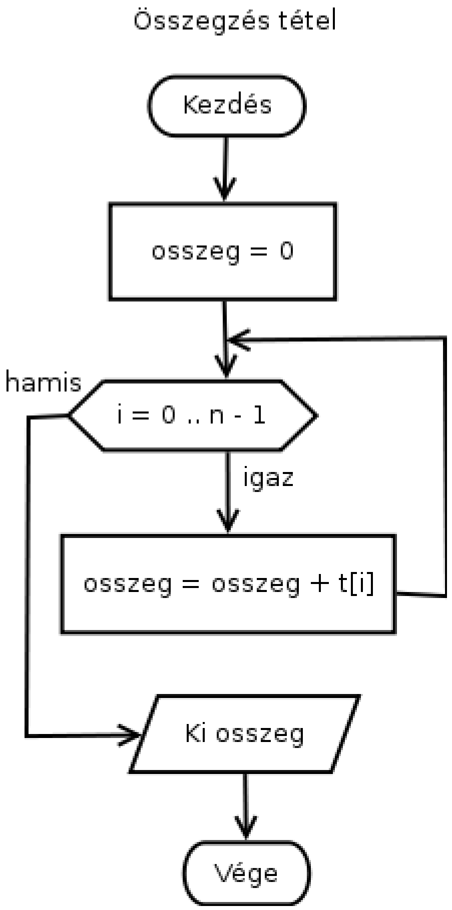
Programozási tételek

Felels: Zsolt Tasnádi

Elméleti órák: 10x1.5
Gyakorlati órák (napközi): 10 nap (4 óra /nap) (Feladatgyjtemények oldal)

Összegzés



Javascript

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

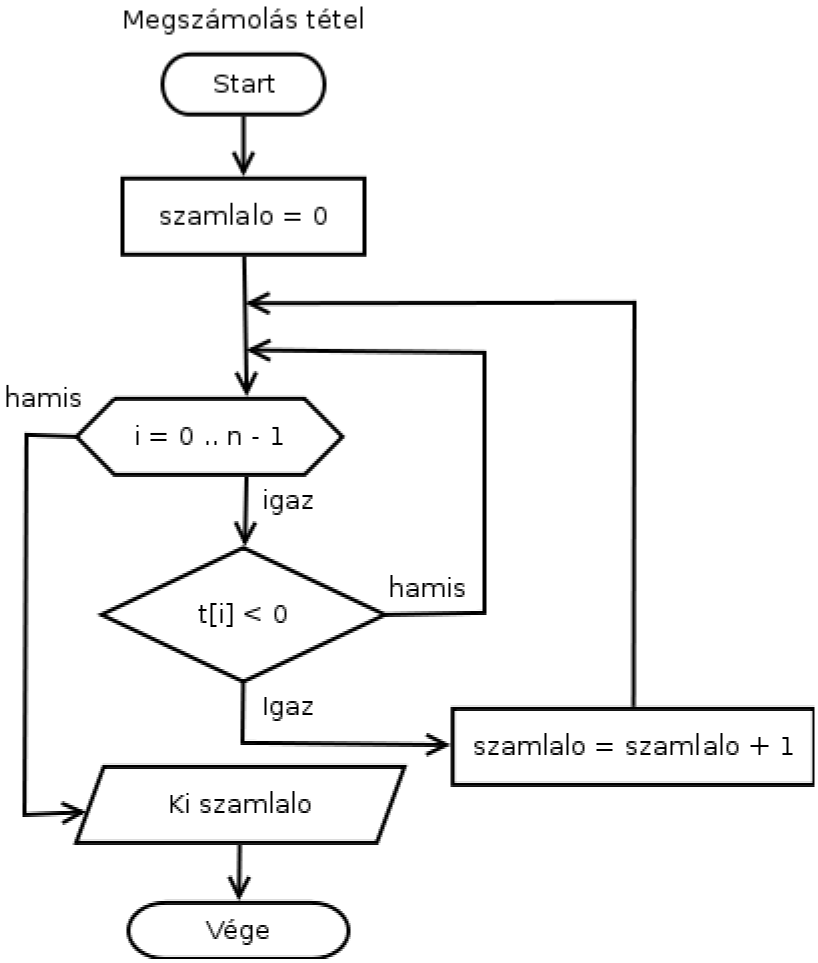
Go

```
package main

import "fmt"

func main() {
 var osszeg = int8(0)
 var tomb = []int8{1,2,3,4,5,6,7,8,9}
 for i := 0;i<len(tomb);i++{
  osszeg += tomb[i]
 }
 fmt.Println(osszeg)
}
```

Python:

```
osszeg = 0
tomb = [1, 2, 3, 4, 5, 6, 7, 8, 9]
for i in range(0, (len(tomb) - 1)):
  osszeg += tomb[i]
print(osszeg)
```

