

# Felvételi tematika – INFORMATIKA

## FEJEZETEK

1. Természetes számok feldolgozása számjegyenként.
2. Számsorozatok feldolgozása elemenként. Egydimenziós tömbök.
3. Mátrixok feldolgozása elemenként/soronként/oszloponként. Kétdimenziós tömbök.
4. Egymásba ágyazott ciklusszerkezetek használatát feltételező algoritmusok.
5. Iteratív alprogramok (eljárások/függvények).
6. Elemi algoritmusok, és ezek alkalmazásai: keresés (lineáris/bináris), rendezés, prím-teszt, luko, lkkt, stb.
7. További egyszerű algoritmusok.

Példák ELMÉLETI tételre a fentebb megadott tematikából:

- ALPROGRAMOK (Pascal vagy C/C++): definíció/deklaráció; formális és aktuális/effektív paraméterek; érték/cím-szerinti paraméterátadás, stb. Adj példákat!
- TÖMBÖK (Pascal vagy C/C++): definíció; egy-, két-dimenziós tömbök; kezdőérték-adás definíciókor; tömb típusú paraméterek átadása alprogramoknak, stb. Adj példákat!

Példák GYAKORLATI feladatokra a fentebb megadott tematikából (fejezetenként):

1. Természetes számok feldolgozása számjegyenként.

A) Mit ír ki az alábbi pszeudokód program, ha a beolvasott érték (természetes szám) 20150328 (a „/” operátor osztási hányadost ad meg)? Mit valósít meg a program (fogalmazd meg tömören)?

```
beolvas n
amíg n > 9 végezd
    n = n / 10
kiír n;
```

B) Legyen az alábbi pszeudokód program (a beolvasott érték természetes szám; a „/” operátor osztási hányadost, a „%” operátor pedig osztási maradékot ad meg):

```
beolvas a
k1 = 0
k2 = 0
amíg a > 0 végezd
    ha a % 2 == 1 akkor
        k1 = k1 * 10 + a % 10
    különben
        k2 = k2 * 10 + a % 10
    a = a / 10
```

```
kiír k1, ",", k2
```

Mit ír ki a program, ha a beolvasott érték 2016?

- C) Írj Pascal vagy C/C++ programot, amely beolvas a billentyűzetről egy természetes számot ( $0..10^{10}$  tartományból), és kiírja a képernyőre a számjegyei átlagát.
  - D) Írj Pascal vagy C/C++ programot, amely beolvas a billentyűzetről egy természetes számot ( $0..10^{10}$  tartományból), és kiírja a képernyőre a szám legkisebb és legnagyobb számjegyeit.
  - E) Írj Pascal vagy C/C++ programot, amely beolvas a billentyűzetről egy természetes számot ( $0..10^{10}$  tartományból), és kiír a képernyőre egy számjegy-előfordulási statisztikát.
2. Számsorozatok feldolgozása elemenként. Egydimenziós tömbök.
- A) Legyen az alábbi pszeudokód programrészlet (az  $x[1..n+1]$  ( $n \geq 1$ ) tömb  $1..n$  szakasza egy növekvő sorrendbe rendezett természetes számsorozatot tárol, az  $n$  változó pedig ennek hosszát):

```
beolvas a
i = n
amíg i > 0 ÉS x[i] ≥ a végezd
  x[i+1] = x[i]
  i = i - 1
x[i+1] = a
minden i = 1, n+1 végezd
  kiír x[i], ",", "
```

Mit ír ki a program, ha  $n=5$ , a számsorozat elemei 13, 17, 23, 23, 53, a beolvasott érték pedig 15? Mit valósít meg a programrészlet (fogalmazd meg tömören)?

- B) Írj Pascal vagy C/C++ programot, amely természetes számokat ( $0..999$  tartományból) olvas be a billentyűzetről 0 végjelig, és kiírja a képernyőre ezek szorzatát.
  - C) Írj Pascal vagy C/C++ programot, amely beolvas a billentyűzetről egy  $n$  ( $1 \leq n \leq 999$ ) értéket, és kiírja a képernyőre az alábbi sorozat első  $n$  darab elemét: 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, ...
  - D) Írj Pascal vagy C/C++ programot, amely beolvas a billentyűzetről egy  $n$  ( $1 \leq n \leq 999$ ) értéket és egy  $n$  elemű egész számokat ( $-999..999$  tartományból) tartalmazó számsorozatot, és kiírja a képernyőre azt, hogy a számsorozat hány elemére igaz, hogy egyenlő a szomszédjai számtani közepével.
  - E) Írj Pascal vagy C/C++ programot, amely beolvas a billentyűzetről egy  $n$  ( $1 \leq n \leq 999$ ) értéket és egy  $n$  elemű egész számokat ( $-999..999$  tartományból) tartalmazó számsorozatot, és kiír a képernyőre egy megfelelő üzenetet aszerint, hogy a számsorozat palindrom-e (a szimmetrikus pozíciókban levő elemek azonosak).
3. Mátrixok feldolgozása elemenként/soronként/oszloponként. Kétdimenziós tömbök.
- 1) Mit ír ki az alábbi pszeudokód program, ha az  $n$  változó 4-et, az  $x[1..4][1..4]$  tömb pedig az 1,2,3,4; 2,1,5,51; 3,5,77,9; 44,5,9,1 értékeket tárolja, soronkénti bontásban?

```
minden i = 1, n végezd
  s = 0;
  minden j = 1, n végezd
    s = s + x[i][j]
  kiír s, ",", "
```

