Programozás
2. előadás

Tartalom

Ciklusok

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∈ℕ
- Kimenet: O∈ℕ
- Előfeltétel: N>1
- Utófeltétel: 1<O≤N és O|N és
  ∀i (2≤i<O): i∤N

Változó
N:**Egész**
O:**Egész**

Programváltozók deklarálása

Reprezentációs „szabály" a specifikáció→reprezentáció áttéréskor:

ℕ → **Egész**

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 $\boxed{\forall i\ (2\leq i<O):\ i \nmid N}$

Változó
i:Egész

i:=2
$i \nmid N$
i:=i+1
O:=i

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

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:

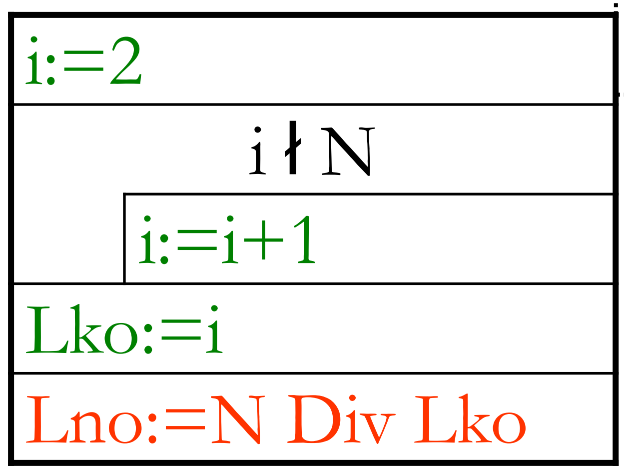
A specifikációból az algoritmus megkapható, de az Lno az utófeltételben az Lko ismeretében másképp is megfogalmazható: Lko*Lno=N!

Az erre építő algoritmus:

Specifikáció:

- Bemenet: N∈N
- Kimenet: Lko,Lno∈N
- Előfeltétel: N>1
- Utófeltétel: 1<Lko≤N és
  Lko|N és
  ∀i (2≤i<Lko): i∤N és
  Lko*Lno=N

Változó
i:**Egész**



Ciklusok

Feladat:

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

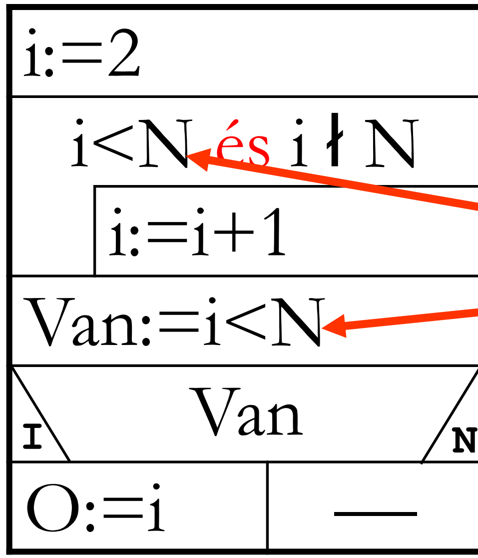
Specifikáció:

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

Ciklusok

Algoritmus:

Specifikáció:
- Bemenet: N∈**N**
- Kimenet: O∈**N**, Van∈𝕃
- Előfeltétel: N>1
- Utófeltétel: Van=∃i (2≤i<N): i|N és
  Van→2≤O<N és O|N és
  ∀i (2≤i<O): i∤N



Változó
i:Egész

$\mathbf{i} \le \sqrt{N}$

$2 \to i \le N \text{ Div } i \leftarrow N \text{ Div } 2$
azaz
$i*i \le N$
azaz
$i \le \sqrt{N}$

Megjegyzés:

Ha i osztója N-nek, akkor (N 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:

