

An aerial photograph of Budapest, Hungary, showing the city's architecture and the Danube River. A semi-transparent white rectangular box is centered over the image, containing the text "Programozás" and "2. előadás".

Programozás

2. előadás

Tartalom

- Ciklusok –
specifikáció+„algoritmika”+kódolás
- Tömbök
 - Egy bevezető példa a tömbhöz
 - A tömb
 - Elágazás helyett tömb
 - Konstans tömbök
 - Mátrixok



Feladat:

Add meg egy természetes szám (>1) **1-től különböző legkisebb osztóját!**

Specifikáció:

- Bemenet: $N \in \mathbb{N}$
- Kimenet: $O \in \mathbb{N}$
- Előfeltétel: $N > 1$
- Utófeltétel: $1 < O \leq N$ és $O \mid N$ és $\forall i (2 \leq i < O): i \nmid N$

Ciklusok

A megoldás reprezentálása:

Specifikáció:

- Bemenet: $N \in \mathbb{N}$
- Kimenet: $O \in \mathbb{N}$
- Előfeltétel: $N > 1$
- Utófeltétel: $1 < O \leq N$ és $O \mid N$ és $\forall i (2 \leq i < O): i \nmid N$

Változó

N : **Egész**

O : **Egész**

Programváltozók
deklarálása

Reprezentációs „szabály” a specifikáció \rightarrow reprezentáció
áttéréskor:

$N \rightarrow \text{Egész}$



Ciklusok

A megoldás ötlete:

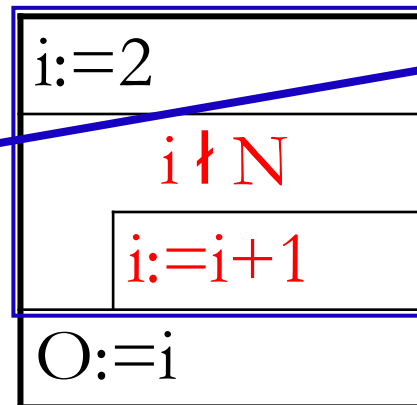
Próbáljuk ki a 2-t; ha nem jó, akkor a 3-at, ha az sem, akkor a 4-et, ...; legkésőbb az N jó lesz!

Az ezt kifejező lényegi algoritmus:

Az i változó szerepe: végigmenni egy halmaz elemein.

Specifikáció:

- Bemenet: $N \in \mathbb{N}$
- Kimenet: $O \in \mathbb{N}$
- Előfeltétel: $N > 1$
- Utófeltétel: $1 < O \leq N$ és $O \mid N$ és $\forall i (2 \leq i < O): i \nmid N$



Változó
 i :Egész

Lokális változó
deklarálása.



Ciklusok

Feladat:

Határozzuk meg egy természetes szám ($N > 1$) **1-től különböző legkisebb** és **önmagától különböző legnagyobb** osztóját!

Specifikáció:

- Bemenet: $N \in \mathbb{N}$
- Kimenet: $L_{ko}, L_{no} \in \mathbb{N}$
- Előfeltétel: $N > 1$
- Utófeltétel: $1 < L_{ko} \leq N$ és $1 \leq L_{no} < N$ és
 $L_{ko} \mid N$ és $\forall i (2 \leq i < L_{ko}): i \nmid N$ és
 $L_{no} \mid N$ és $\forall i (L_{no} < i \leq N-1): i \nmid N$

Specifikáció:

- Bemenet: $N \in \mathbb{N}$
- Kimenet: $O \in \mathbb{N}$
- Előfeltétel: $N > 1$
- Utófeltétel: $1 < O \leq N$ és $O \mid N$ és
 $\forall i (2 \leq i < O): i \nmid N$



Ciklusok

Megjegyzés:

A specifikációból az algoritmus megkapható, de az Lno az utófeltételben az Lko ismeretében másképp is megfogalmazható: $Lko * Lno = N$!

Az erre építő algoritmus:

Specifikáció:

- > Bemenet: $N \in \mathbb{N}$
- > Kimenet: $Lko, Lno \in \mathbb{N}$
- > Előfeltétel: $N > 1$
- > Utófeltétel: $1 < Lko \leq N$ és
 $Lko \mid N$ és
 $\forall i (2 \leq i < Lko): i \nmid N$ és
 $Lko * Lno = N$

$i := 2$

$i \nmid N$

$i := i + 1$

$Lko := i$

$Lno := N \text{ Div } Lko$

Változó

i :Egész



Ciklusok

Feladat:

Határozzuk meg egy természetes szám ($N > 1$) 1-től és önmagától különböző legkisebb osztóját (ha van)!

Specifikáció:

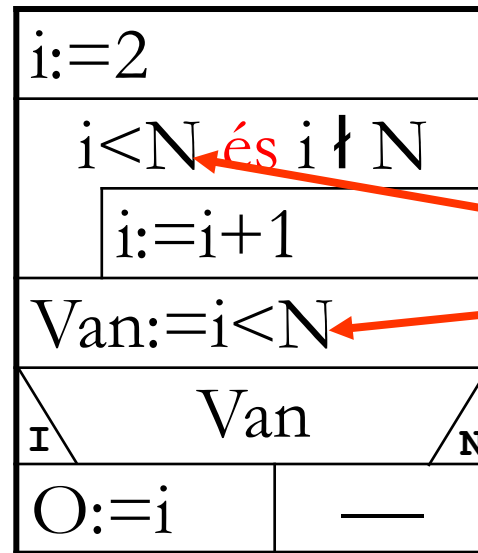
- Bemenet: $N \in \mathbb{N}$
- Kimenet: $O \in \mathbb{N}$, $Van \in \mathbb{L}$
- Előfeltétel: $N > 1$
- Utófeltétel: $Van = \exists i (2 \leq i < N) : i \mid N$ és
 $Van \rightarrow 2 \leq O < N$ és $O \mid N$ és $\forall i (2 \leq i < O) : i \nmid N$

Ciklusok

Algoritmus:

Specifikáció:

- Bemenet: $N \in \mathbb{N}$
- Kimenet: $O \in \mathbb{N}$, $Van \in \mathbb{L}$
- Előfeltétel: $N > 1$
- Utófeltétel: $Van \rightarrow \exists i (2 \leq i < N) : i \mid N$ és $Van \rightarrow 2 \leq O < N$ és $O \mid N$ és $\forall i (2 \leq i < O) : i \nmid N$



Változó
 i : Egész

$$i \leq \sqrt{N}$$

$$2 \rightarrow i \leq N \text{ Div } i \leftarrow N \text{ Div } 2$$

azaz

$$i * i \leq N$$

azaz

$$i \leq \sqrt{N}$$

Megjegyzés:

Ha i osztója N -nek, akkor $(N \text{ Div } i)$ is osztója, azaz elég az osztókat a szám gyökéig keresni!



