

The background of the slide is a black and white aerial photograph of Budapest, Hungary. It shows the city's layout along the Danube River, with the Buda Castle and its surrounding hills on the left and the Pest side on the right. A semi-transparent white rectangle is centered over the image, containing the title text.

Programozás

2. előadás

Tartalom

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



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

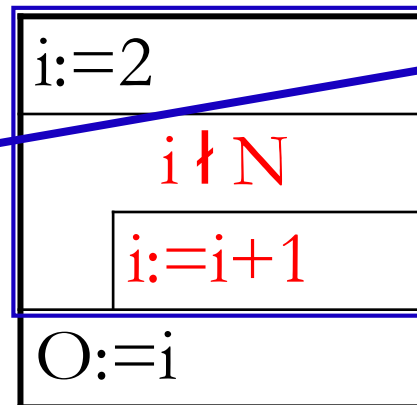
Próbáljuk ki a 2-t; ha nem jó, akkor a 3-at, ha az sem, akkor a 4-et, ...; legkésőbb az N jó lesz!

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: $Lko, Lno \in \mathbb{N}$
- 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

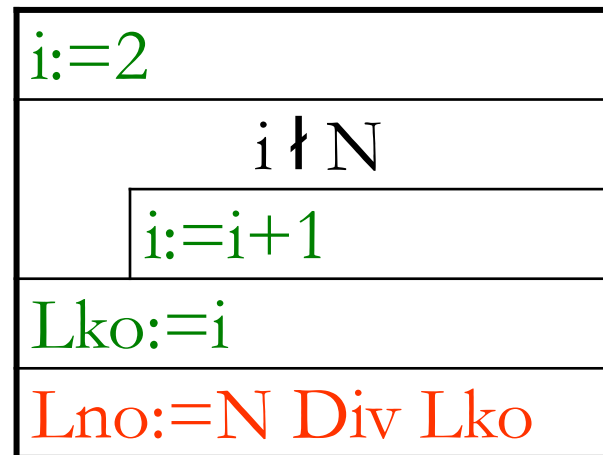
Megjegyzés:

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: $N \in \mathbb{N}$
- 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 $Lko * Lno = N$



Változó
 i : Egész



Ciklusok

Feladat:

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

Specifikáció:

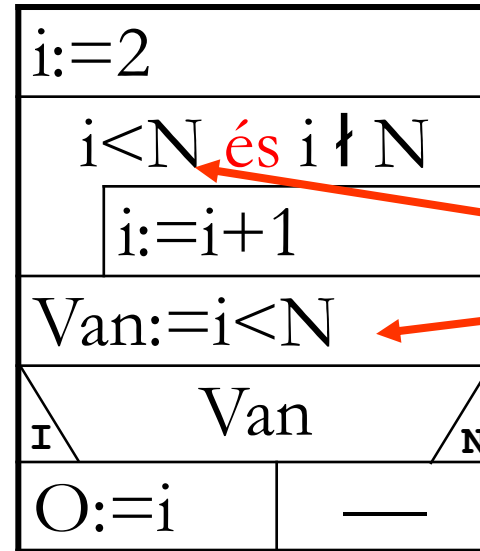
- Bemenet: $N \in \mathbf{N}$
- Kimenet: $O \in \mathbf{N}$, $Van \in \mathbf{L}$
- Előfeltétel: $N > 1$
- Utófeltétel: $Van = \exists i (2 \leq i < N) : i \mid N$ és
 $Van \rightarrow 2 \leq O < N$ és $O \mid N$ és $\forall i (2 \leq i < O) : i \nmid N$

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:

$N=15 \rightarrow$

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

$i=2 : (2 \nmid 15) \rightarrow S=1+0$

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

$i=4 : (4 \nmid 15) \rightarrow S=1+3+0$

...

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

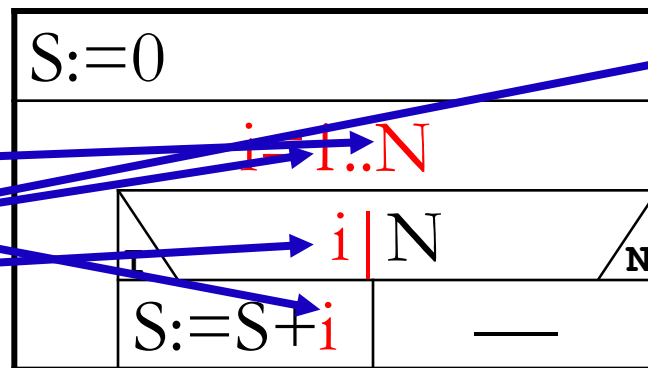


Ciklusok

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?



