

A versenyzők figyelmébe:

1. Minden tömböt 1-től kezdődően indexelünk.
2. A rácstesztekre (A rész) egy vagy több helyes válasz lehetséges. A válaszokat a vizsgadolgozatba írjátok (nem a feladatlapra). Ahhoz, hogy a feltüntetett pontszámot megkapjátok, elengedhetetlenül szükséges, hogy minden helyes választ megadjatok, és kizárólag csak ezeket.
3. A B részben szereplő feladatok megoldásait részletesen kidolgozva a vizsgadolgozatba írjátok.
 - a. A feladatok megoldásait *pszeudokódban* vagy egy *programozási nyelvben* (Pascal/C/C++) kell megadni.
 - b. A megoldások értékelésekor az első szempont az algoritmus **helyessége**, majd a **hatékonysága**, ami a **végrehajtási időt** és a **felhasznált memória méretét** illeti.
 - c. A tulajdonképpeni megoldások előtt, **kötelezően leírjátok szavakkal az alprogramokat (algoritmusokat), és megindokoljátok a megoldásokat lépéseit**. Feltétlenül írjátok **megjegyzéseket** (kommenteket), amelyek segítik az adott megoldás technikai részleteinek megértését. Adjátok meg az azonosítók jelentését és a fölhasznált adatszerkezeteket stb. Ha ez hiányzik, a tételre kapható pontszámotok 10%-kal csökken.
 - d. Ne használjatok különleges fejlécanyagokat, előredefiniált függvényeket (például STL, karakterláncokat feldolgozó sajátos függvények stb.).

A rész (30 pont)

A.1. Vajon mit csinál? (5 pont)

A generál(n) algoritmus egy n természetes számot dolgoz fel ($0 < n < 100$).

```
Algoritmus generál(n):  
    szám ← 0  
    Minden i ← 1, 1801 végezd el  
        használti ← hamis  
    vége(minden)  
    Amíg nem használtn végezd el  
        összeg ← 0; használtn ← igaz  
        Amíg n ≠ 0 végezd el  
            számjegy ← n MOD 10  
            összeg ← összeg + számjegy * számjegy * számjegy  
            n ← n DIV 10  
        vége(amíg)  
        n ← összeg; szám ← szám + 1  
    vége(amíg)  
    térítsd szám  
Vége(algoritmus)
```

Állapítsátok meg a fenti algoritmus hatását.

- A. ismételten kiszámítja az n szám számjegyei köbének összegét ameddig az összeg egyenlővé válik az n számmal és visszatéríti a végrehajtott ismétlések számát
- B. kiszámítja az n szám számjegyei köbének összegét és visszatéríti ezt az összeget
- C. kiszámítja az n szám számjegyei köbének összegét, felülírja n értékét ezzel az összeggel, és visszatéríti ezt az összeget
- D. ismételten kiszámítja az n szám számjegyei köbének összegét ameddig egy összeget másodszorra kap meg, és visszatéríti a végrehajtott ismétlések számát
- E. meghatározza, hogy hányszor lesz felülírva az n szám a számjegyei köbének összegével, ameddig egy már kiszámolt értéket vagy magát a számot kapja és visszatéríti ezt a számot

A.2. Mely értékekre van szükség? (5 pont)

Adott a feldolgoz(v , k) algoritmus, ahol v egy k elemű, természetes számokat tároló sorozat ($1 \leq k \leq 1000$).

```
Algoritmus feldolgoz(v, k)  
    i ← 1; n ← 0  
    Amíg i ≤ k és vi ≠ 0 végezd el  
        y ← vi; c ← 0  
        Amíg y > 0 végezd el  
            Ha y MOD 10 > c akkor  
                c ← y MOD 10  
            vége(ha)  
            y ← y DIV 10  
        vége(amíg)  
        n ← n * 10 + c; i ← i + 1  
    vége(amíg)  
    térítsd n  
Vége(algoritmus)
```

Állapítsátok meg, v és k mely értékeire térít vissza az algoritmus 928-at.