- 2) Legyen az alábbi pszeudokód programrészlet (az  $y[1..n][1..n]$  ( $n \geq 1$ ) tömb egy négyzetes mátrixot tárol, az  $n$  változó pedig ennek méretét):

```

j = n
amíg j > 0 végezd
  i = n
  amíg i > 0 végezd
    kiír y[i][j], ", "
    i = j - 1
  j = i - 1

```

Mit ír ki a program, ha  $n=4$ , a mátrix elemei pedig (soronkénti bontásban) 1, 2, 3, 4; 9, 9, 9, 9; 1, 2, 3, 4; 1, 2, 3, 4?

- 3) Írj Pascal vagy C/C++ programot, amely állományból beolvassa az  $n$  ( $1 \leq n \leq 10$ ) és  $m$  ( $1 \leq m \leq 10$ ) értékeket és egy  $n \times m$  méretű egész számokat (-999..999 tartományból) tartalmazó mátrixot, és kiírja a képernyőre a mátrixot, táblázatos alakban úgy, hogy minden páratlan sorszámú sor fordított sorrendben jelenjen meg.
  - 4) Írj Pascal vagy C/C++ programot, amely állományból beolvassa az  $n$  ( $1 \leq n \leq 10$ ) értékeket és egy  $n \times n$  méretű egész számokat (-999..999 tartományból) tartalmazó mátrixot, és kiír a képernyőre megfelelő üzeneteket aszerint, hogy a mátrix szimmetrikus-e a főátlóra nézve.
  - 5) Írj Pascal vagy C/C++ programot, amely állományból beolvassa az  $n$  ( $1 \leq n \leq 10$ ) és  $m$  ( $1 \leq m \leq 10$ ) értékeket és egy  $n \times m$  méretű egész számokat (-999..999 tartományból) tartalmazó mátrixot, és kiírja egy kimeneti állományba, hogy hány sor tartalmaz szigorúan növekvő sorrendbe rendezett elemeket.
4. Egymásba ágyazott ciklusszerkezetek használatát feltételező algoritmusok.

- A) Legyen az alábbi pszeudokód programrészlet (a beolvasott értékek természetes számok; a „/” operátor osztási hányadost, a „%” operátor pedig osztási maradékot ad meg):

```

beolvas n
minden i = 1, n végezd
  beolvas x
  amíg x > 9 végezd
    s = 0
    amíg x > 0 végezd
      s = s + x%10
      x = x / 10
    x = s
  kiír x, ", "

```

Mit ír ki a program, ha  $n$ -be az 5 értéket, majd az  $x$  változóba sorra az 139, 999, 89, 6, 11 értékeket olvassuk be?

- B) Írj Pascal vagy C/C++ programot, amely beolvas a billentyűzetről egy  $n$  ( $1 \leq n \leq 999$ ) értéket és egy  $n$  elemű természetes számokat ( $0..10^{10}$  tartományból) tartalmazó számsorozatot, és kiírja egy kimeneti állományba, minden szám első számjegyét.
- C) Írj Pascal vagy C/C++ programot, amely természetes számokat ( $0..10^{10}$  tartományból) olvas be a billentyűzetről 0 végjelig, és kiírja a képernyőre a számonkénti legkisebb számjegyek összegét.
- D) Írj Pascal vagy C/C++ programot, amely beolvas a billentyűzetről egy  $n$  ( $1 \leq n \leq 999$ ) értéket és egy  $n$  elemű természetes számokat ( $0..10^{10}$  tartományból) tartalmazó számsorozatot, és kiírja a képernyőre, hogy hány számra igaz, hogy prím és a tükörszáma is prím.