Megszámlálás



Javascript

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

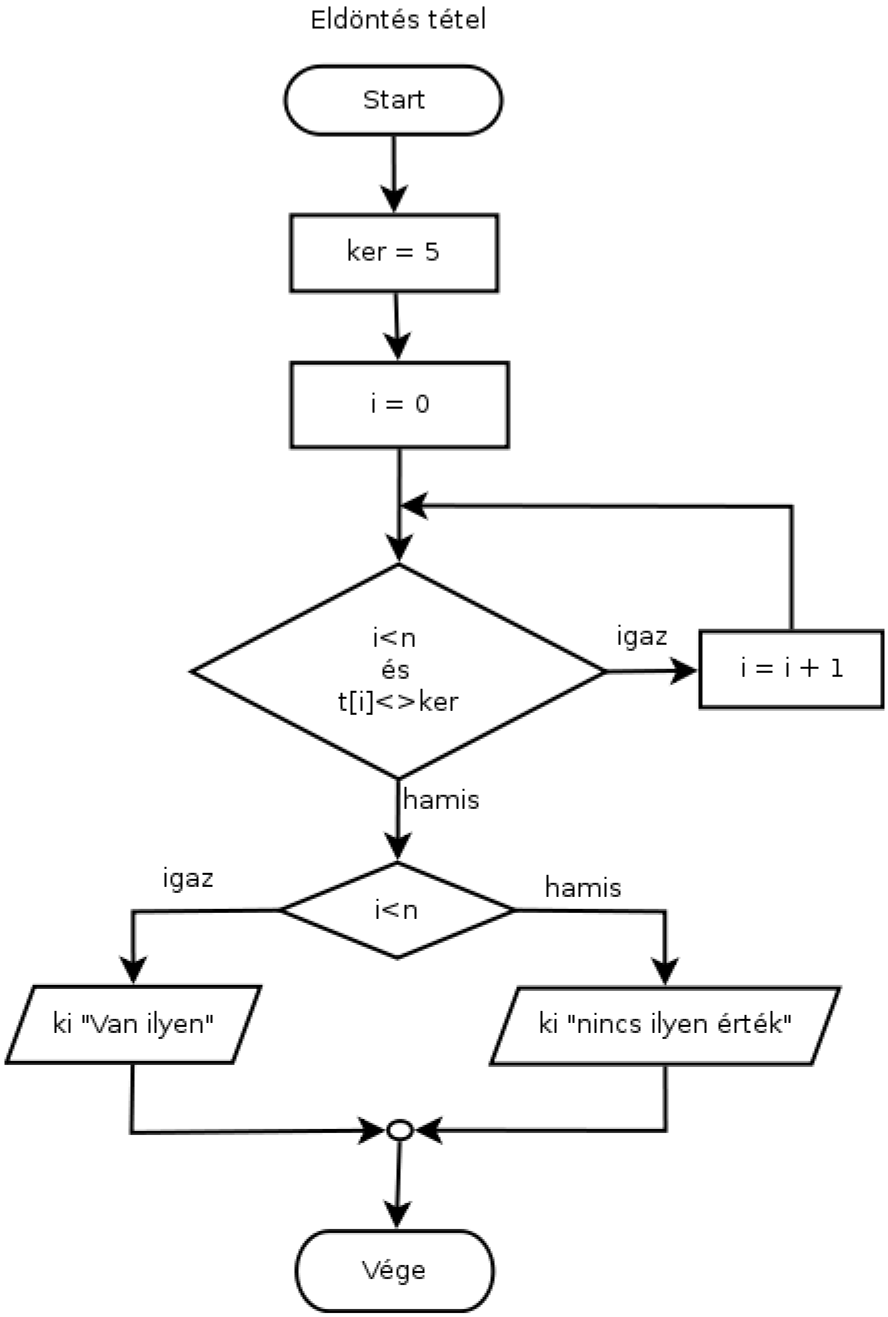
Go

```
package main

import "fmt"

func main() {
 var szamlalo = int8(0)
 var tomb = []int8{1,2,3,4,5,6,7,8,9}
 for i := 0;i<len(tomb);i++{
  if tomb[i] > 3 {
   szamlalo++
  }
 }
 fmt.Println(szamlalo)
}
```

Python:

```
szamlalo = 0
tomb = [1, 2, 3, 4, 5, 6, 7, 8, 9]
for i in range(0, (len(tomb) - 1)):
  if tomb[i] > 3:
    szamlalo++
print(osszeg)
```