Ciklusok

Feladat:

Határozzuk meg egy természetes szám ($N > 1$) osztói összegét!

Specifikáció:

- Bemenet: $N \in \mathbb{N}$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $N > 1$
- Utófeltétel: $S = \sum_{\substack{i=1 \\ i|N}}^N i$

A feltételes szumma értelmezéséhez egy példa:

$N=15 \rightarrow \Sigma=$

$i=1 : (1 \mid 15) \rightarrow 1$

$i=2 : (2 \nmid 15) \rightarrow 1$

$i=3 : (3 \mid 15) \rightarrow 1+3$

$i=4 : (4 \nmid 15) \rightarrow 1+3$

...

$i=15 : (15 \mid 15) \rightarrow 1+3+\dots+15$

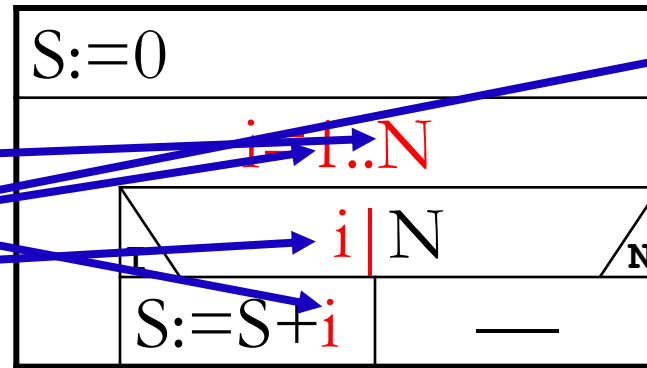
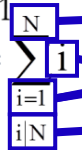


Ciklusok

Algoritmus:

Specifikáció:

- Bemenet: $N \in \mathbb{N}$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $N > 1$
- Utófeltétel: $S = \sum_{i=1}^N i$



Változó
 i :Egész

Az S változót nem egy képlettel számoljuk, hanem gyűjtjük benne az eredményt.

Kérdés:

Lehetne itt is gyök (N) -ig menni?

Az $S := S + i + (N \text{ Div } i)$ értékadással?



Ciklusok

Feladat:

Határozzuk meg egy természetes szám ($N > 1$) páratlan osztói összegét!

Specifikáció:

➤ Bemenet: $N \in \mathbb{N}$

➤ Kimenet: $S \in \mathbb{N}$

➤ Előfeltétel: $N > 1$

➤ Utófeltétel: $S = \sum_{\substack{i=1 \\ i|N \text{ és páratlan}(i)}}^N i$

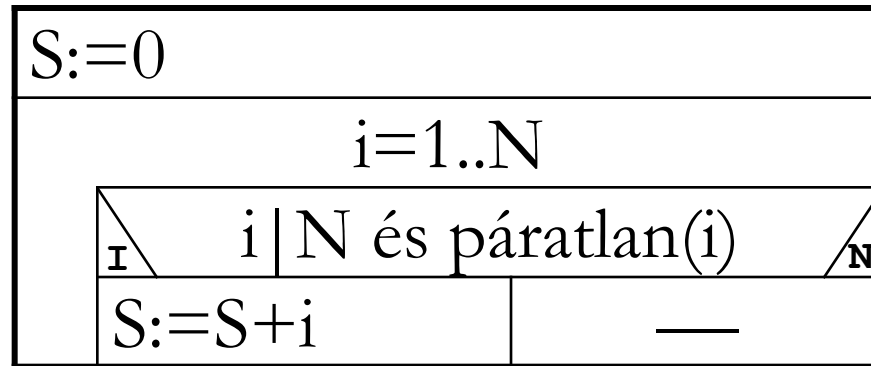
páratlan(i) = ???

Ciklusok

Algoritmus₁:

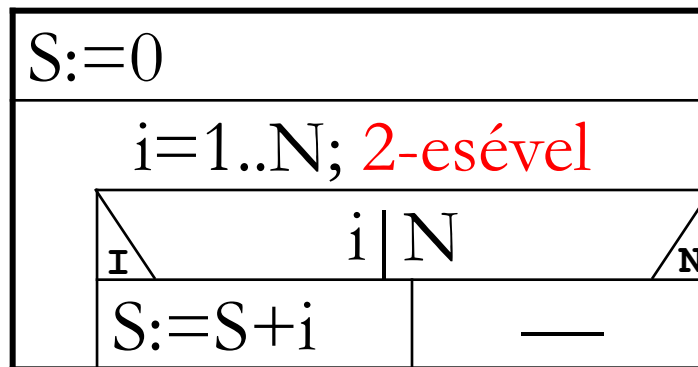
Specifikáció:

- Bemenet: $N \in \mathbb{N}$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $N > 1$
- Utófeltétel: $S = \sum_{\substack{i=1 \\ i|N \text{ és páratlan}(i)}}^N i$



Változó
 i : Egész

Algoritmus₂:



Változó
 i : Egész



Ciklusok

Feladat:

Határozzuk meg egy természetes szám ($N > 1$) prímosztói összegét!

Specifikáció:

- Bemenet: $N \in \mathbb{N}$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $N > 1$
- Utófeltétel: $S = \sum_{\substack{i=2 \\ i|N \text{ és } \text{prím}(i)}}^N i$
 $\text{prím}(i) = ???$

$$\begin{array}{c} N = i_1^{m_1} * i_2^{m_2} * \dots * i_k^{m_k} \\ \downarrow \\ S = i_1 + i_2 + \dots + i_k \end{array}$$



Ciklusok

Algoritmus:

A legkisebb osztó biztosan prím; ha N -t osztjuk vele ahányszor csak tudjuk, a következő osztója (a redukált N -nek) megint prím lesz.

