

Felvételi vizsga – mintatételsor
Informatika írásbeli vizsga

A. tételsor (30 pont)

1. (5p) Egy x biten tárolt egész adattípus (x szigorúan pozitív természetes szám) egész értékek tárolására képes a következő intervallumból:

- a. $[0, 2^x]$
- b. $[0, 2^{x-1}-1]$
- c. $[-2^{x-1}, 2^{x-1}-1]$
- d. $[-2^x, 2^x-1]$
- e. $[0, 10^x]$

2. (5p) Adott a következő alprogram:

Alprogram $f(a, b)$:

Ha $a > 1$, akkor

Visszatérít: $b * f(a - 1, b)$

Különben

Visszatérít: $b * f(a + 1, b)$

Ha vége

Alprogram vége

Határozzuk meg, hogy a következő kód esetén hányszor hívódik meg az f függvény:

- | | |
|-------------------|---------------------|
| $x \cdot 4$ | a. 4-szer |
| $y \cdot 3$ | b. 3-szor |
| $z \cdot f(x, y)$ | c. végtelen sokszor |
| | d. egyszer sem |
| | e. egyszer |

3. (5p) Legyen x egy egész típusú változó, amely a legkisebb olyan nemnulla természetes számot tartalmazza, amely 36 többszöröse és osztható minden 10-nél kisebb prímszámmal. Határozzuk meg, mely kijelentések igazak az alábbiak közül:

- a. $(x < 1000)$ and $((x*x*x) \bmod 1000 = 0)$
- b. $(x \bmod 100 = 0)$ or $(x \operatorname{div} 100 = 0)$
- c. $(x > 1000)$ && $(x \bmod 7 = 0)$
- d. $((x*x) \operatorname{div} 16) \bmod 2 = 1$
- e. $((x*x) \operatorname{div} 16) \bmod 2 = 0$

4. (5p) Tekintsük az összes $l \in \{1, 2, 3\}$ hosszúságú tömböt, melyek az $\{a, b, c, d, e\}$ betűkből épülnek fel. Ezek közül hány tömb lesz szigorúan csökkenő sorrendbe rendezett, páratlan számú magánhangzóval (magánhangzók: a és e)?

- a. 14
- b. 7

- c. 81
- d. 78
- e. 0

5. (5p) Adott a következő kódrészlet:

```
void pozitivSzamok(int m, int a[],
  int &n, int b[]){
  n = 0;
  for(int i = 0; i < n; i++){
    if (a[i] > 0){
      n = n + 1;
      b[n] = a[i];
    }
  }
}
```

```
procedure pozitivSzamok(m:integer;
  a:tomb; var n:integer; var b:tomb)
Begin
  n := 0;
  for i := 1 to n do
    if (a[i] > 0) then
      begin
        n := n + 1;
        b[n] := a[i];
      end;
  end;
End;
```

Mi lesz a *pozitivSzamok(k, x, l, y)* hívás eredménye $k = 4$, $x = (-1, 2, -3, 4)$, $l = -1$ és $y = ()$ (üres tömb) esetén?

- a. $l = 3$ és $y = (2, 4)$.
- b. $l = 0$ és $y = (2, 4)$.
- c. $l = 0$ és $y = ()$.
- d. A k értékétől függ.
- e. Fordítási hiba.

6. (5p) Tekintsük a következő alprogramot:

```
Alprogram SA6(a):
  Ha a < 50, akkor
    Ha a mod 3 = 0, akkor
      Visszatérít: SA6(2 * a - 3)
    Különben
      Visszatérít: SA6(2 * a - 1)
  Ha vége
  Különben
    Visszatérít a
  Ha vége
```

Alprogram vége

Az a paraméter mely értéke esetén lesz 61 a visszatérített érték?

- a. 16
- b. 61
- c. 4
- d. 31
- e. 51

B. tételsor (60 pont)

1. Csokikóstolás (25 pont)

Egy reklámügynökség egy új csokoládéhoz készít reklámot, és csokoládédarabokat készül szétosztani n ($10 \leq n \leq 10\,000\,000$) gyerek között, akik egy körben ülnek. A cég alkalmazottai rájönnek, hogy a csokidarabok szétosztása minden gyerek között túl sokba kerülne. Ezért úgy döntenek, hogy csak minden k -adik ($0 < k < n$) gyerek kap csokit. Megszámolják a körben ülő gyerekeket, és minden k -adiknak adnak egy darab csokoládét (ha az utolsó gyerekhez érnek, akkor a számlálás az elsőtől folytatódik). A számlálásnál minden gyereket figyelembe vesznek, függetlenül attól, hogy kapott-e már csokoládét vagy sem. A számlálás akkor fejeződik be, ha egy olyan gyereknek kellene újból csokoládét osztani, aki már korábban kapott.