$N=15 \rightarrow \qquad \Sigma=$

$i=1 \;: (1\,|\,15) \rightarrow 1$

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

$i=3 \;: (3\,|\,15) \rightarrow 1+3$

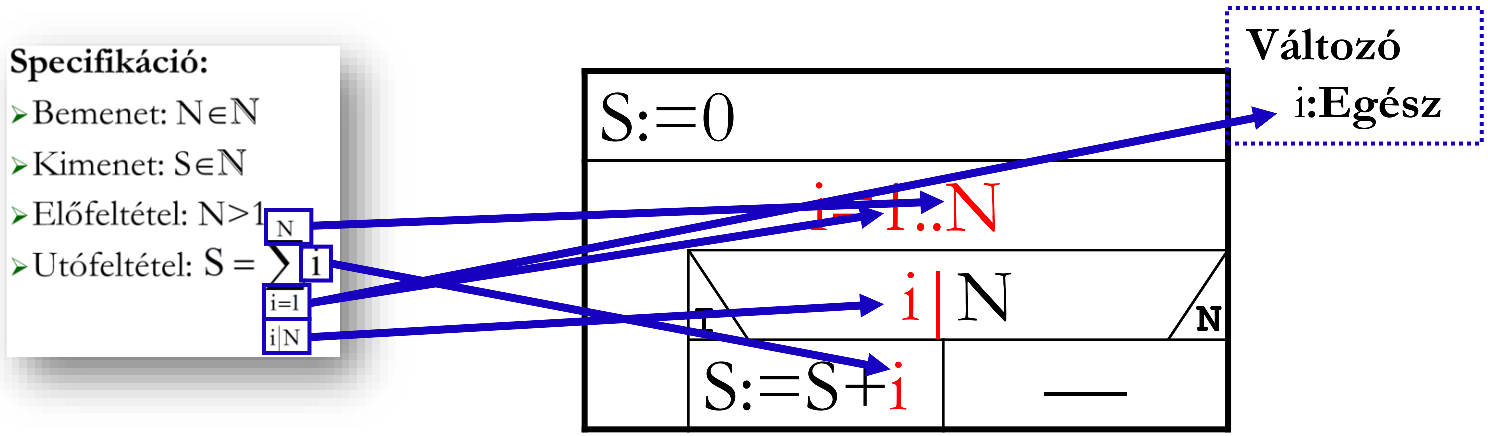
$i=4 \;: (4\nmid 15) \rightarrow 1+3$

…

$i=15: (15\,|\,15) \rightarrow 1+3+\ldots+15$

Ciklusok

Algoritmus:



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 Div i)` értékadással?

Ciklusok

Feladat:

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

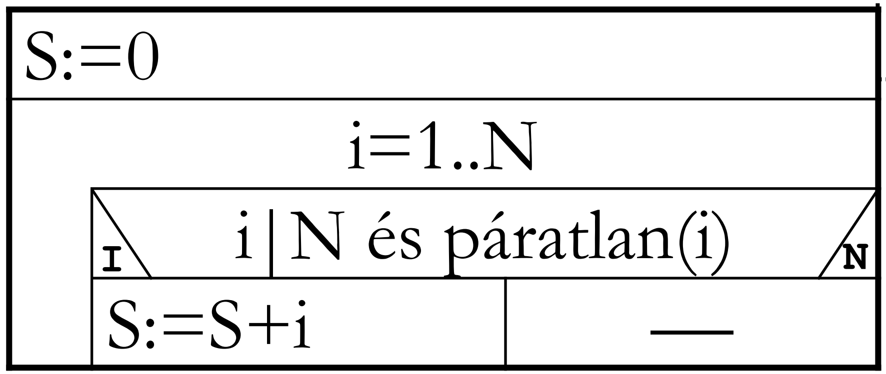
Specifikáció:

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

Ciklusok

Algoritmus$_1$:

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



Változó
i:Egész

Algoritmus$_2$:

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: $S = \sum\limits_{\substack{i=2 \\ i|N \text{ és } \text{prím}(i)}}^{N} i$

prím(i)=???

$$N=i_1^{m_1} * i_2^{m_2} * \ldots * i_k^{m_k}$$
$$\downarrow$$
$$S=i_1+i_2+\ldots+i_k$$

Algoritmus:

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

Specifikáció:

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

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

↓

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

Változó
i:Egész

```
S:=0
i:=2
i≤N
    i|N
    I: S:=S+i
       i|N
           N:=N Div i
    N: —
    i:=i+1
```

Miért nem számlálós a külső ciklus?

Ciklusok

Tanulságok:

- Ha az utófeltételben ∃, ∀, vagy Σ jel van, akkor a megoldás mindig ciklus!
- Ha az utófeltételben ∃ vagy ∀ 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:

feltétel
utasítások

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

Tipikus előfordulás: a beolvasás ellenőrzésénél

utasítások
feltétel

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

Számlálós ciklus:

i=1..N
utasítások

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

i=1..N; x-esével
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≤M≤2043), hogy milyen színű!

Feladat elágazásra,
vagy más megoldás kell?

Specifikáció$_1$:

- Bemenet: év$\in\mathbb{N}$
- Kimenet: s$\in$**Szín**

  Szín$=\{$"zöld","piros","sárga", "fehér","fekete"$\}\subset\mathbb{S}$
- Előfeltétel: 1984$\leq$év és év$\leq$2043
- Utófeltétel: ((év–1984) Mod 10) Div 2=0 és s="zöld" vagy ((év–1984) Mod 10) Div 2=1 és s="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?