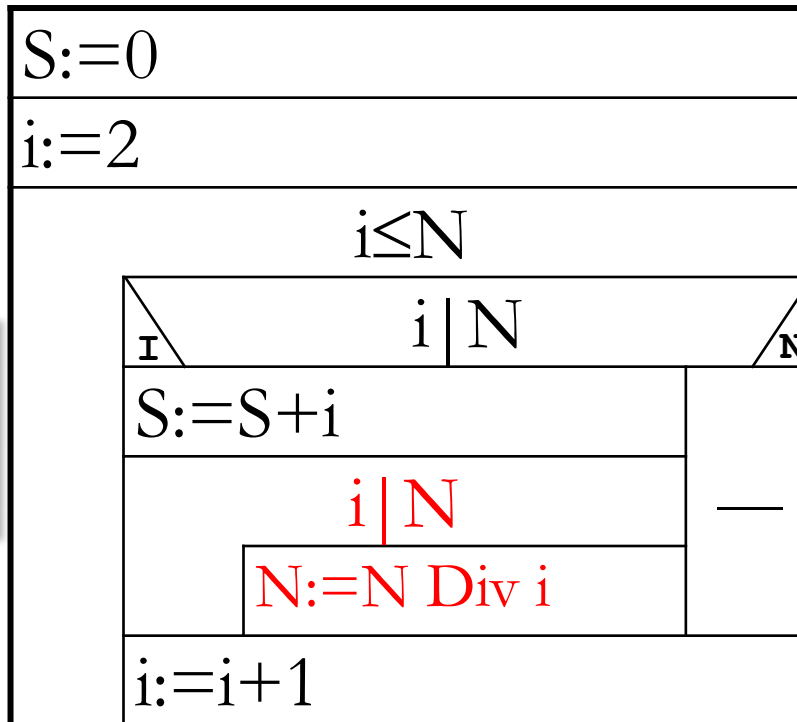
Specifikáció:

- Bemenet: $N \in \mathbb{N}$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $N > 1$
- Utófeltétel: $S = \sum_{\substack{i|N \\ i=2}}^N i$

$$N = i_1^{m_1} * i_2^{m_2} * \dots * i_k^{m_k}$$

↓

$$S = i_1 + i_2 + \dots + i_k$$



Változó
 i : Egész

Miért nem számlálós a külső ciklus?



Tanulságok:

- Ha az utófeltételben \exists , \forall , vagy Σ jel van, akkor a megoldás mindig **ciklus**!
- Ha az utófeltételben \exists vagy \forall jel van, akkor a megoldás sokszor **feltételes ciklus**!
- Ha az utófeltételben Σ jel van, akkor a megoldás sokszor **számlálós ciklus**! (Π is...)
- Feltételes Σ esetén a **ciklusban elágazás** lesz.



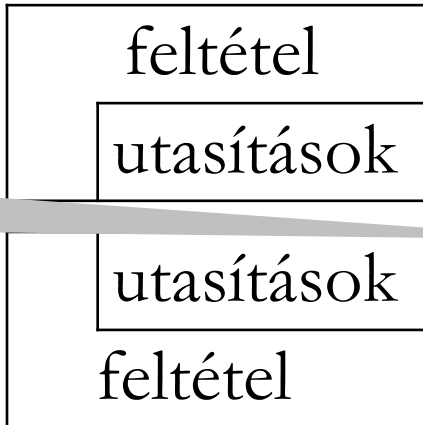
Ciklusok

algoritmus – kód



Feltételes ciklus:

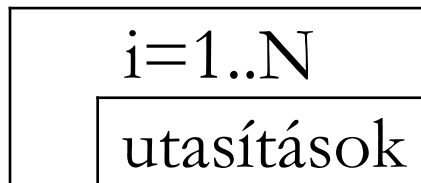
Tipikus előfordulás: a beolvasás ellenőrzésénél



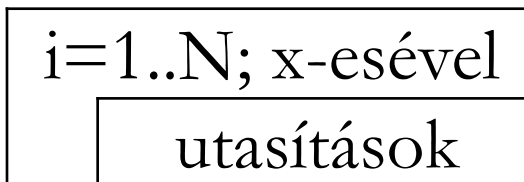
```
while (feltétel){  
    utasítások  
}
```

```
do{  
    utasítások  
}while (feltétel);
```

Számlálós ciklus:



```
for (int i=1;i<=N;++i){  
    utasítások  
}
```



```
for (int i=1;i<=N;i+=x){  
    utasítások  
}
```



Feladat elágazásra, vagy más megoldás kell?



Feladat:

A japán naptár 60 éves ciklusokat tartalmaz, az éveket párosítják, s mindegyik párhoz valamilyen színt rendelnek (zöld, piros, sárga, fehér, fekete).

- 1,2,11,12, ...,51,52: zöld évek
- 3,4,13,14,...,53,54: piros évek
- 5,6,15,16,...55,56: sárga évek
- 7,8,17,18,...57,58: fehér évek
- 9,10,19,20,...,59,60: fekete évek

Tudjuk, hogy 1984-ben indult az utolsó ciklus, amely 2043-ban fog véget érni.

Írj programot, amely megadja egy M évről ($1984 \leq M \leq 2043$), hogy milyen színű!



Feladat elágazásra, vagy más megoldás kell?



Specifikáció₁:

➤ Bemenet: $\text{év} \in \mathbb{N}$

➤ Kimenet: $s \in \mathbf{Szín}$

$\mathbf{Szín} = \{ \text{"zöld"}, \text{"piros"}, \text{"sárga"}, \text{"fehér"}, \text{"fekete"} \} \subset S$

➤ Előfeltétel: $1984 \leq \text{év}$ és $\text{év} \leq 2043$

➤ Utófeltétel: $((\text{év} - 1984) \bmod 10) \text{ Div } 2 = 0$ és $s = \text{"zöld"}$ vagy
 $((\text{év} - 1984) \bmod 10) \text{ Div } 2 = 1$ és $s = \text{"piros"}$ vagy ...

A Szín halmaz
definiálása.

- 1,2,11,12, ...,51,52: zöld évek
- 3,4,13,14,...,53,54: piros évek
- 5,6,15,16,...55,56: sárga évek
- 7,8,17,18,...57,58: fehér évek
- 9,10,19,20,...,59,60: fekete évek



Feladat elágazásra, vagy más megoldás kell?



Specifikáció₂:

➤ Bemenet: $\text{év} \in \mathbb{N}$

➤ Kimenet: $s \in \mathbf{Szín}$

$\mathbf{Szín} = \{ \text{"zöld"}, \text{"piros"}, \text{"sárga"}, \text{"fehér"}, \text{"fekete"} \} \subset S$

➤ Előfeltétel: $1984 \leq \text{év}$ és $\text{év} \leq 2043$

➤ Utófeltétel: $((\text{év} - 1984) \bmod 10) \text{ Div } 2 = 0 \rightarrow$
 $s = \text{"zöld"} \text{ és}$
 $((\text{év} - 1984) \bmod 10) \text{ Div } 2 = 1 \rightarrow$
 $s = \text{"piros"} \text{ és } \dots$

A Szín halmaz
definiálása.

- 1,2,11,12, ...,51,52: zöld évek
- 3,4,13,14,...,53,54: piros évek
- 5,6,15,16,...55,56: sárga évek
- 7,8,17,18,...57,58: fehér évek
- 9,10,19,20,...,59,60: fekete évek



Feladat elágazásra, vagy más megoldás kell?