Feladat:

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

Specifikáció:

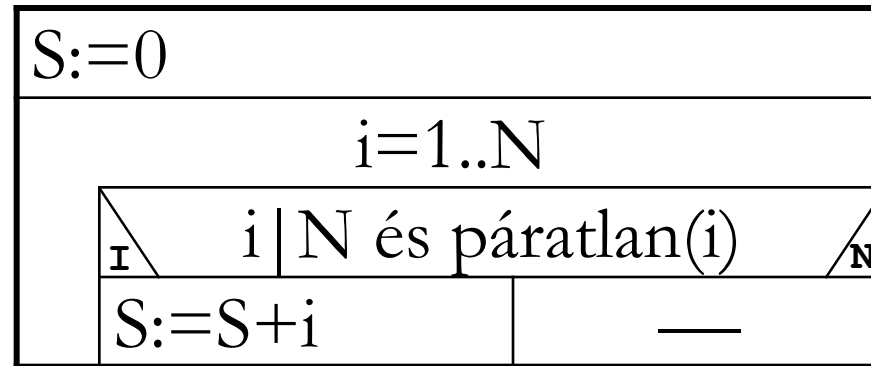
- Bemenet: $N \in \mathbb{N}$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $N > 1$
- Utófeltétel: $S = \sum_{\substack{i=1 \\ i|N \text{ és páratlan}(i)}}^N i$
- Definíció: páratlan ...

Ciklusok

Algoritmus₁:

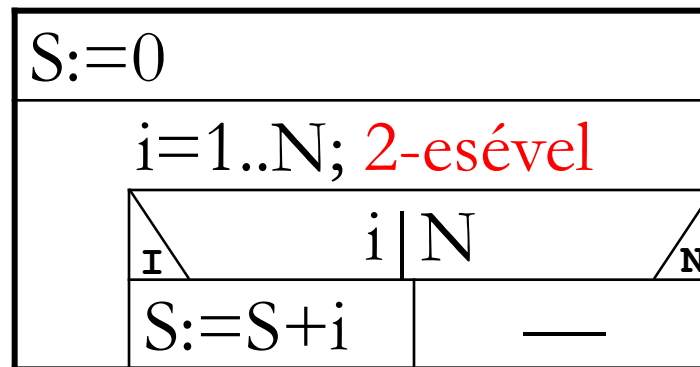
Specifikáció:

- 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: $S = \sum_{\substack{i=2 \\ i|N \text{ és } \text{prím}(i)}}^N i$
- Definíció: $\text{prím} \dots$

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



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.

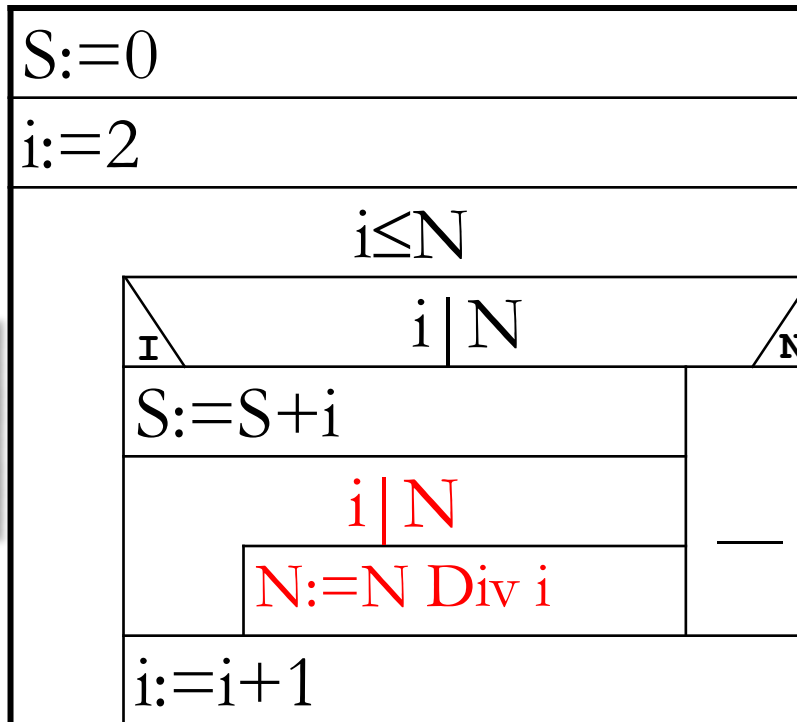
Specifikáció:

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

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

↓

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



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.