Írjunk egy alprogramot, amely meghatározza a gyerekek számát (nr), akik nem kapnak csokoládédarabkát. Az n, k bemenő paraméterek természetes számok, a kimenő paraméter az nr természetes szám.

1. példa: Ha $n = 12$ és $k = 9$, akkor $nr = 8$ (az 1., 2., 4., 5., 7., 8., 10. és 11. gyerek nem kap csokoládét).

2. példa: Ha $n = 15$ és $k = 7$, akkor $nr = 0$ (minden gyerek kap csokoládét).

2. Bűvös számok (15 pont)

Tekintsünk két, p és q ($2 \leq p \leq 10, 2 \leq q \leq 10$) természetes számot. Egy természetes számot *bűvösnek* nevezünk, ha a p alapú számrendszerbeli felírásához felhasznált számjegyek halmaza megegyezik a q alapú számrendszerbeli felírásához szükséges számjegyek halmazával. Például, $p = 9$ és $q = 7$ értékekre $(31)_{10}$ egy bűvös szám, mivel $(34)_9 = (43)_7$, illetve $p = 3$ és $q = 9$ értékek esetén $(9)_{10}$ bűvös szám, mert $(100)_3 = (10)_9$.

Írjunk egy olyan alprogramot, amely a p és q alapszámokra meghatározza azt az x tömböt, amely az összes 0-nál szigorúan nagyobb és egy adott n természetes számnál ($1 < n \leq 10\,000$) szigorúan kisebb bűvös számokat tartalmazza. A bemenő paraméterek a p, q alapszámok, illetve n , a kimenő paraméterek pedig az x tömb, illetve annak k hossza.

Példa: $p = 9, q = 7$ és $n = 500$ esetén az x tömb $k = 11$ elemet fog tartalmazni: (1, 2, 3, 4, 5, 6, 31, 99, 198, 248, 297).

3. Keresés (10 pont)

Adott a következő alprogram:

```
1: Alprogram kereses(x, n, val):
2:   Ha n = 0, akkor
3:     Visszatérít: (x[0] = val)
4:   Különben
5:     Visszatérít: kereses(x, n - 1, val)
6:   Ha vége
7: Alprogram vége
```

Milyen utasítással vagy utasításokkal kell ezt kibővíteni és pontosan hol ahhoz, hogy meghívása esetén a függvény meghatározza, a *val* elem megtalálható-e vagy sem az n elemű x tömbben (az n érték szigorúan nagyobb mint 0)?

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 ← 0
4:     Amíg x > 0, végezd el:
5:       s ← s + x MOD 10 {x mod 10 x 10-zel való osztási maradékát adja meg}
6:       x ← x DIV 10 {x div 10 x 10-zel való osztási hányadosát adja meg}
7:     Amíg vége
8:     x ← s
9:   Amíg vége
10:  Visszatérít: x
11: Alprogram vége
```

Helyettesítsük az alprogram belsejét legfennebb 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
<u>A. TÉTELSOR</u>	30 pont
A.1. b, c válaszok	5 pont
A.2. c válasz	5 pont
A.3. c, d válaszok	5 pont
A.4. a válasz	5 pont
A.5. c válasz	5 pont
A.6. a, b, d válaszok	5 pont
<u>B. TÉTELSOR</u>	60 pont
B.1. Csokikóstolás	25 pont
V1: nr értékének helyes meghatározása az $nr = n - n/\ln k o(n, k)$ képlettel	25 pont
V2: nr értékének helyes meghatározása szimulációval, körkörös listával	15 pont
B.2. Bűvös számok	15 pont
- a bűvös tulajdonság ellenőrzése:	
V1: a két (p és q alapú számrendszerbeli) reprezentáció alapján felépített, a szám számjegyei halmazának karakterisztikus vektorai összehasonlítása alapján	10 pont
V2: más helyes, alacsonyabb hatékonyságú algoritmus	maximum 5 pont
- az x tömb megépítése	5 pont
B.3. Keresés	10 pont
- feltétel beazonosítása ($x[n] = val$)	5 pont
- az 5. sorban lévő összetett feltétel igazságértékének visszatérítése	5 pont
B.4. Ellenőrző számjegy	10 pont
- egy szám ellenőrző számjegyének kiszámítása $nr \bmod 9$ módon	10 pont

MEGOLDÁS – B.1. Csokikóstolás

25 pont