Specifikáció$_2$:

- Bemenet: $év \in \mathbb{N}$
- Kimenet: $s \in$ **Szín**

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

A Szín halmaz definiálása.

- Előfeltétel: $1984 \le év$ és $év \le 2043$
- Utófeltétel: (((év–1984) Mod 10) Div 2=0 → s="zöld") és (((év–1984) Mod 10) Div 2=1 → s="piros") és …

- 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 deklarálása

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"

Specifikáció$_2$:
- 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 …

Kérdés:

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

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

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 ≅ (HónapHosszak_1, … , HónapHosszak_{12})
  - $\text{Emeletek}_{-1..10} \in \mathbb{S}^{12}$ – az Emeletek 12 elemű, szövegeket tartalmazó sorozat ≅ (Emeletek_{-1}, Emeletek_0, …, Emeletek_{10})=("Pince","Földszint",…)
- **Kérdés**: az elemek lehetnek sorozatok, azaz van-e sorozatok sorozata?

Tömbök

Algoritmikus fogalmak:

- Tömb: véges hosszúságú sorozat algoritmikus párja, amely-nek i-edik tagjával végezhetünk műveleteket (adott a legki-sebb 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≤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 → Tömbök

Példa$_1$:

Specifikációban:

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

Utófeltétel: $Z_1 = X_1 + Y_1$ … – hivatkozási példa

Algoritmusban:

X,Y,Z:**Tömb[**1..N**:**Valós**]** – deklarációs példa

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

Sorozatok → Tömbök

Példa$_2$:

Specifikációban:

Bemenet: Emeletek$_{-1..10}$∈$\mathbb{S}^{12}$ – deklarációs példa

Utófeltétel: Emeletek$_{-1}$="Pince" … – hivatkozási példa

Algoritmusban:

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

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

Sorozatok → Tömbök

Példa$_3$:

Az előbbi feladatpélda Szín halmaza a specifikációban egy szöveg konstansokból álló sorozattal ábrázolható:

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

("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: év∈N
- Kimenet: s∈S
  $Színek_{0..4}$∈S^5=
  ("zöld","piros","sárga","fehér","fekete")
- Előfeltétel: 1984≤év és év≤2043
- Utófeltétel: $\boxed{s=Színek_{(((év–1984)\ Mod\ 10)\ Div\ 2)}}$

A Szín halmaz reprezentálása.

Specifikáció$_2$:

- 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 és
  s="zöld" vagy
  ((év-1984) Mod 10) Div 2=1 és
  s="piros" vagy …

A Szín halmaz reprezentálciójához igazított utófeltétel.

Elágazás helyett tömb

Programváltozók deklarálása

- **Adatreprezentálás:**

Változó

    év:**Egész**

    s:**Szöveg**

Konstans

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

- Bemenet: $év \in N$
- Kimenet: $s \in S$
  $Színek_{0..4} \in S^5$=
  ("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:=Színek[((év–1984) Mod 10) Div 2]
```

- Bemenet: év∈N
- Kimenet: s∈S

  $Színek_{0..4} \in S^5$=
  ("zöld","piros","sárga","fehér","fekete")
- Előfeltétel: 1984≤év és év≤2043
- **Utófeltétel: s=Színek**$_{((év–1984)\ \underline{Mod}\ 10)\ \underline{Div}\ 2)}$

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 – a C# kódjukkal:

X:**Tömb[**1..N:Valós**]**

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

E:**Tömb[**-1..10:Szöveg**]**

```
string[] E=new string[12]
```

Az előbbi Szín halmazos példa:

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

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

Tömbök
(Algoritmus→kód)

Kódolási kérdések (`x[0]`-t **nem** használjuk)$_1$:

Algoritmus

Változó
i:**Egész**

```
i=1..n
    x[i]:=i
```

Legegyszerűbb kódolása C#-ban

```
int[] x = new int[n+1];
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**

```
i=1..n
    x[i]:=i
```

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..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

```
i=-1..n
    x[i]:=i+5
```