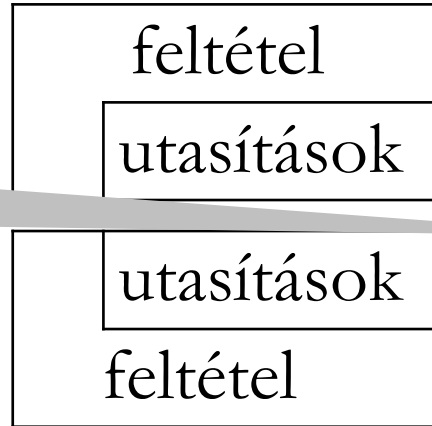
Ciklusok

algoritmus – kód



Feltételes ciklusok:

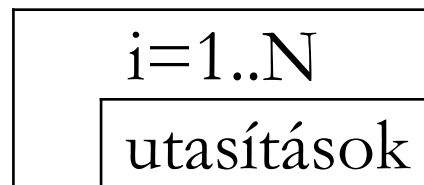
Tipikus előfordulás: a beolvasás ellenőrzésénél



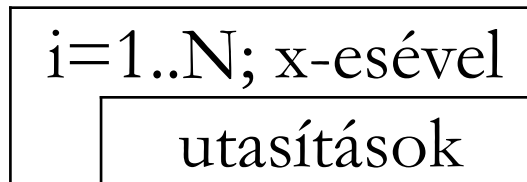
```
while (feltétel){  
    utasítások  
}
```