Lokális változó
deklarációja



Változó
y:Egész

Algoritmus:

y:=((év-1984) Mod 10) Div 2				
y=0	y=1	y=2	y=3	y=4
s="zöld"	s="piros"	s="sárga"	s="fehér"	s="fekete"

Kérdés:

Akkor is ezt tennénk, ha 5 helyett 90 ágat kellene írunk?

A válasz előtt egy új adatszerkezet: a **tömb**.

Specifikáció₂:

- Bemenet: év ∈ ℕ
- Kimenet: s ∈ Szín
Szín = {"zöld", "piros", "sárga", "fehér", "fekete"} ⊂ S
- Előfeltétel: 1984 ≤ év és év ≤ 2043
- Utófeltétel: (((év-1984) Mod 10) Div 2 = 0 → s="zöld") és
(((év-1984) Mod 10) Div 2 = 1 → s="piros") és ...



Sorozatok

Specifikációbeli fogalmak:

- Sorozat: **azonos halmazbeli elemek** egymásutánja, az elemei sorszámozhatók.
- Elem: a sorozat **i-edik elemére** szokásos módon – alul-indexeléssel – hivatkozhatunk: S_i .
- Index: **1**..SorozatHossz vagy **0**..SorozatHossz-**1**...
- Például:
 - $\text{HónapHosszak}_{1..12} \in \mathbb{N}^{12}$ – a HónapHosszak 12 elemű, természetes számokból álló sorozat $\cong (\text{HónapHosszak}_1, \dots, \text{HónapHosszak}_{12})$
 - $\text{Emeletek}_{-1..10} \in \mathbb{S}^{12}$ – az Emeletek 12 elemű, szövegeket tartalmazó sorozat $\cong (\text{Emeletek}_{-1}, \text{Emeletek}_0, \dots, \text{Emeletek}_{10}) = (\text{"Pince"}, \text{"Földszint"}, \dots)$
- Kérdés: az elemek lehetnek sorozatok, azaz van-e **sorozat sorozata**?



Algoritmikus fogalmak:

- **Tömb:** véges hosszúságú sorozat algoritmikus párja, amelynek **i-edik tagjával** végezhetünk műveleteket (**adott a legkisebb és a legnagyobb index, vagy az elemszám**).
- **Index:** sokszor **1..N**, időnként **0..N-1**, ahol **N** az elemek számát jelöli. Más esetekben lehet **a..b** is ($a \leq b$). Egyes nyelvekben nem csak számmal lehet indexelni (pl. hétfő, kedd, ...).
- **Tömb-művelet:** **értékadás** (az értékazonosság operátort nem értelmezzük).
- **Tömbelem-műveletek:** **elemérték-hivatkozás, elemérték-módosítás** (az elem-indexeléssel kiválasztva).



Sorozatok \rightarrow Tömbök

Példa₁:

Specifikációban:

Bemenet: $\dots X_{1..N}, Y_{1..N} \in \mathbb{R}^N$
Kimenet: $Z_{1..N} \in \mathbb{R}^N$ } – deklarációs példa

Utófeltétel: $Z_1 = X_1 + Y_1 \dots$ – hivatkozási példa

Algoritmusban:

$X, Y, Z: \text{Tömb}[1..N: \text{Valós}]$ – deklarációs példa

$Z[1] := X[1] + Y[1]$ – hivatkozási példa



Sorozatok \rightarrow Tömbök

Példa₂:

Specifikációban:

Bemenet: Emeletek_{-1..10} $\in \mathbb{S}^{12}$ – deklarációs példa

Utófeltétel: Emeletek₋₁ = "Pince" ... – hivatkozási példa

Algoritmusban:

Emeletek: **Tömb**[-1..10:Szöveg] – deklarációs példa

Emeletek[-1] := "Pince" – hivatkozási példa



Sorozatok \rightarrow Tömbök

Példa₃:

Az előbbi feladatpélda Szín halmaza a specifikációban egy szöveg konstansokból álló **sorozat**tal ábrázolható:

Színek_{0..4} $\in \mathbb{S}^5 =$

("zöld", "piros", "sárga", "fehér", "fekete")

Az algoritmusban reprezentálhatjuk így:

Konstans Színek: **Tömb**[0..4: Szöveg] =

("zöld", "piros", "sárga", "fehér", "fekete")



Elágazás helyett tömb

Specifikáció (végleges):

➤ Bemenet: $\text{év} \in \mathbb{N}$

➤ Kimenet: $s \in \mathbf{S}$

$\text{Színek}_{0..4} \in \mathbf{S}^5 =$

("zöld", "piros", "sárga", "fehér", "fekete")

➤ Előfeltétel: $1984 \leq \text{év} \text{ és } \text{év} \leq 2043$

➤ Utófeltétel: $s = \text{Színek}_{((\text{év} - 1984) \bmod 10) \text{ Div } 2)}$

A Szín halmaz
reprezentálása.

Specifikáció₂:

➤ Bemenet: $\text{év} \in \mathbb{N}$

➤ Kimenet: $s \in \mathbf{Szín}$

$\mathbf{Szín} = \{"zöld", "piros", "sárga", "fehér", "fekete"\} \subset \mathbf{S}$

➤ Előfeltétel: $1984 \leq \text{év} \text{ és } \text{év} \leq 2043$

➤ Utófeltétel: $((\text{év} - 1984) \bmod 10) \text{ Div } 2 = 0 \text{ és } s = "zöld" \text{ vagy } ((\text{év} - 1984) \bmod 10) \text{ Div } 2 = 1 \text{ és } s = "piros" \text{ vagy } \dots$

A Szín halmaz
reprezentációjához
igazított utófeltétel.



Elágazás helyett tömb

Programparaméterek
deklarálása

➤ Adatreprezentálás:

Változó

év:Egész

s:Szöveg

Konstans

Színek:Tömb[0..4:Szöveg]=
("zöld","piros","sárga","fehér","fekete")

```
> Bemenet: év ∈ ℕ  
> Kimenet: s ∈ S  
Színek0..4 ∈ S5 =  
("zöld", "piros", "sárga", "fehér", "fekete")
```



Elágazás helyett tömb

Algoritmus:

➤ Adatreprezentálás:

Változó

év:Egész

s:Szöveg

Konstans

Színek:Tömb[0..4:Szöveg]=
("zöld","piros","sárga",
"fehér","fekete")

Tevékenység:

$$s := \text{Színek}[(\text{év} - 1984) \bmod 10 \text{ Div } 2]$$

➤ Bemenet: év ∈ ℕ
➤ Kimenet: s ∈ S
Színek_{0..4} ∈ S⁵ =
("zöld", "piros", "sárga", "fehér", "fekete")
➤ Előfeltétel: 1984 ≤ év és év ≤ 2043
➤ Utófeltétel: s = Színek_{(((év-1984) Mod 10) Div 2)}





Tömbök

(Algoritmus→kód)

indexelés **0..??? ⇒**
Tömb-**elemszám**

C++ **0**-val kezdi a tömbindexelést!