- A. $v = (194, 121, 782, 0)$ és $k = 4$
- B. $v = (928)$ és $k = 1$
- C. $v = (9, 2, 8, 0)$ és $k = 4$
- D. $v = (8, 2, 9)$ és $k = 3$
- E. $v = (912, 0, 120, 8, 0)$ és $k = 5$

A.3. Logikai kifejezés kiértékelése (5 pont)

Adott a k elemű s sorozat, amelynek elemei logikai (boolean) típusúak és a $kiértékelés(s, k, i)$ algoritmus, ahol k és i természetes számok ($0 \leq i \leq k \leq 100$).

```

Algoritmus kiértékelés(s, k, i)
  Ha  $i \leq k$  akkor
    Ha  $s_i$  akkor
      térítsd  $s_i$ 
    különben
      térítsd ( $s_i$  vagy  $kiértékelés(s, k, i + 1)$ )
  vége(ha)
  különben
    térítsd hamis
  vége(ha)
Vége(algoritmus)

```

Határozzátok meg, hányszor hívja meg önmagát a $kiértékelés(s, k, i)$ algoritmus a következő programrészlet végrehajtásának következtében:

```

s ← (hamis, hamis, hamis, hamis, hamis, hamis, igaz, hamis, hamis, hamis)
k ← 10
i ← 3
kiértékelés(s, k, i)

```

- A. 3-szor
- B. ugyanannyiszor, mint a következő programrészlet esetében

```

s ← (hamis, hamis, hamis, hamis, hamis, hamis, hamis, igaz)
k ← 8
i ← 4
kiértékelés(s, k, i)

```

- C. 6-szor
- D. egyszer sem
- E. végtelenszer

A.4. Egyesítés (5 pont)

Adottnak tekintjük az $eleme(x, a, n)$ algoritmust, amely eldönti, hogy az x természetes szám eleme-e az n elemű a halmaznak; a egy n elemű sorozat, amely egy természetes számokat tartalmazó halmazt ábrázol ($1 \leq n \leq 200, 1 \leq x \leq 1000$). Legyenek az alább megadott egyesítés(a, n, b, m, c, p) és számol(a, n, b, m, c, p) algoritmusok, ahol a, b és c sorozatok, amelyek természetes számokat tároló és rendre n, m és p elemű halmazokat ábrázolnak ($1 \leq n \leq 200, 1 \leq m \leq 200, 1 \leq p \leq 400$). A bemeneti paraméterek a, n, b, m és p , kimeneti paraméterek pedig c és p .

```

1. Algoritmus egyesítés(a, n, b, m, c, p):
2.   Ha  $n = 0$  akkor
3.     Minden  $i \leftarrow 1, m$  végezd el
4.        $p \leftarrow p + 1$ 
5.        $c_p \leftarrow b_i$ 
6.     vége(minden)
7.   különben
8.     Ha nem  $eleme(a_n, b, m)$  akkor
9.        $p \leftarrow p + 1$ 
10.       $c_p \leftarrow a_n$ 
11.    vége(ha)
12.    egyesítés(a, n - 1, b, m, c, p)
13.  vége(ha)
14. Vége(algoritmus)

```

```

1. Subalgoritmus számol(a, n, b, m, c, p):
2.    $p = 0$ 
3.   reuniune(a, n, b, m, c, p)
4. SfSubalgoritmus

```

A következő állítások közül melyek bizonyulnak mindig igaznak?

- A. ha az **a** halmaz egyetlen elemet tartalmaz, a számol(a, n, b, m, c, p) algoritmus meghívása végtelen ciklust okoz
- B. ha az **a** halmaznak négy eleme van, a számol(a, n, b, m, c, p) algoritmus meghívása maga után vonja az egyesítés algoritmus 12. sorában található utasítás végrehajtását négyszer
- C. ha az **a** halmaznak öt eleme van, a számol(a, n, b, m, c, p) algoritmus meghívása maga után vonja az egyesítés algoritmus második sorában található utasítás végrehajtását ötször
- D. ha az **a** halmaznak ugyanannyi eleme van, mint a **b** halmaznak, a számol(a, n, b, m, c, p) algoritmus végrehajtása után a **c** halmaznak ugyanannyi eleme lesz, mint az **a** halmaznak
- E. ha az **a** és **b** halmazok elemei azonosak, a számol(a, n, b, m, c, p) algoritmus végrehajtása után a **c** halmaznak ugyanannyi eleme lesz, mint az **a** halmaznak

A.5. Hatványra emelés (5 pont)

Melyik alábbi algoritmus számítja ki helyesen a^b értékét, ahol **a** és **b** természetes számok ($1 \leq a \leq 11, 0 \leq b \leq 11$)?