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

Számlálós ciklusok:



```
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: $s \in \mathbf{Szín}$
- 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 ...

Egy még
„definiálatlan”
halmaz

- 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

- Definíció:

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

A Szín halmaz
definiálása, visszavezetés a
Szöveg halmazra



Feladat elágazásra, vagy más megoldás kell?

A Szín halmaz
definiálása itt is
lehetséges

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



Feladat elágazásra, vagy más megoldás kell?

Specifikáció₂:

- Bemenet: $\text{év} \in \mathbb{N}$
- Kimenet: $s \in \mathbf{Szín}$
 $\mathbf{Szín} = \{ \text{"zöld"}, \text{"piros"}, \text{"sárga"}, \text{"fehér"}, \text{"fekete"} \} \subset \mathbb{S}$
- Előfeltétel: $1984 \leq \text{év}$ és $\text{év} \leq 2043$
- Utófeltétel: $((\text{év} - 1984) \bmod 10) \text{ Div } 2 = 0 \rightarrow$
 $s = \text{"zöld"} \text{ és}$
 $((\text{év} - 1984) \bmod 10) \text{ Div } 2 = 1 \rightarrow$
 $s = \text{"piros"} \text{ és } \dots$



Feladat elágazásra, vagy más megoldás kell?

Lokális változó
deklarálása

► Utófeltétel: (((év-1984) Mod 10) Div 2=0 →
s="zöld") és
(((év-1984) Mod 10) Div 2=1 →
s="piros") és ...

Algoritmus:

Változó
y:Egész

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**.



Sorozatok

Specifikációbeli fogalmak:

- Sorozat: **azonos halmazbeli elemek** egymásutánja, az elemei sorszámozhatók.
- Elem: a sorozat **i-edik elem**ére szokásos módon – alul-indexeléssel – hivatkozhatunk: S_i .
- Index: **1**..SorozatHossz.
- Például:
 - $\text{HónapHosszak} \in \mathbb{N}^{12}$ – a HónapHosszak 12 elemű, természetes számokból álló sorozat $\cong (\text{HónapHosszak}_1, \dots, \text{HónapHosszak}_{12})$
 - $\text{Évszakok} \in \mathbb{S}^4$ – az Évszakok 4 elemű, szövegeket tartalmazó sorozat $\cong (\text{Évszakok}_1, \text{Évszakok}_2, \text{Évszakok}_3, \text{Évszakok}_4)$
- Kérdés: az elemek lehetnek sorozatok, azaz van-e sorozatok sorozata?



Tömbök

Algoritmikus fogalmak:

- **Tömb:** véges hosszúságú sorozat (\rightarrow azonos típusú elemekből), a sorozat **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. hétfői ebéd, keddi ebéd, ...).
- **Tömbelem-műveletek:** **elemérték-hivatkozás, elemérték-módosítás** (az elem-indexeléssel kiválasztva).



Sorozatok \rightarrow Tömbök

Példa₁:

Specifikációban:

$X, Y, Z \in \mathbb{R}^N$ – deklarációs példa

$Z_1 = X_1 + Y_1$ – 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:

$Szk \in S^5$ – deklarációs példa

$Szk_1 = \text{"első szó"}$ – hivatkozási példa

Algoritmusban:

$Szk: \text{Tömb}[0..4: \text{Szöveg}]$ – deklarációs példa

$Szk[0] := \text{"első szó"}$ – hivatkozási példa



Sorozatok \rightarrow Tömbök

Példa₃:

Az előbbi feladat-példa Szín halmaza a specifikációban egy szöveg konstansokból álló **sorozat**:

Színek $\in \mathbb{S}^5 =$

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

Az algoritmusban reprezentálhatjuk így:

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

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

A lényeg az azonos elemszám!

Ügyelni kell az „indexek” konverziójára! Itt:

Színek_{*i*} \rightarrow Színek[***i*-1**]



Elágazás helyett tömb

Specifikáció:

➤ Bemenet: $\text{év} \in \mathbb{N}$

➤ Kimenet: $s \in \mathbf{Szín}$

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

$\mathbf{Színek} \in \mathbf{Szín}^5 = (\text{"zöld"}, \text{"piros"}, \text{"sárga"}, \text{"fehér"}, \text{"fekete"})$

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

➤ Utófeltétel: $s = \mathbf{Színek}_{(((\text{év} - 1984) \bmod 10) \text{ Div } 2) + 1}$

Specifikáció₂:

➤ Bemenet: $\text{év} \in \mathbb{N}$

➤ Kimenet: $s \in \mathbf{Szín}$

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

➤ 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 ...



Elágazás helyett tömb

Specifikáció (egyszerűsítve):

- Bemenet: $\text{év} \in \mathbb{N}$
- Kimenet: $s \in S$

$\text{Színek} \in S^5 = ("zöld", "piros", "sárga", "fehér", "fekete")$

- Előfeltétel: $1984 \leq \text{év} \leq 2043$
- Utófeltétel: $s = \text{Színek}_{((\text{év} - 1984) \bmod 10) \text{ Div } 2) + 1}$

Specifikáció₂:

- Bemenet: $\text{év} \in \mathbb{N}$
- Kimenet: $s \in \text{Szín}$
 $\text{Szín} = {"zöld", "piros", "sárga", "fehér", "fekete"} \subset S$
- Előfeltétel: $1984 \leq \text{év} \leq 2043$
- Utófeltétel: $((\text{év} - 1984) \bmod 10) \text{ Div } 2 = 0$ és $s = "zöld"$ vagy $((\text{év} - 1984) \bmod 10) \text{ Div } 2 = 1$ és $s = "piros"$ vagy ...



Elágazás helyett tömb

Programparaméterek
deklarációja

➤ Adatreprezentálás:

Változó

év:Egész

s:Szöveg

Konstans

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

Specifikáció (egyszerűsítve):

➤ Bemenet: év ∈ ℕ

➤ Kimenet: s ∈ S

Színek ∈ S⁵ =

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



Adatreprezentálás:

Változó

év:Egész

s:Szöveg

Konstans

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

Elágazás helyett tömb

Algoritmus:

Tevékenység:

Specifikáció (egyszerűsítve):

➤ Bemenet: év ∈ ℕ

➤ Kimenet: s ∈ S

Színek ∈ S⁵ =

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

➤ Előfeltétel: 1984 ≤ év és év ≤ 2043

➤ Utófeltétel: s = Színek_{(((év-1984) Mod 10) Div 2)+1}

$$y := (((\text{év} - 1984) \text{ Mod } 10) \text{ Div } 2) + 1$$

$$s := \text{Színek}[y-1]$$

Változó

y:Egész

észrevéve az egyszerűsítési lehetőséget:

$$y := ((\text{év} - 1984) \text{ Mod } 10) \text{ Div } 2$$

$$s := \text{Színek}[y]$$

Változó

y:Egész



Tömbök

(Algoritmus→kód)



Tömb-**elemszám** ⇒
indexelés 0..N

A 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]

Y:Tömb[**0**..4:Szöveg]

Az előbbi Szín halmazos példa:

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

a C++ kódjukkal:

```
float X[N1]
```

```
string Y[5]
```

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

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



Tömbök

(C++ kódban – áttekintés)

Fordításkor kiderülő méret esete.

Statikus tömbök:

➤ Deklaráció:

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

Eltérés a specifikációban szokásostól!

➤ Hivatkozások:

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



Tömbök

(C++ kódban – áttekintés)

Fordításkor kiderülő méret esete.

Statikus tömb konstansok:

➤ Deklaráció:

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

vagy

```
const típus tömb[]={t1,t2,...,tN};
//konstans tömb deklarációja
const int N=sizeof(tömb)/sizeof(típus);
//tömb elemszáma,
//indexek: 0..N-1 közötti
```



Tömbök

(C++ kódban – áttekintés)

Csak futáskor kiderülő méret esete.

Dinamikus tömbök₁:

➤ Deklaráció:

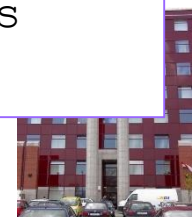
```
int N;           //tömb aktuális elemszám  
...
```

➤ 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  
...
```



Tömbök

(C++ kódban – áttekintés)

Csak futáskor kiderülő méret esete.