Eldöntés



Javascript

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

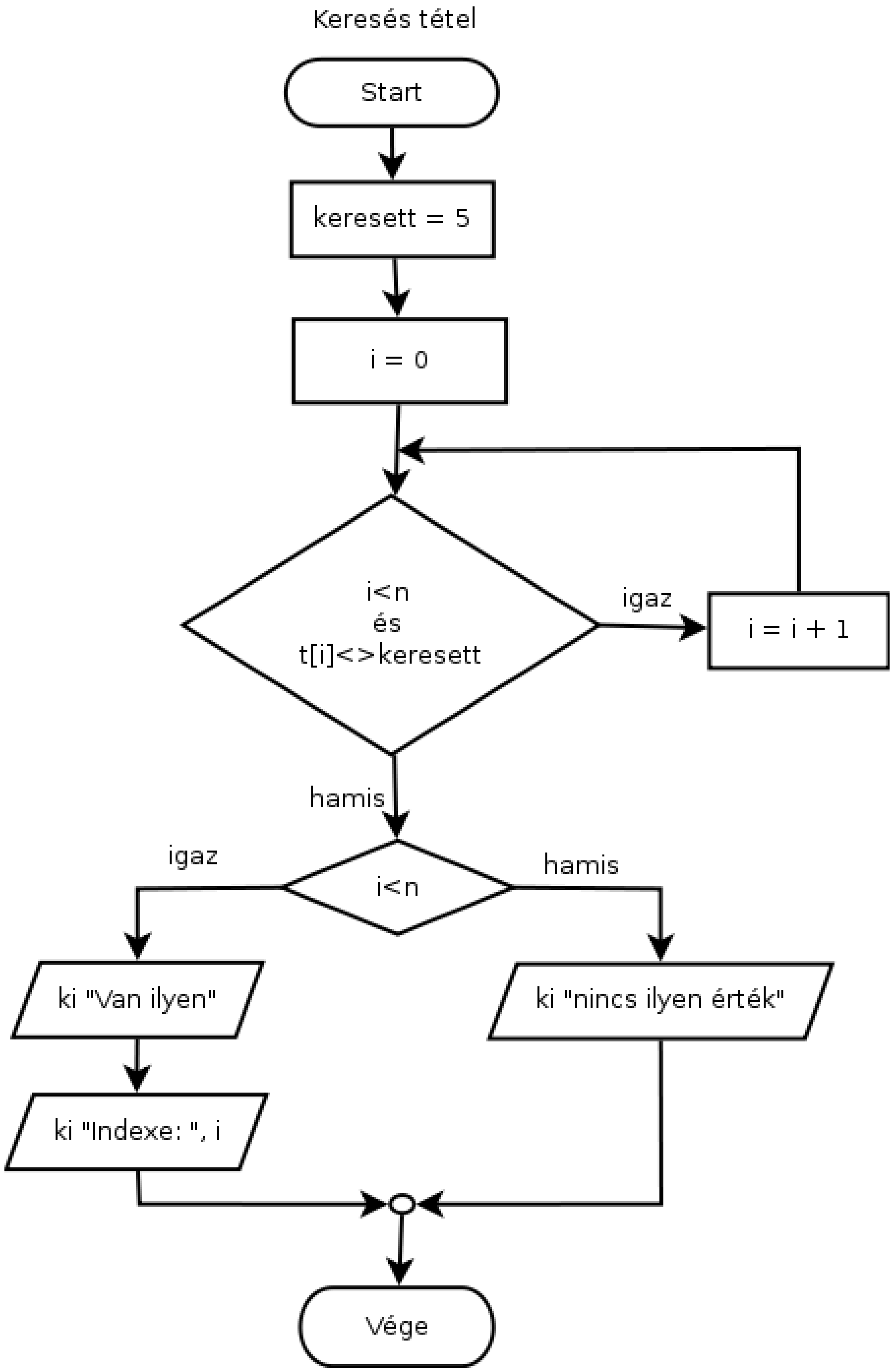
Go

```
package main

import "fmt"

func main() {
 var van = false
 var tomb = []int8{1,2,3,4,5,6,7,8,9}
 for i := 0;i<len(tomb);i++{
  if tomb[i] > 3 {
   van = true
  }
 }
 fmt.Println(van)
}
```

Python:

```
szamlalo = 0
tomb = [1, 2, 3, 4, 5, 6, 7, 8, 9]
for i in range(0, (len(tomb) - 1)):
  if tomb[i] > 3:
    szamlalo++
print(osszeg)
```

Kiválasztás


Keresés tétel
Start
keresett = 5
i = 0
i<n
és
t[i]<>keresett
igaz
i = i + 1
hamis
igaz
i<n
hamis
ki "Van ilyen"
ki "nincs ilyen érték"
ki "Indexe: ", i
Vége

Javascript

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

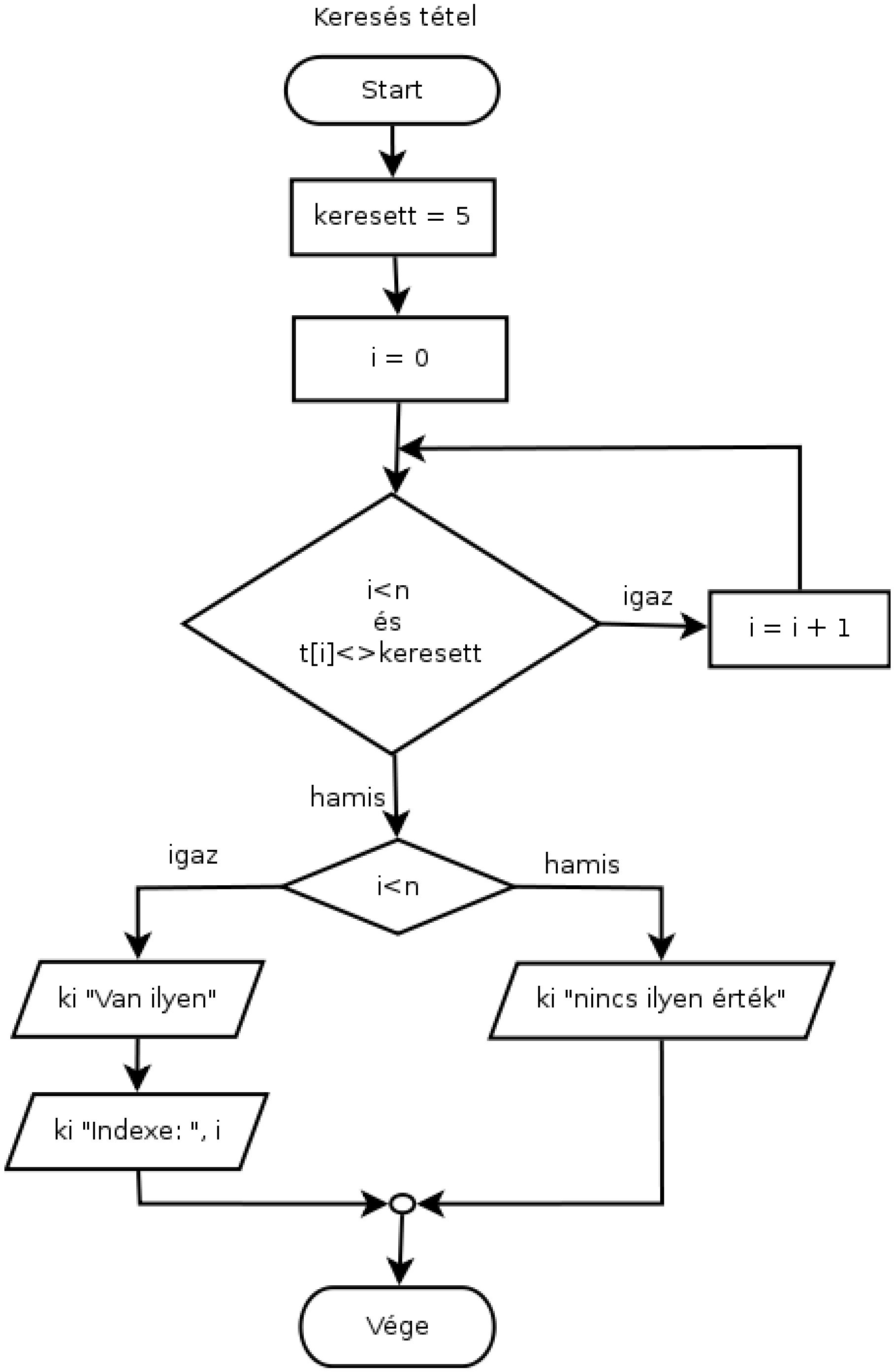
Go

```
package main

import "fmt"

func main() {
 var i = 0
 var keresett = 7
 var tomb = []int{1, 2, 3, 4, 5, 6, 7, 8, 9}
 for i < len(tomb) && tomb[i] != keresett {
  i++
 }
 if i < len(tomb) {
  fmt.Println("van ilyen, a ", i+1, ". elem")
 } else {
  fmt.Println("nincs ilyen")
 }
}
```