A.	Algoritmus hatvány(a, b): eredmény $\leftarrow$ 1 Amíg b > 0 végezd el Ha b MOD 2 = 1 akkor eredmény $\leftarrow$ eredmény * a vége(ha) b $\leftarrow$ b DIV 2 a $\leftarrow$ a * a vége(amíg) térítsd eredmény Vége(algoritmus)	B.	Algoritmus hatvány(a, b): Ha b $\neq$ 0 akkor Ha b MOD 2 = 1 akkor térítsd hatvány(a * a, b / 2) * a különb térítsd hatvány(a * a, b / 2) vége(ha) különb térítsd 1 vége(ha) Vége(algoritmus)
C.	Algoritmus hatvány(a, b): eredmény $\leftarrow$ 1 Amíg b > 0 végezd el eredmény $\leftarrow$ eredmény * a b $\leftarrow$ b - 1 vége(amíg) térítsd eredmény Vége(algoritmus)	D.	Algoritmus hatvány(a, b): Ha b = 0 akkor térítsd 1 vége(ha) aux $\leftarrow$ hatvány(a, b DIV 2) Ha b MOD 2 = 0 akkor térítsd aux * aux különb térítsd a * aux * aux vége(ha) Vége(algoritmus)
E.	Algoritmus hatvány(a, b): Ha b = 0 akkor térítsd 1 vége(ha) térítsd a * hatvány(a, b - 1) Vége(algoritmus)		

A.6. Legnagyobb többszörös (5 pont)

Melyik alábbi algoritmus téríti az **a** természetes számnak azt a legnagyobb többszörösét, amely kisebb vagy egyenlő a **b** természetes számmal ($0 < a < 10\,000, 0 < b < 10\,000, a < b$)?

A.	Algoritmus f(a, b): c $\leftarrow$ b Amíg c MOD a $\neq$ 0 végezd el c $\leftarrow$ c - 1 vége(amíg) térítsd c Vége(algoritmus)	B.	Algoritmus f(a, b): Ha a < b akkor térítsd f(2 * a, b) különb Ha a = b akkor térítsd a különb térítsd b vége(ha) vége(ha) Vége(algoritmus)
C.	Algoritmus f(a, b): térítsd (b DIV a) * a Vége(algoritmus)		
D.	Algoritmus f(a, b): Ha b MOD a = 0 akkor térítsd b vége(ha) térítsd f(a, b-1) Vége(algoritmus)	E.	Algoritmus f(a, b): c $\leftarrow$ a Amíg c < b végezd el c $\leftarrow$ c + a vége(amíg) Ha c = b akkor térítsd c különb térítsd c - a vége(ha) Vége(algoritmus)

B rész (60 pont)

B.1. Polinom értéke (10 pont)

Adott a kiértékelés(n , $egyh$, x) algoritmus, ahol **egyh** egy $n + 1$ elemű, valós számokat tároló sorozat, amelynek értékei a $[-100, 100]$ intervallumhoz tartoznak és amelyek az n fokú $P(x) = egyh_1 * x^n + egyh_2 * x^{n-1} + \dots + egyh_n * x + egyh_{n+1}$ polinom együtthatói, x csökkenő hatványainak sorrendjében (n természetes szám, $1 \leq n \leq 10$). Az algoritmus meghatározza a polinom értékét egy adott x pontban (x valós szám, amely a $[-10, 10]$ intervallumhoz tartozik).

```
Algoritmus kiértékelés( $n$ ,  $egyh$ ,  $x$ ):  
    érték  $\leftarrow 0.0$   
    Minden  $i \leftarrow 1, n + 1$  végezd el  
        érték  $\leftarrow$  érték  $\cdot x + egyh[i]$   
    vége(minden)  
    térítsd érték  
Vége(algoritmus)
```

Írjátok le a kiértékelés(n , $egyh$, x) algoritmus *rekurzív* változatát (ismétlő struktúrák használata nélkül) úgy, hogy a fejléce és a hatása legyen azonos a fenti algoritmuséval.

B.2. Keresztmetszet (25 pont)

Adott két sorozat, amelyek $-30\,000$ és $30\,000$ közötti egész számokat tárolnak, amelyek az egyes sorozatokon belül *különbözők*. Az **a** sorozatnak n ($0 < n \leq 10\,000$) eleme, a **b** sorozatnak m ($0 < m \leq 10\,000$) eleme van, és *növekvően rendezett*.