- E) Írj Pascal vagy C/C++ programot, amely állományból beolvassa az  $n$  ( $1 \leq n \leq 10$ ) értékeket és egy  $n \times n$  méretű egész számokat ( $-999..999$  tartományból) tartalmazó mátrixot, és kiírja, hogy hány főátlóra szimmetrikus pozícióban levő elem párra igaz, hogy relatív prímelek (legnagyobb közös osztójuk 1).
5. Iteratív alprogramok (eljárások/függvények).
- A) Írj eljárást, a 3/C feladat megoldása keretében, a mátrix egy adott sorának előre/fordított sorrendben való kiírására.
- B) Írj függvényt, a 4/B feladat megoldása keretében, egy természetes szám első számjegyének meghatározására.
- C) Írj függvényt, a 4/C feladat megoldása keretében, egy természetes szám legkisebb számjegyének meghatározására.
- D) Írj függvényt, a 4/D feladat megoldása keretében, a tükörszám előállításra, illetve a prímtesztre.
- E) Írj függvényt, a 4/E feladat megoldása keretében, két szám legnagyobb közös osztójának meghatározására.
6. Elemi algoritmusok, és ezek alkalmazásai.
- A) Összeg/szorzat/átlag-számítás.
- B) Lineáris/bináris-keresés, min/max-keresés, stb.
- C) Rendezési algoritmusok: buborékos, beszűrős, kiválasztásos.
- D) Prím-teszt, loko, lkk, stb.
- E) Írj függvényt (Pascal vagy C/C++ nyelven), amely visszatéríti egy keresett érték pozícióját egy növekvő sorrendbe rendezett számsorozatban. Tekintsd úgy, hogy a keresett értéket az  $x$  változó, a számsorozat hosszát az  $n$  változó, a számsorozatot pedig az  $a[1..n]$  tömb (Pascal változat) vagy az  $a[0..n-1]$  tömb (C/C++ változat) tárolja. Ha többször is előfordul az illető érték a számsorozatban (monoton növekvő), akkor az utolsó előfordulási pozícióban vagyunk érdekeltek. Ha nem szerepel a keresett érték a számsorozatban, akkor a függvény térítse vissza a  $-1$  értéket. Add meg azt is, hogy miként hívnád meg függvényedet. Törekedj hatékony megoldásra!
7. További egyszerű algoritmusok.
- A) Egy sajátos módszert definiálunk az  $\{1,2,3,\dots,n\}$  halmaz valamely permutációjának rendezésére, amelyben egy *lépés* abból áll, hogy a számsorozat valamely elemét a számsorozat végére tesszük. Írj függvényt (Pascal vagy C/C++ nyelven), amely visszatéríti a minimális *lépés*-számot, amellyel a paraméterként kapott permutáció (mint  $n$  elemű számsorozat) elemi növekvő sorrendbe rendezhetőek. Például a  $3,1,5,2,4$  permutáció/számsorozat esetén a minimális lépésszám 3. (Példa egy *tetszőleges* lépésre: az adott példában az 5-ös elemet a számsor végére tesszük; e lépés nyomán a számsor  $3,1,2,4,5$ -re változik)
- B) Adott  $n+1$  fa, amelyek 1-től  $(n+1)$ -ig vannak megszámozva. Az első  $n$  fa mindenikén elhelyezkedett egy-egy madár. Ezeket is megszámozzuk 1-től  $n$ -ig. Az  $i$ -edik fára az  $i$ -edik madár szállt ( $i = 1, n$ ). A madarak elkezdnek áthelyezkedni. Minden lépésben valamelyik madár átrepül az éppen üres fára (egyszerre csak egyetlen madár van a levegőben). Ismerve, hogy egy idő után melyik madár éppen melyik fára szállt, „repítsük vissza” a madarakat eredeti helyükre *minimális* számú repüléssel (egyszerre csak egyetlen madár van a levegőben).

# Minta tétel (INFORMATIKA)

(Munkaidő 3 óra; megjelenésre 1 pont jár)

## ELMÉLET (max 2 pont)

- CIKLUS-utasítások (Pascal vagy C/C++): **while**; **do-while** / **repeat-until**; **for** (szintaxisok; működésük; sajátosságok; stb; adj példákat)!

GYAKORLAT (a több alpontos feladatok esetében csak egyik alpontot kell választani; ha több alpontot is megoldana a felvételiző, akkor arra kapja meg a pontszámot, amelyik a legtöbbet éri)

- 1) **(maximum 1 pont)** Mit ír ki az alábbi pszeudokód program, ha az  $n$  változó 9-et, az  $x[1..9]$  tömb pedig az 11,7,32,4,5,4,3,2,1 értékeket tárolja? (a „%”operátor osztási maradékot ad meg)

```
k = 0
minden i = 1, n végezd
    ha i % 2 == 1 ÉS x[i] % 2 == 1 akkor
        k = k + 1
kiír k
```