Python

```
i = 0
keresett = 7
tomb = [1, 2, 3, 4, 5, 6, 7, 8, 9]
while i < len(tomb) && tomb[i] != keresett:
  i += 1

if i < tomb.length:
  console.log("van ilyen, a ", i+1, ". elem")
else:
  console.log("nincs ilyen")
```

Keresés



Javascript

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

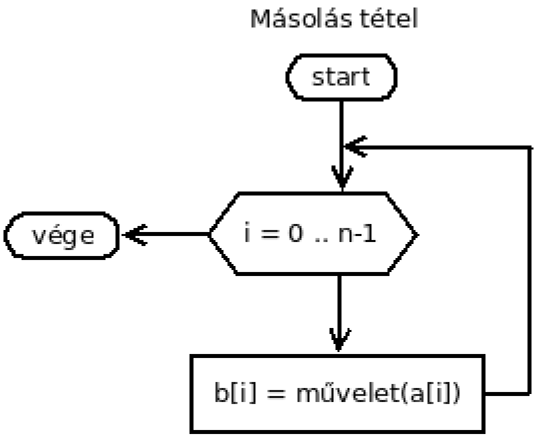
Go

```
package main

import "fmt"

func main() {
 var i = 0
 var keresett = 7
 var tomb = []int{1, 2, 3, 4, 5, 6, 7, 8, 9}
 for i < len(tomb) && tomb[i] != keresett {
  i++
 }
 if i < len(tomb) {
  fmt.Println("van ilyen, a ", i+1, ". elem")
 } else {
  fmt.Println("nincs ilyen")
 }
}
```

Python

```
i = 0
keresett = 7
tomb = [1, 2, 3, 4, 5, 6, 7, 8, 9]
while i < len(tomb) && tomb[i] != keresett:
  i += 1

if i<n:
  console.log("van ilyen, a ", i+1, ". elem")
else:
  console.log("nincs ilyen")
```

Másolás



Javascript

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

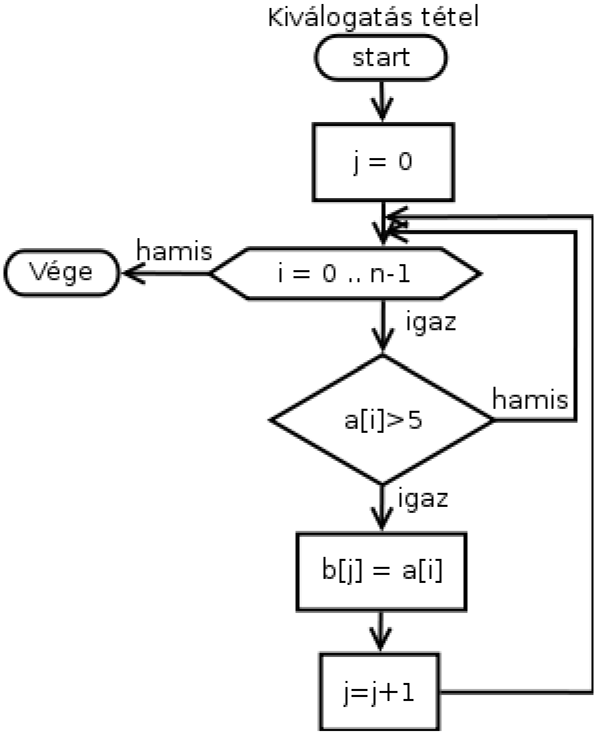
Go

```
package main

import "fmt"

func main() {
 var innenTomb = []int{1, 2, 3, 4, 5, 6, 7, 8, 9}
 var ideTomb = make([]int, len(innenTomb), len(innenTomb))
 for i := 0; i < len(innenTomb); i++ {
  ideTomb[i] = innenTomb[i] * 2
 }
 fmt.Println(ideTomb)
}
```