Írjatok algoritmust, amely meghatározza azt a k elemű ($0 \leq k \leq 10\,000$) **c** sorozatot, amely az adott két sorozat közös elemeit tartalmazza, minden elemet csak egyszer, tetszőleges sorrendben. Az algoritmus bemeneti paraméterei a két sorozat (**a** és **b**) valamint a hosszúságaik (n és m). Kimeneti paraméterek a **c** sorozat és ennek k hosszúsága. Ha nincsenek közös elemek, k értéke 0.

Példa: ha $n = 4$, $a = (5, -7, -2, 3)$, $m = 5$ és $b = (-2, 3, 5, 7, 8)$, a **c** sorozatnak $k = 3$ eleme van és $c = (5, -2, 3)$.

B.3. Tömbszakasz – testvérszámok nélkül (25 pont)

Adott az n elemű ($0 < n \leq 10\,000$) **a** sorozat, amely $30\,000$ -nél kisebb, nem nulla *különböző* természetes számokat tárol. Két számot *testvéreknek* nevezünk, ha *nem azonosak* és *van legkevesebb két különböző közös számjegyük*. Például, 5867 és 17526 *testvérek*, miközben 5867 és 152 nem *testvérek*. Hasonlóképpen, 131 és 114 nem *testvérek*.

Követelmények:

- Írjatok algoritmust, amely eldönti, hogy az **a** természetes szám testvére-e a **b** természetes számnak ($0 < a \leq 30\,000$, $0 < b \leq 30\,000$). Az algoritmus bemeneti paraméterei az **a** és **b** számok. Kimeneti paraméter a **testvér** logikai változó, amelynek értéke *igaz* ha **a** testvére **b**-nek, különben *hamis*. **(11 pont)**
- Írjatok algoritmust, amely meghatározza az **a** sorozatnak azt a leghosszabb tömbszakaszát, amely az **a** sorozat azon elemeit tartalmazza, amelyeknek *nincs egyetlen testvérük sem* az **a** sorozatban. Egy sorozat tömbszakasza a sorozat egymás utáni pozícióin található elemeit tartalmazza. Az algoritmus bemeneti paraméterei az **a** sorozat és az n hosszúsága. Kimeneti paraméterek a leghosszabb tömbszakasz kezdeti pozíciója (**start**) és a tömbszakasz k hosszúsága. Ha több azonos hosszúságú leghosszabb tömbszakasz létezik, a legutolsót kell megadnotok. Ha nem létezik egyetlen, adott tulajdonságú tömbszakasz sem, **start** értéke -1 és k értéke 0. **(14 pont)**

Példa: Legyen $n = 10$ és $a = (12345, 9, 100, 567, 5678, 345, 123, 8989, 222, 11, 78)$. Az **a** sorozat elemei, amelyeknek nincs testvérük: 9, 100, 8989, 222, 11, a keresett tömbszakasz pedig: (8989, 222, 11), tehát **start** = 8 és $k = 3$.

Megjegyzések:

- Minden tétel kidolgozása kötelező.
- A piszkozatokat nem vesszük figyelembe.
- Hivatalból jár 10 pont.
- Rendelkezésekre áll 3 óra.

BAREM

OFICIU 10 puncte

Partea A 30 puncte

- A. 1. Oare ce face? Răspunsul E..... 5 puncte
- A. 2. Ce valori sunt necesare? Răspunsurile A, C 5 puncte
- A. 3. Evaluare logică. Răspunsul B 5 puncte
- A. 4 Reuniune. Răspunsurile B, E 5 puncte
- A. 5. Exponențiere. Răspunsurile A, B, C, D, E 5 puncte
- A. 6. Cel mai mare multiplu. Răspunsurile C, D, E..... 5 puncte

Partea B 60 puncte

B. 1. Evaluare polinom 10 puncte

- respectarea parametrilor de intrare și ieșire..... 2 puncte
- condiția de oprire din recursivitate..... 2 puncte
- autoapel 2 puncte
- valoarea returnată la oprirea recursivității..... 2 puncte
- valoarea returnată la continuarea recursivității 2 puncte

B. 2. Intersecție 25 puncte

- respectarea parametrilor de intrare și ieșire..... 2 puncte
 - variante:
 - folosirea căutării binare 23 puncte
 - fără căutare binară maxim 18 puncte

B. 3. Secvență de numere fără frați..... 25 puncte

- respectarea parametrilor de intrare și ieșire..... 2 puncte
- proprietatea de frate..... 10 puncte
- determinarea unei secvențe 9 puncte
- determinarea celei mai lungi secvențe..... 4 puncte