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;
}
```

Konstans tömbök alkalmazása

Feladat:

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

Specifikáció:

- Bemenet: $N \in \mathbb{N}$

  $egyes_{0..9} \in \mathbb{S}^{10} = ("", "egy", \ldots, "kilenc")$

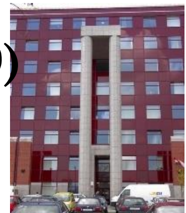
  $tizes_{0..9} \in \mathbb{S}^{10} = ("", "tizen", \ldots, "kilencven")$

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

- Kimenet: $S \in \mathbb{S}$
- Előfeltétel: $1 \le N \le 99$
- Utófeltétel: $N=10 \rightarrow S="tíz"$ és

  $N=20 \rightarrow S="húsz"$ és

  $N \notin \{10,20\} \rightarrow S=tizes_{(N\ Div\ 10)}+egyes_{(N\ Mod\ 10)}$



Konstans tömbök alkalmazása

Programváltozók 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 $\in \mathbb{S}^{10}$= ("","egy",…,"kilenc")
  - tizes $\in \mathbb{S}^{10}$= ("","tizen",…,"kilencven")
- Kimenet: $S \in \mathbb{S}$

- Utófeltétel:
  - N=10 → S="tíz" és
  - N=20 → S="húsz" és
  - $N \notin \{10,20\} \rightarrow S = tizes_{(N \text{ Div } 10)+1} + egyes_{(N \text{ Mod } 10)+1}$

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

Konstans tömbök alkalmazása

Feladat:

Írj programot, amely egy hónapnévhez a sorszámát rendeli!

Specifikáció:

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

Konstans tömbök alkalmazása

Programváltozók deklarálása

Algoritmus:

Változó H:**Szöveg**, S:**Egész**
Konstans HóNév:**Tömb**[1..12:**Szöveg**]=
("január",…,"december")

Specifikáció:
- Bemenet: H∈S
  $HóNév_{1..12}$∈S^{12}=("január",…
- Kimenet: S∈N
- Előfeltétel: H∈HóNév
- Utófeltétel: 1≤S≤12 és $HóNév_S$=H

S:=1
HóNév[S]≠H
S:=S+1

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

Konstans tömb – mit tárolunk?

Feladat:

Egy nap a nem szökőév hányadik napja?

Specifikáció$_1$:

- Bemenet: $H,N \in \mathbb{N}$
  $hó_{1..12} \in \mathbb{N}^{12} = (31,28,31,\ldots,31)$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $1 \leq H \leq 12$ és $1 \leq N \leq hó_H$
- Utófeltétel: $S = N + \sum_{i=1}^{H-1} hó_i$

Konstans tömb – mit tárolunk?

Programváltozók deklarálása

Algoritmus:

Változó H,N,S:**Egész**
Konstans hó:**Tömb**[1..12:**Egész**]=
(31,28,31,…,31)

Specifikáció$_1$:
- Bemenet: $H,N \in \mathbb{N}$
  $hó_{1..12} \in \mathbb{N}^{12}=(31,28,31,$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $1 \le H \le 12$ és $1 \le N \le hó_{H}$
- Utófeltétel: $S = N + \sum_{i=1}^{H-1} hó_i$

Változó
i:**Egész**

S:=N
i=1..H–1
S:=S+hó[i]

Lokális változó deklarálása

Megjegyzés: szökőév esetén H≥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!

Specifikáció$_2$:

- Bemenet: …
  $hó_{1..12} \in \mathbb{N}^{12} = (0,31,59,90,…,334)$
- Utófeltétel: $S = hó_H + N$

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.

Példa$_1$:

Egy sakkjátszma állása.

Az alaphalmaz:

{♔, ♕, ♖, ♗, ♘, ♙,
♚, ♛, ♜, ♝, ♞, ♟, Üres}

Mátrix

Példa$_2$:

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

Az alaphalmaz: 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

Feladat$_1$:

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

Feladat$_2$:

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 elemére szokásos módon – alul-indexeléssel – hivatkozhatunk: $M_{i,j}$.
- Index: $i \in 1..SorSzám$, $j \in 1..OszlopSzám$.

Mátrix – Specifikációban

Feladat$_1$:

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

Specifikáció$_1$:

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

Mátrix – Specifikációban

Feladat$_2$:
Határozzuk meg az egyes áruházakban tárolt készlet összértékét, ha ismerjük az egyes termékek árát!

Specifikáció$_2$:

- Bemenet: $N,M \in \mathbb{N}$, $Készlet_{1..N,1..M} \in \mathbb{N}^{N \times M}$
  $Ár_{1..M} \in \mathbb{N}^{M}$
- Kimenet: $ÖsszÉrték_{1..N} \in \mathbb{N}^{N}$
- Előfeltétel: –
- Utófeltétel: $\forall i \in [1..N]$: $ÖsszÉrték_i = \sum_{j=1}^{M} Készlet_{i,j} * Ár_j$

Mátrix – Algoritmusban

Deklaráció$_1$:

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

Programváltozók deklarálása