Python

```
innenTomb = [1, 2, 3, 4, 5, 6, 7, 8, 9]
ideTomb = []
for i in range(0, len(innenTomb)):
  ideTomb.append(innenTomb[i] * 2)
print(ideTomb)
```

Kiválogatás



Javascript

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

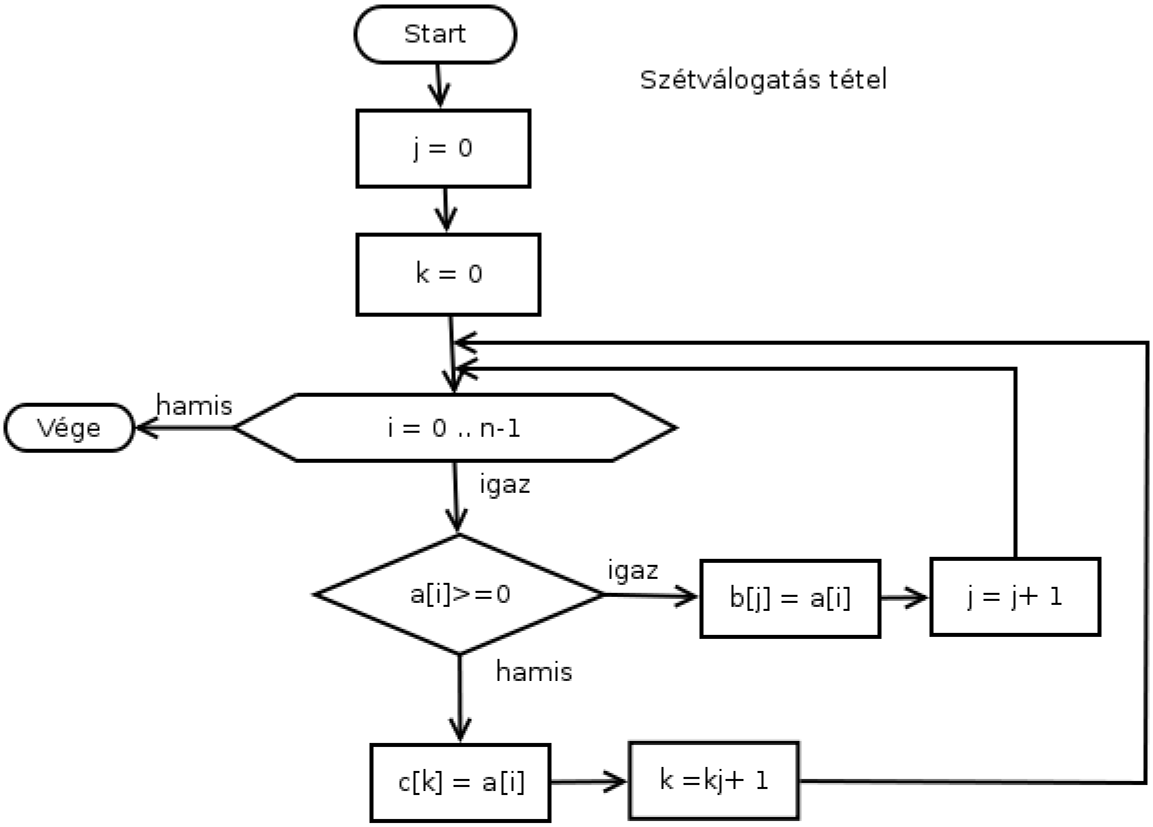
Go

```
package main

import (
 "fmt"
)

func main() {
 var innenTomb = []int{1, 2, 3, 4, 5, 6, 7, 8, 9}
 var ideTomb []int
 for i := 0; i < len(innenTomb); i++ {
  if (innenTomb[i] % 2) == 0 {
   ideTomb = append(ideTomb, innenTomb[i])
  }
 }
 fmt.Println(ideTomb)
}
```

Python

```
i = 0
j = 0
innenTomb = [1, 2, 3, 4, 5, 6, 7, 8, 9]
ideTomb = []
for i in range(0, len(innenTomb)):
  if (innenTomb[i] % 2) == 0:
 ideTomb[j] = innenTomb[i]
 j += 1
print(ideTomb)
```