- 2) **(maximum 1 pont)** Mit ír ki az alábbi pszeudokód program, ha az  $n$  változó 4-et, az  $x[1..4][1..4]$  tömb pedig az 1,2,3,4; 2,1,5,1; 3,5,7,1; 4,5,2,1 értékeket tárolja, soronkénti bontásban?

```
s = 0;
minden i = 1, n végezd
    p = 1;
    minden j = 1, n végezd
        s = s + x[i][j]
        p = p * x[i][j]
    kiír p, ", "
kiír s
```

- 3) **(maximum 1 pont)** Írj Pascal vagy C/C++ programot, amely beolvas a billentyűzetről egy természetes számot ( $0..10^{10}$  tartományból), és kiírja a képernyőre
- a beolvasott számot, a 10-zel való osztási hányadosát, és a 10-zel való osztási maradékát; **(0.25 pont)**
  - a beolvasott szám utolsó számjegyét; **(0.5 pont)**
  - a beolvasott szám első számjegyét. **(0.75 pont)**
  - a beolvasott szám számjegy-előfordulási statisztikáját. **(1 pont)**
- 4) **(maximum 1 pont)** Írj Pascal vagy C/C++ programot, amely beolvas a billentyűzetről egy  $n$  ( $1 \leq n \leq 999$ ) értéket, és kiírja a képernyőre
- az 1, 2, 3, 4, ...,  $n$  számsorozatot; **(0.25 pont)**
  - az  $n-1, n-2, \dots, 2, 1, 0$  számsorozatot; **(0.5 pont)**

- c. az 0, 1, 1, 2, 3, 5, 8, . . . számsorozat első  $n$  elemét (a következő az előző kettő összege); (Például,  $n=6$ -ra a 0, 1, 1, 2, 3, 5 számsorozat kell, hogy megjelenjen a képernyőn) **(0.75 pont)**
- d. az 1, 1, 2, 1, 2, 3, 1, 2, 3, 4, . . . számsorozat első  $n$  elemét. (Például,  $n=5$ -re az 1, 1, 2, 1, 2 számsorozat kell, hogy megjelenjen a képernyőn) **(1 pont)**
- 5) **(maximum 1 pont)** Írj Pascal vagy C/C++ programot, amely beolvas a billentyűzetről
- két egész számot (-999..999 tartományból) és kiírja ezeket csökkenő sorrendben a képernyőre; **(0.25 pont)**
  - három egész számot (-999..999 tartományból) és kiírja ezeket növekvő sorrendben a képernyőre; **(0.5 pont)**
  - egy  $n$  ( $1 \leq n \leq 999$ ) értéket, majd  $n$  darab egész számot (-999..999 tartományból), és kiírja a képernyőre a növekvő sorrendbe rendezett számsorozatot. **(1 pont)**
- 6) **(maximum 1 pont)** Írj Pascal vagy C/C++ programot, amely
- billentyűzetről beolvas két egész számot (-999..999 tartományból) és kiírja a képernyőre a kisebbiket; **(0.25 pont)**
  - billentyűzetről beolvas három egész számot (-999..999 tartományból) és kiírja a képernyőre a legnagyobbikat; **(0.5 pont)**
  - állományból beolvas egy  $n$  ( $1 \leq n \leq 999$ ) értéket, majd  $n$  darab egész számot (-999..999 tartományból), és kiírja a képernyőre a legkisebb és legnagyobb értékek első előfordulási helyeinek *sorszámát*; **(0.75 pont)**
  - állományból beolvassa az  $n$  ( $1 \leq n \leq 10$ ) és  $m$  ( $1 \leq m \leq 10$ ) értékeket és egy  $n \times m$  méretű egész számokat (-999..999 tartományból) tartalmazó mátrixot, és kiírja a képernyőre a páratlan sorszámú sorok elemeinek minimum *értékét*, a páros sorszámú sorok elemeinek pedig a maximum *értékét*. Adott sor minimumát/maximumát egy-egy függvénnyel határozd meg. **(1 pont)**
- 7) **(maximum 1 pont)** Képzeld el  $n$  várost, amelyeket az  $1, 2, \dots, n$  természetes számok azonosítanak (bármely két város között van közvetlen út). Egy bementi állományban adott az  $n$  ( $1 \leq n \leq 10$ ) érték és egy  $n \times n$  méretű természetes számokat (1..999 tartományból) tartalmazó mátrix. A mátrix  $(i, j)$  pozíciójú eleme azt ábrázolja, hogy mekkora a közvetlen távolság az  $i$  és  $j$  városok között. Írj Pascal vagy C/C++ programot, amely meghatározza (és kiírja a képernyőre) egy utazókereskedő útvonalának a hosszát, aki az alábbi algoritmus szerint járja be a városokat (minden várost pontosan egyszer érintve):
- indul az 1-es városból;
  - minden lépésben a kurrens városhoz legközelebbi, még meg nem látogatott városba megy;
  - az utolsóként érintett városból visszatér (a közvetlen úton) az 1-es városba.