De szabad nem használni a 0-dikat. ☺

De negatív index sajnos nem használható! ☹

Deklarációs példák –

X:Tömb[**1**..N:Valós]

E:Tömb[**-1**..10:Szöveg]

Az előbbi Szín halmazos példa:

Konstans Színek:Tömb[0..4:Szöveg]=
("zöld","piros","sárga","fehér","fekete")

```
const string Szinek[5]=  
    {"zöld","piros","sárga","fehér","fekete"};
```

a C++ kódjukkal:

```
float X[N+1]
```

```
string E[12]
```



Tömbök

(Algoritmus → kód)

Kódolási kérdések ($x[0]$ -t **nem** használjuk)₁:

Algoritmus

$i = 1 \dots n$
$x[i] := i$

Változó
 i : Egész

Legegyszerűbb kódolása C++-ban

```
for (int i=1; i<=n; ++i) {  
    x[i]=i;  
}
```



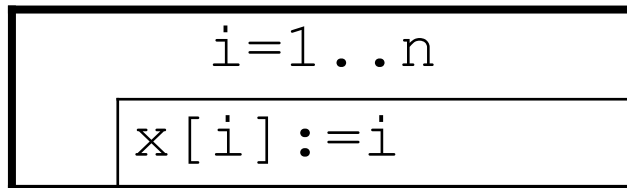
Tömbök

(Algoritmus → kód)



Kódolási kérdések ($x[0]$ -t **is** használjuk)₂:

Algoritmus



Változó
 i : Egész

kódolása C++-ban

a.

```
for (int i=1; i<=n; ++i) {  
    x[i-1]=i;  
}
```

b.

```
for (int i=0; i<=n-1; ++i) {  
    x[i]=i+1;  
}
```



Tömbök

(Algoritmus → kód)

Kódolási kérdések ($x[0]$ -t **is** használjuk)₃:

Algoritmus

$i = -1 \dots n$
$x[i] := 0$

Változó
 i : Egész

kódolása C++-ban

a.

```
for (int i=-1; i<=n; ++i) {  
    x[i+1]=0;  
}
```

b.

```
for (int i=0; i<=n+1; ++i) {  
    x[i]=0;  
}
```

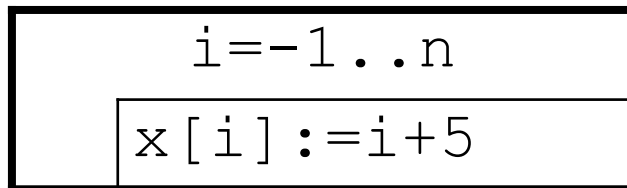


Tömbök

(Algoritmus → kód)

Kódolási kérdések ($x[0]$ -t **is** használjuk)₄:

Algoritmus



Változó
 i : Egész

kódolása C++-ban

a.

```
for (int i = -1; i <= n; ++i) {  
    x[i+1] = i + 5;  
}
```

b.

```
for (int i = 0; i <= n + 1; ++i) {  
    x[i] = i + 4;  
}
```





Tömbök

(C++ kódban – áttekintés)

Fordításkor kiderülő méret esete.

Statikus tömbök:

➤ Deklaráció:

```
const int MaxN=???; //tömb elemszáma  
típ tömb[MaxN];      //tömbdeklaráció,  
                      //0..MaxN-1 közötti indexekkel  
...
```

➤ Hivatkozások:

```
... tömb[ind] ... //tömbérték-hivatkozás  
...  
tömb[ind]=kif; //tömbérték-módosítás  
...
```



Tömbök

(C++ kódban – áttekintés)

Statikus tömb konstansok:

➤ Deklaráció:

```
const int N=???; //tömb elemszáma  
const típ tömb[N]={t1,t2,...,tN};  
                //konstans tömb deklarációja,  
                //0..N-1 közötti indexekkel
```



Tömbök

(C++ kódban – áttekintés)

Csak futáskor kiderülő méret esete.

Dinamikus tömbök:

➤ Deklaráció:

```
int N;           //tömb aktuális elemszám  
...
```

➤ Létrehozás:

```
N=???; //N meghatározása, pl. beolvasása  
típ tömb[N]; //tömbhelyfoglalás N db  
           //típ típusú elem számára
```

➤ Hivatkozások (nincs változás):

```
... tömb[ind] ... //tömbérték-hivatkozás  
...  
tömb[ind]=kif; //tömbérték-módosítás  
...
```



Konstans tömbök alkalmazása

Feladat:

Írj programot, amely egy 1 és 99 közötti számot betűkkel ír ki!

Specifikáció:

➤ Bemenet: $N \in \mathbb{N}$

Leglogikusabb helyre téve.
Az algoritmus szempontjából
„adottság”, azaz bemenet...

$\text{egy}es_{0..9} \in S^{10} = ("", "egy", \dots, "kilenc")$
 $\text{tíz}es_{0..9} \in S^{10} = ("", "tizen", \dots, "kilencven")$

➤ Kimenet: $S \in \mathcal{S}$

➤ Előfeltétel: $1 \leq N \leq 99$

➤ Utófeltétel: $N=10 \rightarrow S="tíz"$ és
 $N=20 \rightarrow S="húsz"$ és
 $N \notin \{10, 20\} \rightarrow S = \text{tíz}es_{(N \text{ Div } 10)} + \text{egy}es_{(N \text{ Mod } 10)}$



Konstans tömbök alkalmazása

Programparaméterek
deklarálása

Algoritmus:

Változó N:Egész

Konstans egyes:Tömb[0..9:Szöveg]=
("", "egy", ..., "kilenc")

tizes:Tömb[0..9:Szöveg]=
("", "tizen", ..., "kilencven")

Változó S:Szöveg

Specifikáció:

➤ **Bemenet:** $N \in \mathbb{N}$

egyes $\in S^{10} =$
("", "egy", ..., "kilenc")

tizes $\in S^{10} =$
("", "tizen", ..., "kilencven")

➤ **Kimenet:** $S \in S$

➤ **Utófeltétel:**

$N=10 \rightarrow S = \text{"tíz"}$ és

$N=20 \rightarrow S = \text{"húsz"}$ és

$N \notin \{10, 20\} \rightarrow S = \text{tizes}_{(N \text{ Div } 10)+1} +$
 $\text{egyes}_{(N \text{ Mod } 10)+1}$

$N=10$	$N=20$	$N \notin \{10, 20\}$
$S := \text{"tíz"}$	$S := \text{"húsz"}$	$S := \text{tizes}[N \text{ Div } 10] +$ $\text{egyes}[N \text{ Mod } 10]$



Konstans tömbök alkalmazása



Feladat:

Írj programot, amely egy hónapnévhez a sorszámát rendeli!