Szétválogatás



Javascript

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

Go

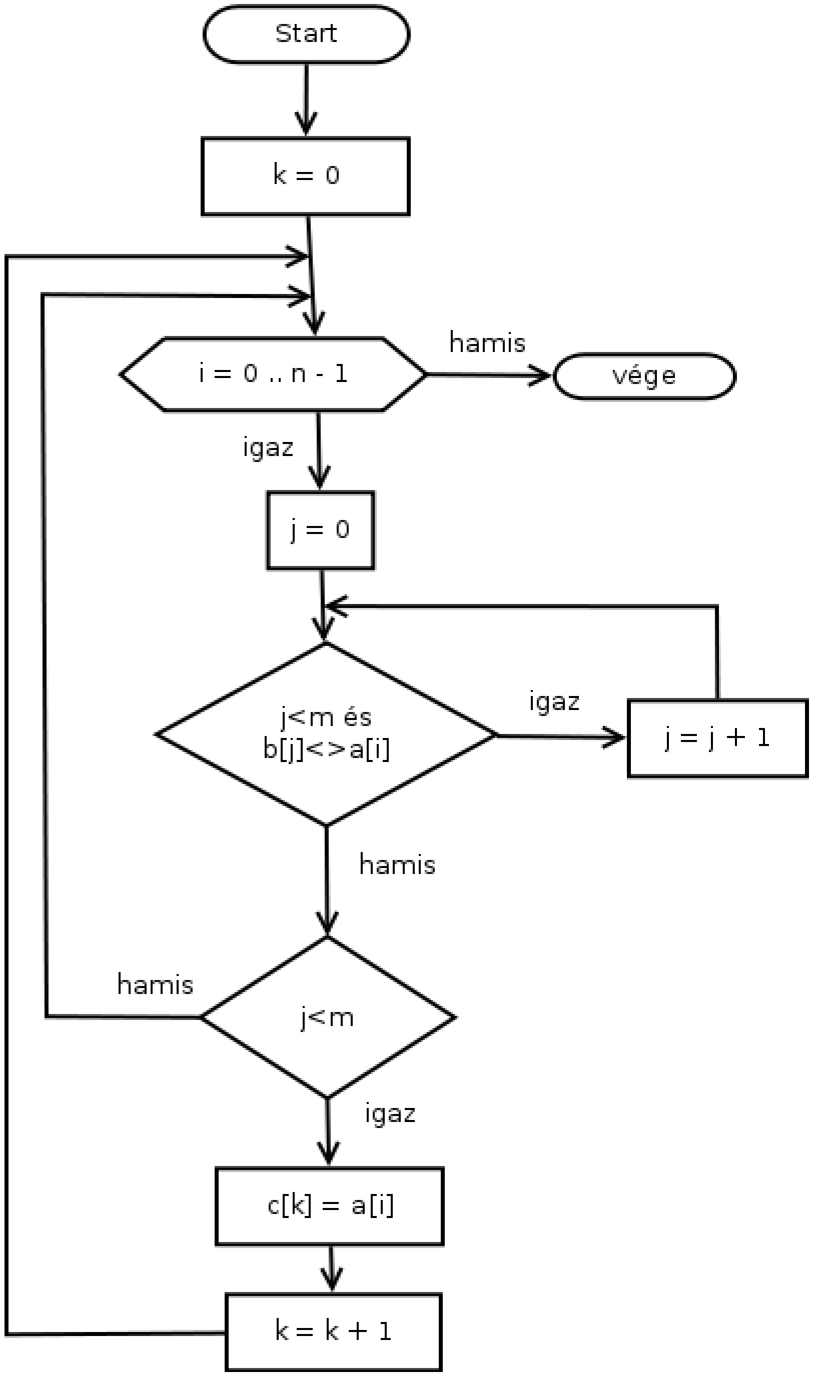
```
package main

import (
 "fmt"
)

func main() {
 var innenTomb = []int{1, 2, 3, 4, 5, 6, 7, 8, 9}
 var ideTomb1 []int
 var ideTomb2 []int
 for i := 0; i < len(innenTomb); i++ {
  if (innenTomb[i] % 2) == 0 {
   ideTomb1 = append(ideTomb1, innenTomb[i])
  } else {
   ideTomb2 = append(ideTomb2, innenTomb[i])
  }
 }
 fmt.Println(ideTomb1)
 fmt.Println(ideTomb2)
}
```

Python

```
i = 0
j = 0
k = 0
innenTomb = [1, 2, 3, 4, 5, 6, 7, 8, 9]
ideTomb1 = []
ideTomb2 = []
for i in range(0, len(innenTomb)):
  if (innenTomb[i] % 2) == 0:
 ideTomb[j] = innenTomb[i]
 j += 1
  else:
    idetomb[k] = innenTomb[i]
    k += 1
print(ideTomb1)
print(ideTomb2)
```

