

**BBTE Matek-Infó verseny – mintatételsor**  
**Informatika írásbeli vizsga**

**A. tételsor (30 pont)**

1. (5p) Tekintsük a következő alprogramot:

Alprogram  $f(a)$ :

Ha  $a \neq 0$ , akkor

visszatérít:  $a + f(a - 1)$

különben

visszatérít 0

Ha vége

Alprogram vége

Melyik kijelentés hamis az alábbiak közül?

- $f$  egy rekurzív módon definiált alprogram.
- Hogyha  $a$  negatív, az alprogram 0-t térít vissza.
- Az  $f$  által kiszámolt érték  $a * (a+1) / 4$ .
- Az alprogram kiszámolja az  $a$ -nál kisebb vagy egyenlő természetes számok összegét.
- Az  $f(-5)$  hívás végtelen ciklust eredményez.

2. (5p) Legyen a következő alprogram:

Alprogram  $f(a, b)$ :

Ha  $(a > 1)$ , akkor

visszatérít:  $b * f(a - 1, b)$

különben

visszatérít: 1

Ha vége

Alprogram vége

Határozzuk meg,  $f$  hányszor kerül meghívásra az alábbi kódban:

$x \leftarrow 4;$

$y \leftarrow 3;$

$z \leftarrow f(x, y);$

a. 4-szer

b. 2-szer

c. végtelen sokszor

d. egyszer sem

e. egyszer

3. (5p) Tekintsük az összes  $l \in \{1, 2\}$  hosszúságú,  $\{a, b, c, d, e\}$  elemekből felépített tömböt. Ezek közül hánynak lesznek szigorúan növekvő sorrendbe rendezett és páros számú magánhangzót tartalmazó elemei? ( $a$  és  $e$  magánhangzók)

a. 7

b. 80

c. 81

d. 78

e. 2

4. (5p) Egy 8 sorból álló mátrix, amely a 0 és 1 elemekből épül fel, a következő három tulajdonsággal rendelkezik:

a. az első sorban egyetlen 1-es értékű elem található,

b. a  $j$ -edik sor kétszer annyi nemnulla elemet tartalmaz mint a  $j-1$ -edik, minden  $j \in \{2, 3, \dots, 8\}$ -ra,

c. az utolsó sor egyetlen 0 értékű elemet tartalmaz.

Összesen hány 0 értékű elemet tartalmaz a mátrix?

a. 777

b. 769

c. 528

d. nem létezik ilyen mátrix

e. 1

5. (5p) Adottak az  $a$ ,  $b$ , és  $c$  tömbök, melyek hossza rendre  $n$ ,  $m$  és  $k$ , illetve adottak a következő alprogramok:

Alprogram F1( $x$ , 1):

$s \leftarrow 0$

Minden  $i \leftarrow 1$ ,  $l$ -re végezd el:

$s \leftarrow s + x[i]$

Minden vége

visszatérít:  $s$

Alprogram vége

Alprogram F2( $n_1$ ,  $n_2$ ):

visszatérít:  $n_1 + n_2$

Alprogram vége

Melyek utasítások lesznek helyesek az alábbiak közül a rendre  $n$ ,  $m$  és  $k$  elemű  $a$ ,  $b$  és  $c$  tömbök létezése esetén?

a.  $F_2(F_2(F_1(a, n), F_1(b, m)), k)$

b.  $val = F_1(c, k) + F_2(F_1(b, m), F_1(a, n))$

c.  $val = F_1(c, k) + F_2(F_1(a, m), b, n)$

d.  $F_2(F_2(F_1(a, n), F_1(b, m)), F_1(c, k))$

e.  $val = F_1(k, c) + F_2(F_1(m, b), F_1(n, a))$

6. (5p) Adott a következő alprogram:

Alprogram  $fc(a, s)$ :

$k = 0$

Minden  $i \leftarrow 1$ ,  $hossz(s)$ -re végezd el:

$k = k + a$

Minden vége

visszatérít:  $k$

Alprogram vége

Határozzuk meg, az alábbi utasítássorozatok közül melyek hatására jelenik meg 75?

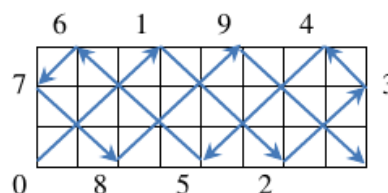
Megjegyzés: feltételeztük, hogy a tömbök indexelése 1-től kezdődik.