Specifikáció:

- Bemenet: $H \in S$
 $HóNév_{1..12} \in S^{12} = ("január", \dots, "december")$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $H \in HóNév$
- Utófeltétel: $1 \leq S \leq 12$ és $HóNév_S = H$



Konstans tömbök alkalmazása

Programparaméterek
deklarálása

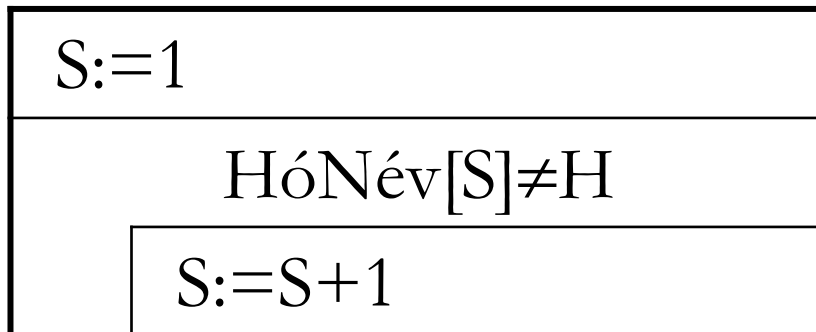
Algoritmus:

Változó H:Szöveg, S:Egész

Konstans HóNév:Tömb[1..12:Szöveg]=
("január",...,"december")

Specifikáció:

- Bemenet: $H \in S$
 $HóNév_{1..12} \in S^{12} = ("január", \dots)$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $H \in HóNév$
- Utófeltétel: $1 \leq S \leq 12$ és $HóNév_S = H$



Kérdés: mi lenne, ha az előfeltétel nem teljesülne?
Futási hiba? Végtelen ciklus?



Konstans tömb – mit tárolunk?

Feladat:

Egy nap a nem szökőév hányadik napja?

Specifikáció₁:

- Bemenet: $H, N \in \mathbb{N}$
 $\text{hó}_{1..12} \in \mathbb{N}^{12} = (31, 28, 31, \dots, 31)$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $1 \leq H \leq 12$ és $1 \leq N \leq \text{hó}_H$
- Utófeltétel: $S = N + \sum_{i=1}^{H-1} \text{hó}_i$



Konstans tömb – mit tárolunk?

Programparaméterek
deklarálása

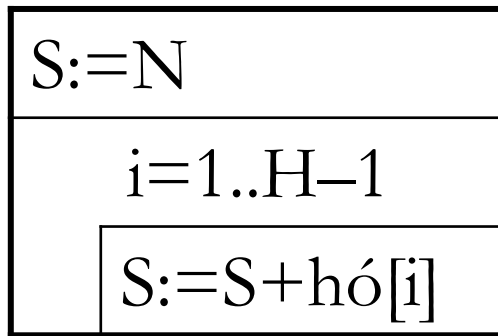
Algoritmus:

Változó H, N, S : **Egész**

Konstans hó: **Tömb** $[1..12$:**Egész**] $=$
 $(31, 28, 31, \dots, 31)$

Specifikáció₁:

- Bemenet: $H, N \in \mathbb{N}$
 $\text{hó}_{1..12} \in \mathbb{N}^{12} = (31, 28, 31, \dots, 31)$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $1 \leq H \leq 12$ és $1 \leq N \leq \text{hó}_1$
- Utófeltétel: $S = N + \sum_{i=1}^{H-1} \text{hó}_i$



Változó
 i : **Egész**

Lokális változó
deklarálása

Megjegyzés: szökőév esetén $H \geq 3$ esetén S -et 1-gyel meg kellene növelni! (És az előfeltétel is módosul.)



Konstans tömb – mit tárolunk?

Egy másik megoldás:

Tároljuk minden hónapra, hogy az előző hónapokban összesen hány nap van!

Specifikáció₂:

➤ Bemenet: ...

$$\text{hó}_{1..12} \in \mathbb{N}^{12} = (0, 31, 59, 90, \dots, 334)$$

➤ Utófeltétel: $S = \text{hó}_H + N$

Kérdés: Ez jobb megoldás? Mi lesz az előfeltétellel?



Mátrix

Lényeg:






Olyan sorozatféle, amely

1. azonos halmazbeli elemekből áll,
2. az elemeinek kiválasztásához 2 index kell.

Példa₁:

Egy sakkjátszma állása.

Az alaphalmaz:

{
 ,  ,  ,  ,  ,  ,
 ,  ,  ,  ,  ,  , Üres }



Mátrix

Példa₂:

N áruházban M-féle terméket árulnak. Nyilvántartjuk az egyes áruházak készletét.

Az alaphalmaz: \mathbb{N} – mennyiség

termék áruház	1.	...	M.
1.	Ennyi van az 1. áruházban az 1. termékből	...	Ennyi van az 1. áruházban az M. termékből
...
N.	Ennyi van az N. áruházban az 1. termékből	...	Ennyi van az N. áruházban az M. termékből



Mátrix



Feladat₁:

Feljegyeztük egy játszma végállását. Számoljuk meg, hány világos és hány sötét bábú maradt a táblán!

Feladat₂:

Határozzuk meg az egyes áruházakban tárolt készlet összértékét, ha ismerjük az egyes termékek árát!



Mátrix – Specifikációban

Specifikációbeli fogalmak:

- Sorozat: homogén, azaz **azonos halmazbeli elemek** egymás utánja, az elemei két indexszel sorszámozhatók.
- Elem: a sorozat **(i,j)-edik elem**ére szokásos módon – alul-indexeléssel – hivatkozhatunk: $M_{i,j}$.
- Index: $i \in 1..SorSzám$, $j \in 1..OszlopSzám$.



Mátrix – Specifikációban

Feladat₁:

Feljegyeztük egy játszma végállását. Számoljuk meg, hány világos és hány sötét bábu maradt a táblán!

Specifikáció₁:

Előzetes megfontolások:

* a mezőállapotokat kódoljuk:

Gyalog=1, Huszár=2, Futó=3, Bástya=4, Király=5, Vezér=6;

* a világos bábu pozitív, a sötét negatív értékű;

* az üres mező legyen 0.

➤ Bemenet: $VégÁllás_{1..8,1..8} \in \mathbb{Z}^{8 \times 8}$

➤ Kimenet: $DbV, DbS \in \mathbb{N}$

➤ Előfeltétel: $\forall i,j \in [1..8]: VégÁllás_{i,j} \in [-6..6]$

➤ Utófeltétel: $DbS = \sum_{i,j=1}^8 1$ és $DbV = \sum_{i,j=1}^8 1$
 $VégÁllás_{i,j} < 0$ $VégÁllás_{i,j} > 0$



Mátrix – Specifikációban

Feladat₂:

Határozzuk meg az egyes áruházakban tárolt készlet összértékét, ha ismerjük az egyes termékek árát!