Metszet

Metszet



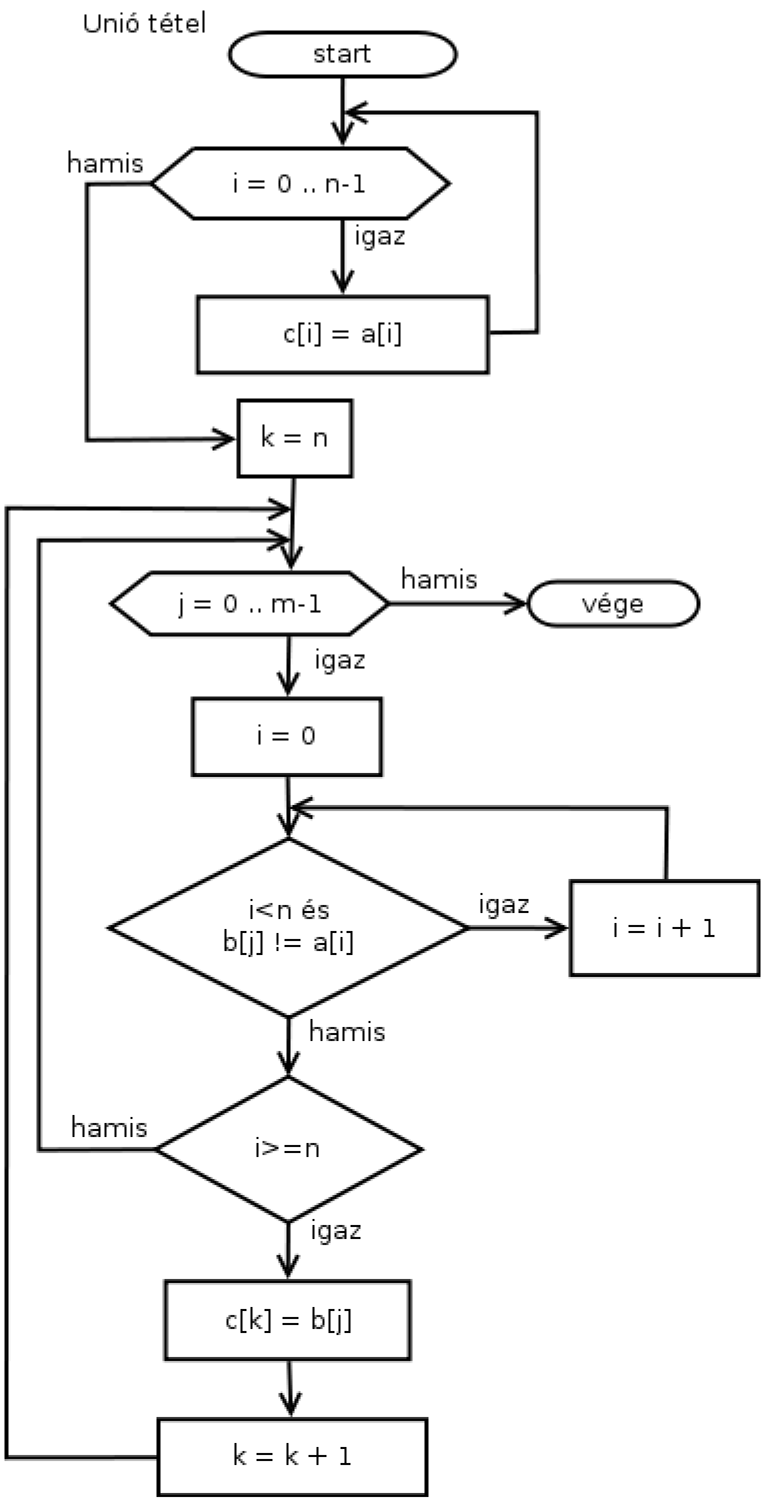
Javascript

```
let i = 0;
let j;
let k = 0;
let forrasTomb1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 0];
let forrasTomb2 = [1, 2, 3, 5, 6, 8, 9, 12, 13];
let metszetTomb = [];

for(i; i < forrasTomb1.length; i++) {
    j = 0;
    while(j < forrasTomb2.length && forrasTomb1[i] != forrasTomb2[j]) {
        j += 1;
    }
    if(j < forrasTomb2.length) {
        metszetTomb[k] = forrasTomb1[i];
        k += 1;
    }
}
console.log(metszetTomb)
```

Go

```
package main

import (
 "fmt"
)

func main() {
 var j int
 var forrasTomb1 = []int{1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 0}
 var forrasTomb2 = []int{1, 2, 3, 5, 6, 8, 9, 12, 13}
 var metszetTomb []int

 for i := 0; i < len(forrasTomb1); i++ {
  j = 0
  for j < len(forrasTomb2) && forrasTomb1[i] != forrasTomb2[j] {
   j += 1
  }
  if j < len(forrasTomb2) {
   metszetTomb = append(metszetTomb, forrasTomb1[i])
  }
 }
 fmt.Println(metszetTomb)
}
```

Únió



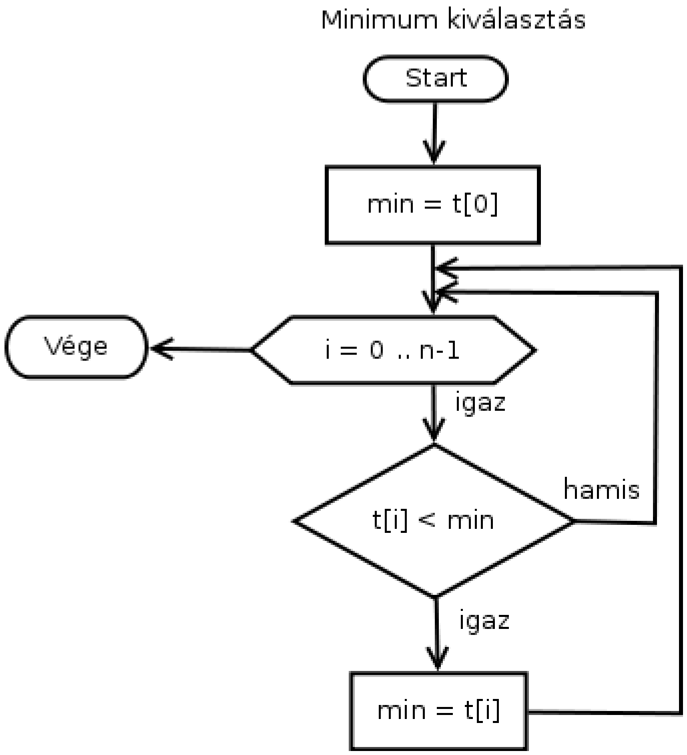
Javascript

```
'use strict';

let i;
let j;
let k;
let forrasTomb1 = [1, 2, 3, 4];
let n = forrasTomb1.length;
let forrasTomb2 = [1, 2, 3, 5, 6, 7];
let m = forrasTomb2.length;
let unioTomb = [];

for(i=0; i <= n-1; i++) {
   unioTomb[i] = forrasTomb1[i];
}

k = n;

for(j=0; j <= m-1; j++) {
   i = 0;
   while(i < n && forrasTomb2[j] != forrasTomb1[i]) {
       i++;
   }
   if(i>=n) {
       unioTomb[k] = forrasTomb2[j];
       k++;
   }
}

console.log(unioTomb)
```