- $nr = fc("ana", 25)$   
kiírás(nr)
- $nr = fc(25, "ana")$   
kiírás(nr)
- kiírás( $fc(25, "ana")$ )
- nem létezik ilyen hívás
- kiírás( $fc("ana", 25)$ )

## B. tételsor (60 pont)

### 1. Fénysugár (25 pont)

Van egy befelé néző tükrökből álló téglalap alakú keretünk. Egy fénysugár indul a téglalap bal alsó sarkából annak alsó oldalával  $45^\circ$ -os szöget bezárva, és nekiütközik a téglalap felső oldalának. Itt a fénysugár visszatükröződik (elindul egy másik oldal irányába, újfent  $45^\circ$ -os szöget bezárva azzal az oldallal, melynek nekiütközött). A sugár addig folytatja az útját, amíg a téglalap egy sarkába nem ér.



Írjunk egy alprogramot, amely kiszámítja, hogy hányszor változtat irányt a fénysugár (*valtSzam*), amíg végül megáll egy sarokban. A kezdőpontot nem számítjuk be. Az alprogram bemenő paraméterei a téglalap hosszúsága ( $1 < a < 10\,000$ ) és szélessége ( $1 < b < 10\,000$ ), a *valtSzam* pedig a kimenő paraméter ( $a, b, \text{valtSzam} \in \mathbf{N}$ ).

**1. példa:** ha  $a = 8$  és  $b = 3$ , akkor *valtSzam* = 9.

**2. példa:** ha  $a = 8$  és  $b = 4$ , akkor *valtSzam* = 1.

### 2. Vírusok (15 pont)

Egy kísérlet keretén belül egy  $n$  ( $3 \leq n \leq 1000$ ) vírusból álló populáció a következőképpen fejlődhet:

- ha egy óra kezdetekor a populáció *páros* számú vírusból áll, akkor az óra végén a populáció 50%-kal csökken;
- ha egy óra kezdetekor a populáció *páratlan* számú vírusból áll, akkor az óra végén a víruspopuláció 1 vírussal szaporodik;
- ha egy óra végén a populáció egy *túlélési küszöbértéknél szigorúan kevesebb* vírusból áll, akkor a populáció eltűnik.

Írjunk egy alprogramot, amely meghatározza az órák számát (*oraSzam*), mely a kezdeti  $n$  vírusból álló populáció kipusztulásához szükséges, ismerve a  $k$  ( $2 \leq k < n$ ) túlélési küszöbértéket. A bemenő paraméterek  $n$  és  $k$ , a kimenő paraméter pedig az *oraSzam*.

**Példa:** ha  $n = 11$  és  $k = 3$ , akkor a populáció *oraSzam* = 5 óra alatt kipusztul.

### 3. Rendezés (10 pont)

Adott a következő alprogram:

```

1:   Alprogram rendezés(a, n):
2:       Ha  $n > 0$ , akkor
3:           rendezés(a,  $n - 1$ )
4:            $x \leftarrow a[n]$ 
5:            $j \leftarrow n - 1$ 
6:           Amíg  $(j \geq 0 \text{ and } a[j] > x)$ , végezd el:
7:                $j \leftarrow j - 1$ 
8:           Amíg vége
9:            $a[j + 1] \leftarrow x$ 
10:      Ha vége
11:      Alprogram vége

```

Milyen utasítással/utasításokkal kell ezt kiegészítenünk, illetve pontosan hol ahhoz, hogy a rendezés(a, n) hívás következtében az az  $n$  darab természetes számot tartalmazó tömb rendezve legyen?

#### 4. Ellenőrző számjegy (10 pont)

Adott az alábbi alprogram egy legkevesebb 2 számjegyű természetes szám ellenőrző számjegyének kiszámítására:

```

1: Alprogram ellenorzoSzamjegy(x):
2:     Amíg  $x > 9$ , végezd el:
3:          $s \leftarrow 0$ 
4:         Amíg  $x > 0$ , végezd el:
5:              $s \leftarrow s + x \text{ MOD } 10$  { $x \text{ mod } 10$   $x$  10-zel való osztási maradékát adja meg}
6:              $x \leftarrow x \text{ DIV } 10$  { $x \text{ div } 10$   $x$  10-zel való osztási hányadosát adja meg}
7:         Amíg vége
8:          $x \leftarrow s$ 
9:     Amíg vége
10:    Visszatérít:  $x$ 
11: Alprogram vége

```

Helyettesítsük az alprogram belsejét legfeljebb 2 utasítással úgy, hogy az alprogram ugyanazt a műveletet végezze.