Dinamikus tömbök₂:

➤ Deklaráció:

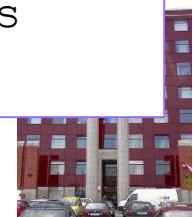
```
int N;           //tömb aktuális elemszám  
...
```

➤ Létrehozás:

```
N=???; //N meghatározása, pl. beolvasása  
típ* tömb=new típ[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...

$\text{egy} \in S^{10} = ("", "egy", \dots, "kilenc")$

$\text{tíz} \in S^{10} = ("", "tizen", \dots, "kilencven")$

➤ Kimenet: $S \in \mathbb{S}$

➤ Előfeltétel: $1 \leq N \leq 99$

➤ Utófeltétel: $N=10 \rightarrow S="tíz"$ és $N=20 \rightarrow S="húsz"$ és

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



Konstans tömbök alkalmazása

Programparaméterek
deklarációja

Algoritmus:

Változó N:Egész

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

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

Változó S:Szöveg

Specifikáció:

➤ **Bemenet:** $N \in \mathbb{N}$

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

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

➤ **Kimenet:** $S \in S$

➤ **Utófeltétel:**

$N=10 \rightarrow S=\text{"tíz"}$ és

$N=20 \rightarrow S=\text{"húsz"}$ és

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

Figyelembe véve az index-elcsúszást:

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



Konstans tömbök alkalmazása

Feladat:

Írj programot, amely egy hónapnévhez a sorszámát rendeli!

Specifikáció:

- Bemenet: $H \in S$
 $HóNév \in S^{12} = ("január", \dots, "december")$
- 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")

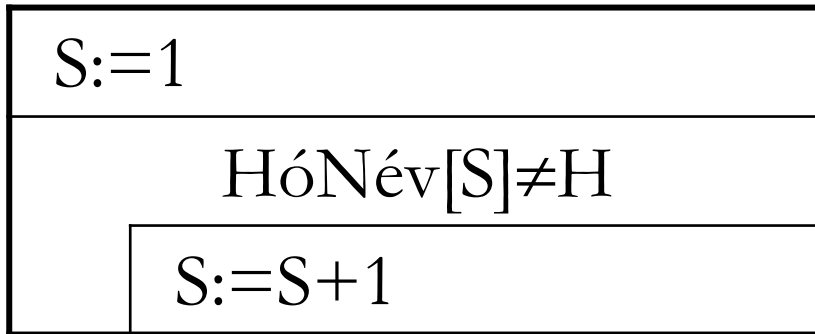
Specifikáció:

➤ Bemenet: $H \in S$

$HóNév \in S^{12} =$

("január",...,"december")

➤ Kimenet: $S \in N$



➤ 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: $H, N \in \mathbb{N}$
 $\text{hó} \in \mathbb{N}^{12} = (31, 28, 31, \dots, 31)$
- 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?

Algoritmus:

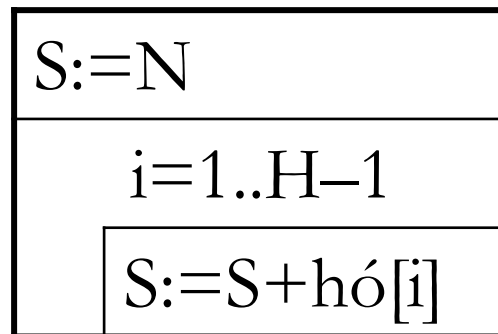
Programparaméterek
deklarálása

Specifikáció₁:

- Bemenet: $H, N \in \mathbb{N}$
 $hó \in \mathbb{N}^{12} = (31, 28, 31, \dots, 31)$
- Kimenet: $S \in \mathbb{N}$
- Előfeltétel: $1 \leq H \leq 12$ és $1 \leq N \leq hó_1$
- Utófeltétel: $S = N + \sum_{i=1}^{H-1} hó_i$

Változó H, N, S : Egész

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



**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: ...

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

➤ Utófeltétel: $S = \text{hó}_H + N$

Kérdés:

Ez jobb megoldás? Mi lesz az előfeltétellel?



Áttekintés



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