Specifikáció₂:

- Bemenet: $N, M \in \mathbb{N}$, $\text{Készlet}_{1..N, 1..M} \in \mathbb{N}^{N \times M}$
 $\text{Ár}_{1..M} \in \mathbb{N}^M$
- Kimenet: $\text{ÖsszÉrték}_{1..N} \in \mathbb{N}^N$
- Előfeltétel: –
- Utófeltétel: $\forall i \in [1..N]: \text{ÖsszÉrték}_i = \sum_{j=1}^M \text{Készlet}_{i,j} * \text{Ár}_j$



Mátrix – Algoritmusban

Deklaráció₁:

- Bemenet: $VégÁllás_{1..8,1..8} \in \mathbb{Z}^{8 \times 8}$
- Kimenet: $DbV, DbS \in \mathbb{N}$

Programparaméterek
deklarálása

Változó

$VégÁllás : \text{Tömb}[1..8, 1..8 : \text{Egész}]$
 $DbV, DbS : \text{Egész}$

Mátrix-elemre hivatkozás:

$TömbNév[sorIndex, oszlopIndex]$

Itt: $VégÁllás[i, j]$



„Dupla Σ ”:
 $i=1..8$ -ra és
 $j=1..8$ -ra

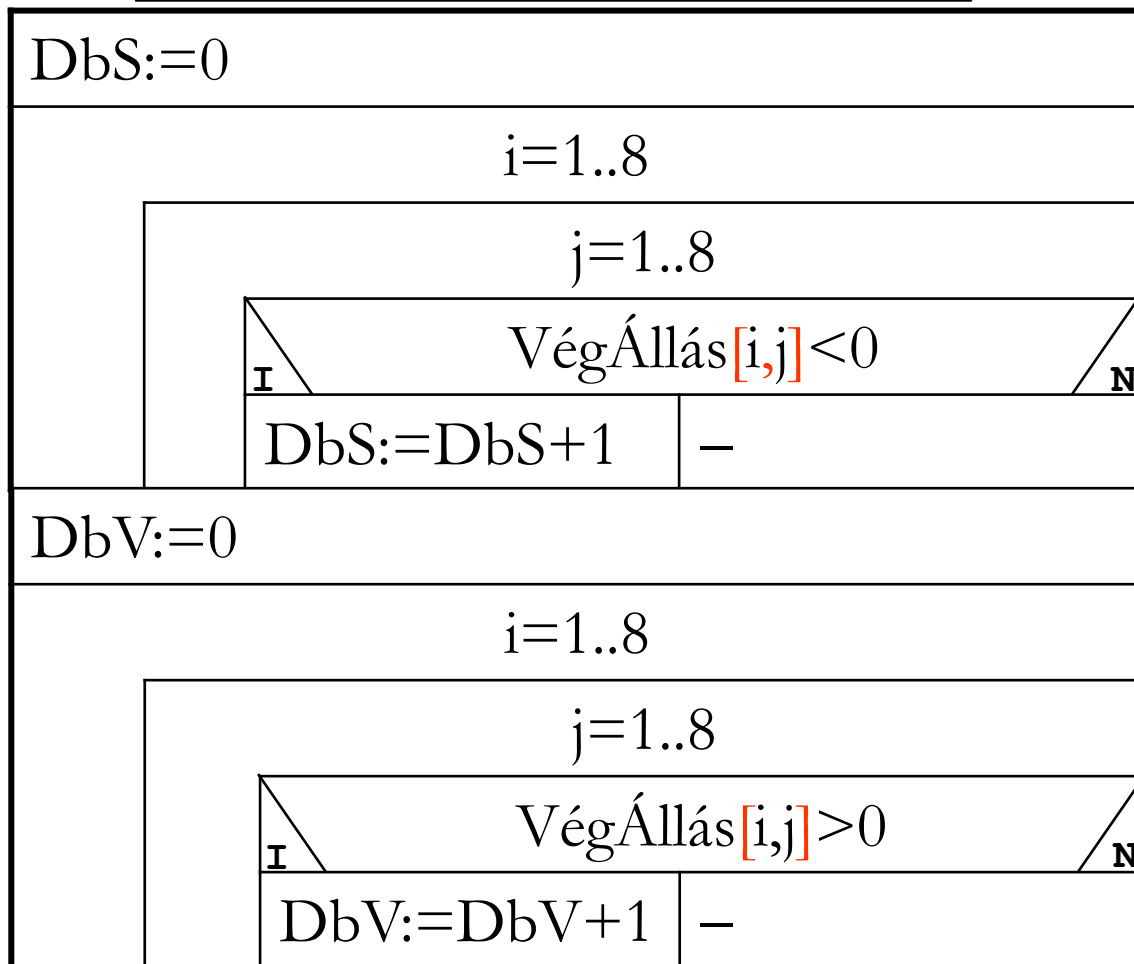
Mátrix – Algoritmusban

Lokális
 változók
 deklarálása

Algoritmus₁:

Utófeltétel: $\text{DbS} = \sum_{\substack{i,j=1 \\ \text{VégÁllás}_{i,j} < 0}}^8 1$ és $\text{DbV} = \sum_{\substack{i,j=1 \\ \text{VégÁllás}_{i,j} > 0}}^8 1$

Változó
 i,j :Egész



Mátrix – Algoritmusban

Deklaráció₂:

- Bemenet: $N, M \in \mathbb{N}$, $\text{Készlet}_{1..N, 1..M} \in \mathbb{N}^{N \times M}$
 $\text{Ár}_{1..M} \in \mathbb{N}^M$
- Kimenet: $\text{ÖsszÉrték}_{1..N} \in \mathbb{N}^N$

Változó

N, M : **Egész**

Készlet : **Tömb** $[1.. \text{MaxN}, 1.. \text{MaxM}]$: **Egész**

Ár : **Tömb** $[1.. \text{MaxM}]$: **Egész**

ÖsszÉrték : **Tömb** $[1.. \text{MaxN}]$: **Egész**

Mátrix-elemre hivatkozás:

