

## 12. gyakorlat

### 1. Bemelegít? feladatok



Írjuk ki a következ? két *Beatty-sorozat* els? 12 tagjának sorozatát: a jelöléssel  $a_n = \lfloor n\phi \rfloor$ , illetve  $a_n = \lfloor n\phi^2 \rfloor$ .  $\phi = \frac{\sqrt{5} + 1}{2}$

- ◆ Keressük meg az összes olyan *Pitagorászi számhármast*, ahol a három szám összege 1000.
- ◆ Írjuk ki a barátságos számpárok listáját 10000-ig.

### 2. Nest

- ◆ Írjunk egy *Nest* nev? függvényt, mely a *Nest(f, x, n)* hívásra visszaadja az  $f(f(f(\dots f(x)\dots)))$  értéket, ahol  $f$   $n$ -szer szerepel.
- ◆ Keressünk numerikus megoldást a  $\cos(x)=x$  egyenletre!
- ◆ Írjunk egy *NestList* nev? függvényt, mely a *NestList(f, x, n)* hívásra visszaadja az  $[x, f(x), f(f(x)), \dots, f(f(f(\dots f(x)\dots)))]$  listát.
- ◆ Tudjuk, hogy az  $[1, 1, 1, \dots]$  *végtelen lánc tört* értéke az *arany metszés* arányát adó

$$\phi = \frac{\sqrt{5} + 1}{2}$$

. Számítsuk ki a lánc tört els? néhány szeletének értékét!

### 3. Fold

- ◆ Írjunk *Fold* nev? programot, mely a *Fold(f, x, [a, b, c, d])* hívásra az  $f(f(f(f(x,a),b),c),d)$  értéket adja vissza.
- ◆ Számoljuk ki a *Horner-módszerrel* az  $x^3+2x^2+3x+4$  polinom értékét a -2 helyen!

### 4. Generátorfüggvény, bejáró

- ◆ Írjunk programot a *Collatz-problémára* (más néven  $3x+1$ ), és írjuk meg annak generátorfüggvényes változatát (ami a *next* hívására a következ? számot adja)!
- ◆ Programozzuk be azt az *újraírási rendszert*, mely a  $0 \rightarrow 01$ ,  $1 \rightarrow 0$  helyettesítéseket végzi a 0-ból indulva. Vegyük észre, hogy a 0-k és 1-ek sorszámai a Beatty-sorozat elemei.
- ◆ Olyan karakterláncot keresünk, amely egy  $n$ -elem? ábécéb?l készült, maximális hosszú, és *nincs benne két egymást követ? azonos részlánc*. Be van bizonyítva, hogy 2-elem? ábécé esetén e maximum 3, 3-elem? ábécé esetén viszont van ilyen végtelen hosszú sorozat is, melyet a  $A \rightarrow ABC$ ,  $B \rightarrow AC$ ,  $C \rightarrow B$  helyettesítésekkel megkaphatunk az 'A' karakterláncból indulva. Írjunk e karakterláncot adó generátort!

### 5. SUDOKU

- ◆ Írjunk programot, mely minden SUDOKU-feladványt megold. A program inputja egy fájlban legyen megadva, ahol 9 sorban 9-hosszú karakterláncok szerepeljenek az alábbi minták szerint. Az output a megoldás legyen, vagyis 9 sorban 9-hosszú karakterláncok, melyekben csak az 1-t?l 9-ig terjed? számok szerepelhetnek.
- ◆ Írjunk olyan SUDOKU-feladványt megoldó programot, mely csak azt az egyetlen technikát alkalmazza, hogy ha egy mez? oszlopában, sorában és blokkjában egyet kivéve minden szám el?fordul, akkor azt a hiányzó számot oda beírja, és ezt addig folytatja, míg van ilyen mez? a táblán.

Néhány feladvány:

```
5.2...8.6
...548...
7..3.2..4
.25.9.41.
.7.1.5.8.
.91.8.65.
2..8.6..9
...754...
4.3...7.5
```

.56...81.  
 2.1...6.5  
 43..5..97  
 ...543...  
 ..38.61..  
 ...219...  
 31..6..79  
 5.9...4.8  
 .64...32.

És néhány, ami csak a második technikával nem oldható meg:

51.3.4.72  
 2..5.9..1  
 ....2....  
 64.....89  
 ..2...1..  
 39.....65  
 ....4....  
 9..2.6..7  
 45.8.7.23

..2.3...8  
 .....8...  
 .31.2....  
 .6..5.27.  
 .1.....5.  
 2.4.6..31  
 ....8.6.5  
 .....13  
 ..531.4..

Azoknak, akik nem tudják mi a SUDOKU: A játék alapja egy  $9 \times 9 = 81$  mezű álló négyzet, mely 9 db  $3 \times 3$ -as blokkra van osztva. Induláskor a feladványban a 81 mezű némelyike ki van töltve a 1-től 9-ig terjedő számok valamelyikével. A feladat az, hogy töltsük ki a tábla minden mezűjét úgy, hogy minden sorban, minden oszlopban és minden blokkban az 1, 2, 3, 4, 5, 6, 7, 8, 9 számok mindegyike szerepeljen. Az előre megadott számok nem változtathatók meg. (Egy SUDOKU-feladvány akkor korrekt, ha annak pontosan egy megoldása létezik)

A fenti első feladat megoldása például:

542971836  
 936548271  
 718362594  
 825693417  
 674125983  
 391487652  
 257836149  
 169754328  
 483219765

Ugyanez a blokkokat is mutató áttekinthető kiírásban:

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 5 | 4 | 2 | 9 | 7 | 1 | 8 | 3 | 6 |
| 9 | 3 | 6 | 5 | 4 | 8 | 2 | 7 | 1 |
| 7 | 1 | 8 | 3 | 6 | 2 | 5 | 9 | 4 |
| 8 | 2 | 5 | 6 | 9 | 3 | 4 | 1 | 7 |
| 6 | 7 | 4 | 1 | 2 | 5 | 9 | 8 | 3 |
| 3 | 9 | 1 | 4 | 8 | 7 | 6 | 5 | 2 |

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 2 | 5 | 7 | 8 | 3 | 6 | 1 | 4 | 9 |
| 1 | 6 | 9 | 7 | 5 | 4 | 3 | 2 | 8 |
| 4 | 8 | 3 | 2 | 1 | 9 | 7 | 6 | 5 |