Go

```go
package main

import (
 "fmt"
)

func main() {
 var i int
 var forrasTomb1 = []int{1, 2, 3, 4}
 var n = len(forrasTomb1)
 var forrasTomb2 = []int{1, 2, 3, 5, 6, 7}
 var m = len(forrasTomb2)
 var unioTomb []int

 for i := 0; i <= n-1; i++ {
  unioTomb = append(unioTomb, forrasTomb1[i])
 }

 for j := 0; j <= m-1; j++ {
  i = 0
  for i < n && forrasTomb2[j] != forrasTomb1[i] {
   i++
  }
  if i >= n {
   unioTomb = append(unioTomb, forrasTomb2[j])
  }
 }

 fmt.Println(unioTomb) // output: [1 2 3 4 5 6 7]
}
```

Minimum / maximum érték keresése



Javascript

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

Go

```
package main

import (
 "fmt"
)

func main() {
 var tomb = []int{1, 2, 3, 4, 5, 6}
 var maximum = tomb[0]
 for i := 0; i < len(tomb); i++ {
  if tomb[i] > maximum {
   maximum = tomb[i]
  }
 }
 fmt.Println(maximum)
}
```

Rendezési algoritmusok

Minimum-kiválasztásos rendezés

Javascript

```
'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)
```

Go

```
package main

import (
 "fmt"
)

func main() {
 var tomb = []int{4, 5, 2, 13, 8, 9}
 var len = len(tomb)
 var min = 0

 for i := 0; i < len; i++ {
  min = i
  for j := i + 1; j < len; j++ {
   if tomb[j] < tomb[min] {
    min = j
   }
  }
  if i != min {
   var temp = tomb[i]
   tomb[i] = tomb[min]
   tomb[min] = temp
  }
 }

 fmt.Println(tomb)
}
```

Beszúrásos rendezés

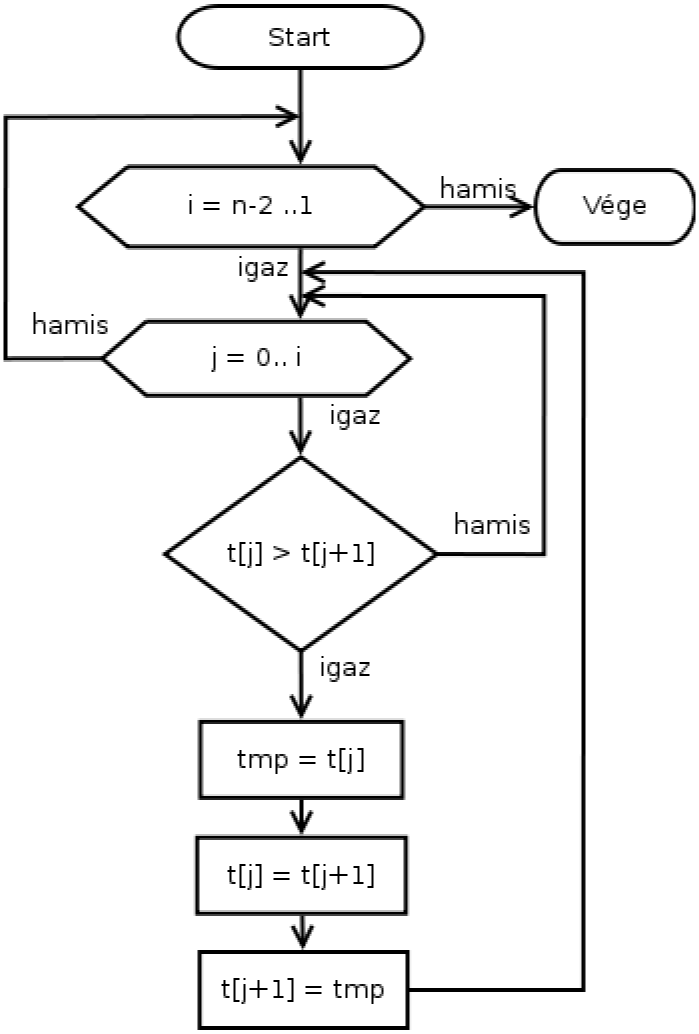
Javascript

```
'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)
```

Go

```
package main

import (
 "fmt"
)

func main() {
 var tomb = []int{9, 8, 7, 6, 5, 4, 3, 2, 1}
 var n = len(tomb)

 for i := 1; i < n; i++ {
  var temp = tomb[i]
  var j = i - 1
  for j >= 0 && tomb[j] > temp {
   tomb[j+1] = tomb[j]
   j = j - 1
  }
  tomb[j+1] = temp
 }

 fmt.Println(tomb)
}
```

Buborékos rendezés (szomszédos elemek cseréje)



Javascript

```
'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)
```

Go

```
package main

import (
 "fmt"
)

func main() {
 var tomb = []int{4, 5, 2, 13, 8, 9}
 var n = len(tomb)

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

 fmt.Println(tomb)
}
```

Cserél rendezés

Javascript

```
'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)
```

Go

```
package main

import (
 "fmt"
)

func main() {
 var tomb = []int{4, 5, 2, 13, 8, 9}
 var n = len(tomb)
 for i := 0; i < n-1; i++ {
  for j := i + 1; j < n; j++ {
   if tomb[i] > tomb[j] {
    var temp = tomb[i]
    tomb[i] = tomb[j]
    tomb[j] = temp
   }
  }
 }

 fmt.Println(tomb)
}
```

Kapcsolódó anyagok

http://szit.hu/doku.php?id=oktatas:programoz%C3%A1s:programoz%C3%A1si_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

http://info.nytta.hu/temak/prog/Programozasi_tetelek.pdf

http://progalap.elte.hu/downloads/seged/eTananyag/lecke4_lap1.html#hiv2