```
#include <iostream>
using namespace std;

//Kiszámítja és visszatéríti az a és b számok legnagyobb közös osztóját
int lnko(int a, int b){
    if ((a == b) && (a == 0))
        return 1;
    if (a * b == 0)
        return a + b;
    while (b != 0){
        int c = b;
        b = a % b;
        a = c;
    }
    return a;
}

// Meghatározza és visszatéríti azon gyerekek számát, akik nem kapnak
// csokoládét az n gyerek közül, számlálással meghatározva minden k-adik
// gyereket. A körkörös számlálást úgy is tekinthetjük, mint egy lineáris
// számlálás több kisebb tömbben, mindegyik n gyereket tartalmazva, így egy
// nagy, p gyereket tartalmazó tömböt kapva (p többszöröse n-nek).
// A számolás akkor fejeződik be, amikor (egy kicsi tömbben) az n-edik
// gyerek kap csokoládét (így a következő gyerek, akinek csokoládét kellene
// kapnia, a k-adik lesz a következő kicsit tömbben), tehát p k-nak is
// többszöröse kell legyen. Ezért  $p = \text{lnko}(n, k)$ . Az összes p gyerek közül
// pontosan  $p/k$  kapott csokoládét, tehát a csokoládét nem kapó gyerekek
// száma  $nr = n - p/k = n - \text{lnko}(n, k)/k = n - (n*k/\text{lnko}(n, k))/k$ 
//  $= n - n/\text{lnko}(n, k)$ .
int csokiKostolas(int n, int k){
    return n - n/lnko(n, k);
}
```

MEGOLDÁS – B.2. Bűvös számok

15 pont

```
#include <iostream>
using namespace std;

// Megépítjük az x szám p alapú számrendszerbeli ábrázolásában szereplő
// számjegyek megjelenéseinek vektorát.
// Rendre meghatározzuk x q alapú ábrázolásának számjegyeit.
// ha az aktuális számjegy nem jelenik meg a p alapú ábrázolásban,
// akkor az x szám nem bűvös;
// különben a számjegyek megfelelő értéket növeljük a számjegyek vektorában.
// Ha a számjegyek vektorában bizonyos számjegyek esetén 1-es maradt,
// és ezek megjelennek a p alapú reprezentációban, de a q alapúban nem,
// akkor a szám nem bűvös.
bool buvosSzam(int x, int p, int q){
```

```

//ellenőrzni, hogy x bűvös-e p és q alapszámok esetén
int szamjegyek[10] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
int mas = x;
while (mas != 0) { //meghatározzuk x számjegyeinek halmazát p alapszámmal
    int usz = mas % p; //usz - utolsó számjegy (p alapszámmal)
    szamjegyek[usz] = 1;
    mas = mas / p;
}
mas = x;
while (mas != 0) { //meghatározzuk x számjegyeinek halmazát q alapszámmal
    int usz = mas % q;
    if (szamjegyek[usz] == 0) //ha az aktuális számjegy (q alapszámmal) nincs
        //használva p alapszám esetén
        return false;
    szamjegyek[usz]++;
    mas = mas / q;
}
for (int i = 0; i < 10; i++) {
    if (szamjegyek[i] == 1) //ha az i számjegy használva van p alapszám esetén,
        //de q alapszám esetén nincs
        return false;
}
return true;
}

void buvosSzamok(int p, int q, int n, int &k, int tomb[]) {
    k = 0;
    for (int i = 1; i < n; i++) {
        if (buvosSzam(i, p, q))
            tomb[k++] = i;
    }
}

```

MEGOLDÁS – B.3. Keresés

10 pont

Az 5. sort a következőképpen kell módosítani: Visszatérít: $((x[n] = \text{val}) \text{ és kereses}(x, n - 1, \text{val}))$

```

1: Alprogram kereses(x, n, val):
2:   Ha n = 0, akkor
3:     Visszatérít: x[0] = val
4:   Különben
5:     Visszatérít: ((x[n] = val) és kereses(x, n - 1, val))
6:   Ha vége
7: Alprogram vége

```

MEGOLDÁS – B.4. Ellenőrző számjegy

10 pont

```

1: Alprogram ellenorzoSzamjegy(x):
2:   Visszatérít: x mod 9
3: Alprogram vége

```