```
Változó
  VégÁllás:Tömb[1..8,1..8:Egész]
  DbV,DbS: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

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

Algoritmus$_1$:

Változó
i,j:**Egész**

```
DbS:=0
i=1..8
  j=1..8
    VégÁllás[i,j]<0
      I: DbS:=DbS+1
      N: –
DbV:=0
i=1..8
  j=1..8
    VégÁllás[i,j]>0
      I: DbV:=DbV+1
      N: –
```

Mátrix – Algoritmusban

Deklaráció$_2$:

- Bemenet: $N,M \in \mathbb{N}$, $Készlet_{1..N,1..M} \in \mathbb{N}^{N \times M}$
  $Ár_{1..M} \in \mathbb{N}^{M}$
- Kimenet: $ÖsszÉrték_{1..N} \in \mathbb{N}^{N}$

```
Változó
  N,M:Egész
  Készlet:Tömb[1..MaxN,1..MaxM:Egész]
  Ár:Tömb[1..MaxM:Egész]
  ÖsszÉrték:Tömb[1..MaxN:Egész]
```

Programváltozók deklarálása

Mátrix-elemre hivatkozás:

TömbNév[*sorIndex*,*oszlopIndex*]

Itt: Készlet[i,j]

Mátrix – Algoritmusban

Algoritmus$_2$:

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

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

Lokális változók deklarálása

Változó
i,j**:Egész**

i=1..N

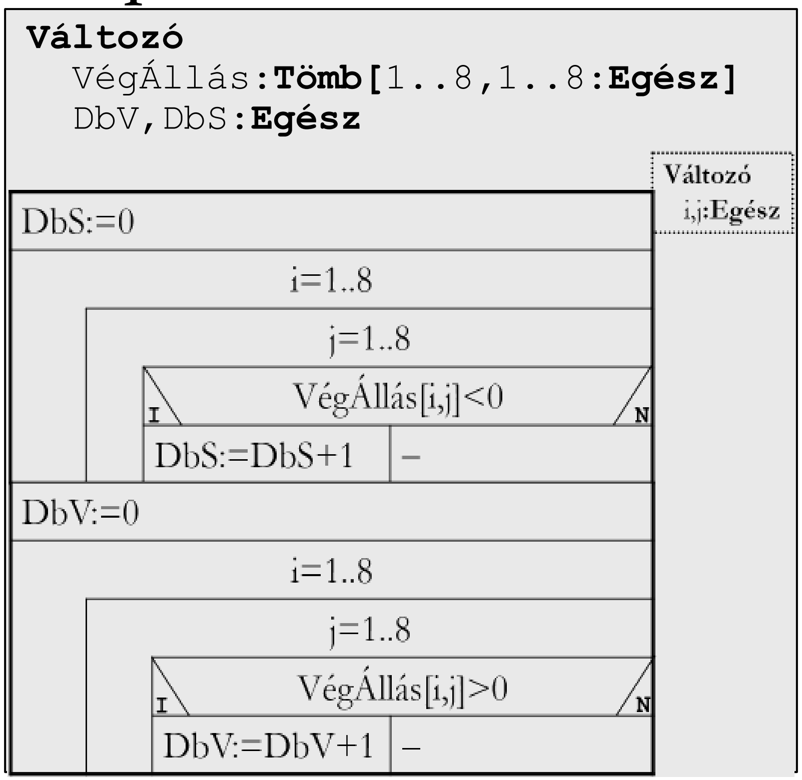
ÖsszÉrték[i]:=0

j=1..M

ÖsszÉrték[i]:=ÖsszÉrték[i]+
Készlet[i,j]*Ár[j]

Mátrix – Kódban

Kód$_1$:




```
int[,] VégÁllás=new int[8+1,8+1];
int DbV,DbS;

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

Mátrix-elemre hivatkozás:

TömbNév[*sorIndex*,*oszlopIndex*]

Itt: VégÁllás[i,j]

Mátrix – Kódban

Kód$_2$:

```
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]*Ár[j]
```

```
int N,M;
int[,] Készlet=new int[MaxN,MaxM];
int[] Ár=new int[MaxM];
int[] ÖsszÉrték=new int[MaxN];

for(int i=0;i<N;++i){
  ÖsszÉrték[i]=0;
  for(int j=0;j<M;++j){
    ÖsszÉrték[i]+=Készlet[i,j]*Ár[j];
  }
}
```

Mátrix-elemre hivatkozás:

TömbNév[*sorIndex*,*oszlopIndex*]

Itt: Készlet[i,j]

Áttekintés

- Ciklusok –

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