$\text{TömbNév}[\text{sorIndex}, \text{oszlopIndex}]$

Itt: $\text{Készlet}[i, j]$

Programparaméterek
deklarálása



Mátrix – Algoritmusban

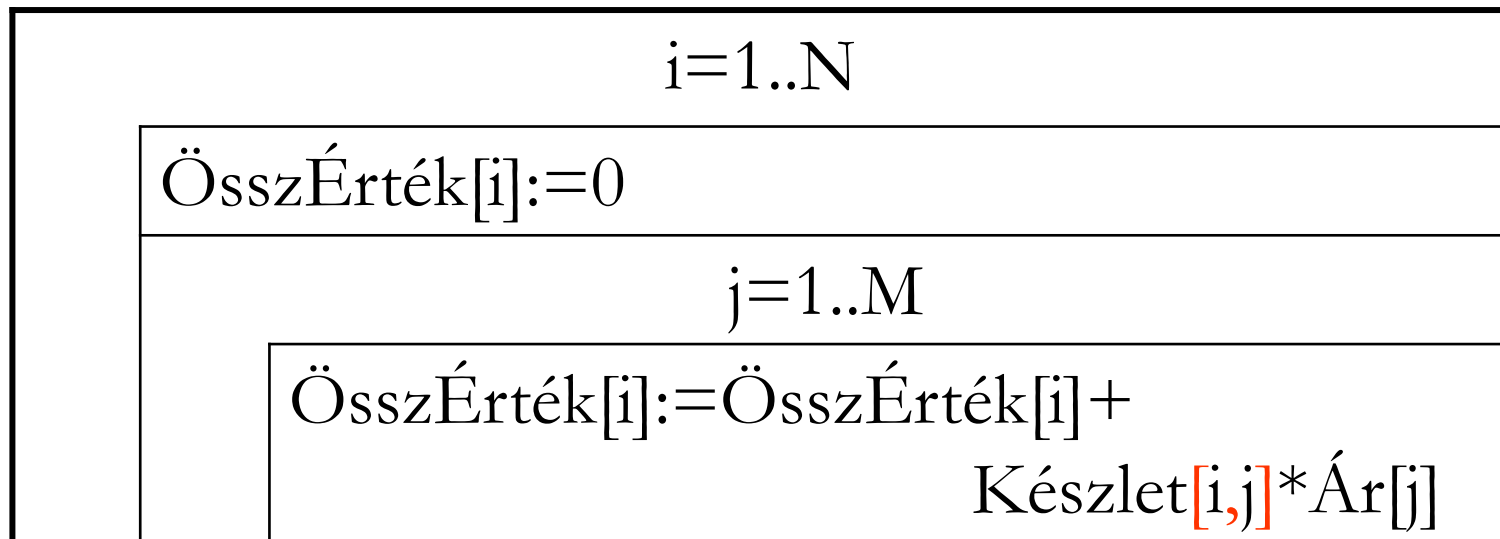
Algoritmus₂:

➤ Utófeltétel: $\forall i \in [1..N]$:

$$\text{ÖsszÉrték}_i = \sum_{j=1}^M \text{Készlet}_{i,j} * \text{Ár}_j$$

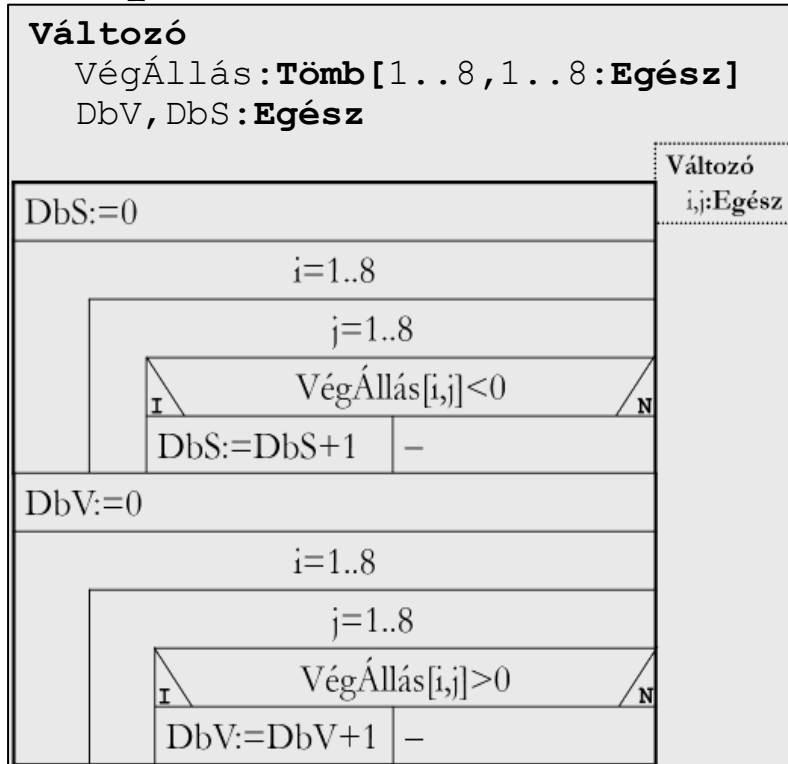
Lokális
változók
deklarációja

Változó
 i, j : Egész



Mátrix – Kódban

Kód₁:



```

int VegAllas[8+1][8+1];
int DbV, DbS;

DbS=0;
for(int i=1; i<=8; ++i) {
    for(int j=1; j<=8; ++j) {
        if (VegAllas[i][j]<0 then
            DbS++;
        }
    }
DbV=0;
for(int i=1; i<=8; ++i) {
    for(int j=1; j<=8; ++j) {
        if (VegAllas[i][j]>0 then
            DbV++;
        }
    }
}

```

Mátrix-elemre hivatkozás:

TömbNév[sorIndex][oszlopIndex]

Itt: VegAllas[i][j]



Mátrix – Kódban

Kód₂:

Változó

N, M : **Egész**

Készlet: **Tömb** $[1..MaxN, 1..MaxM]$: **Egész**

Ár: **Tömb** $[1..MaxM]$: **Egész**

ÖsszÉrték: **Tömb** $[1..MaxN]$: **Egész**

Változó
 i, j : **Egész**

$i = 1..N$

ÖsszÉrték $[i] := 0$

$j = 1..M$

ÖsszÉrték $[i] :=$ ÖsszÉrték $[i] +$
Készlet $[i, j] * \text{Ár}[j]$

```
int N, M;
int Keszlet[MaxN][MaxM];
int Ar[MaxM];
int OsszErtek[MaxN];
```

```
for(int i=0; i<N; ++i) {
    OsszErtek[i]=0;
    for(int j=0; j<M; ++j) {
        OsszErtek[i] += Keszlet[i][j] * Ar[j];
    }
}
```

Mátrix-elemre hivatkozás:

$\text{TömbNév}[\text{sorIndex}][\text{oszlopIndex}]$

Itt: Keszlet $[i][j]$



Áttekintés



- Ciklusok –
specifikáció+„algoritmika”+kódolás
- Tömbök
 - Egy bevezető példa a tömbhöz
 - A tömb
 - Elágazás helyett tömb
 - Konstans tömbök
 - Mátrixok