#### Megjegyzések:

1. Minden tétel kötelező.
2. A megoldásokat a vizsgalapokra kell felvezetni (a piszkozatokat nem vesszük figyelembe).
3. Hivatalból 10 pont jár.
4. A munkaidő 3 óra.

## JAVÍTÓKULCS

### HIVATALBÓL

10 pont

### A. TÉTELSOR

30 pont

A.1. b, c és d válaszok

5 pont

A.2. a válasz

5 pont

A.3. a válasz

5 pont

A.4. a válasz

5 pont

A.5. b válasz

5 pont

A.6. b és c válaszok

5 pont

### B. TÉTELSOR

60 pont

#### **B.1. Fénysugár**

25 pont

V1: *valtSzam* helyes értékének kiszámítása az *lnko(a, b)* felhasználásával

25 pont

*lnko(a, b)* (vagy *lkkt(a, b)*)

10 pont

*valtSzam* értékének kiszámítása

15 pont

V2: *valtSzam* értékének helyes meghatározása egy másik helyes algoritmussal (szimulációval)

15 pont

#### **B.2. Vírusok**

15 pont

- Iteratív vagy rekurzív megoldás

10 pont

- Helyes számítások (a populáció kihalt egy óra végén)

5 pont

#### **B.3. Rendezés**

10 pont

- feltétel beazonosítása ( $a[j + 1] \leftarrow a[j]$ )

5 pont

- az utasítás beszúrása a 6. és 7. sorok közé

5 pont

#### **B.4. Ellenőrző számjegy**

10 pont

- egy szám ellenőrző számjegyének kiszámítása *nr* mod 9 módon

10 pont

## MEGOLDÁS

### MEGOLDÁS – B.1. Fénysugár

25 pont

// Az a és b számok legnagyobb közös osztójának kiszámítása.

```
int lnko(int a, int b){
    if ((a == b) && (a == 0))
        return 1;
    if (a * b == 0)
        return a + b;
    while (a != b)
        if (a > b)
            a -= b;
        else
            b -= a;
    return a;
}
```

// A sugár irányváltoztatásai számának meghatározása.

```
int sugar(int a, int b){
    int d = lnko(a, b);
    return b / d + a / d - 2;
}
```

### MEGOLDÁS – B.2. Vírusok

15 pont

// Adott k túlélési küszöbérték mellett meghatározza egy víruspopuláció

// kihalásához szükséges órák számát.

```
int virusok(int n, int k){
    bool kihal = (n < k);
    int oraSzam = 0;
    while (!kihal){
        if (n % 2 == 0) // Páros számú vírus esetén felezzük a populációt.
            n = n / 2;
        else // Páratlan vírus esetén növeljük 1-gyel a populáció méretét.
            n = n + 1;
        oraSzam = oraSzam + 1;
        kihal = (n < k); // Ellenőrizzük a populáció kipusztulását.
    }
    return oraSzam;
}
```

### MEGOLDÁS – B.3. Rendezés

10 pont

A 6. és 7. sor közé az  $a[j + 1] \leftarrow a[j]$  utasítást kell beilleszteni, azaz:

```
1:   Alprogram rendezés(a, n):
2:       Ha  $n > 0$ , akkor
3:           rendezés(a,  $n - 1$ )
4:            $x \leftarrow a[n]$ 
5:            $j \leftarrow n - 1$ 
6:           Amíg  $(j \geq 0 \text{ and } a[j] > x)$ , végezd el:
7:                $a[j + 1] \leftarrow a[j]$ 
```

```
8:          j ← j - 1
9:          Amíg vége
10:         a[j + 1] ← x
11:         Ha vége
12:        Alprogram vége
```

**MEGOLDÁS – B.4. Ellenőrző számjegy**

**10 pont**

Alprogram ellenorzoSzamjegy(x):  
 Visszatérít: x mod 9  
Alprogram vége