

Felvételi verseny – 2023 július 19
Informatika írásbeli

FONTOS MEGJEGYZÉS:

Más pontosítások hiányában:

- Feltételezhetjük, hogy az aritmetikai műveleteket végtelen adattípusokon végezzük, vagyis nincs *túlcsordulás* és *alulcsordulás*.
- Minden sorozatot 1-től sorszámozunk (indexelünk).
- Az aktuális paraméterek értékeire vonatkozó megszorítások az első hívás pillanatában érvényesek.
- Egy vektor tömbszakaszát a vektor egymás utáni pozícióin található elemek alkotják.

1. Legyen az $F(x)$ algoritmus, ahol x természetes szám ($1 \leq x \leq 10^6$):

Algorithm $F(x)$:

```
If  $x = 0$  then
    Return 0
Else
    If  $x \bmod 3 = 0$  then
        Return  $F(x \text{ DIV } 10) + 1$ 
    Else
        Return  $F(x \text{ DIV } 10)$ 
    EndIf
EndIf
EndAlgorithm
```

A következő hívások közül melyik térít vissza 4-et?

- A. $F(21369)$
- B. $F(6933)$
- C. $F(4)$
- D. $F(16639)$

2. Legyen a $ceFace(a, b)$ algoritmus, ahol a és b természetes számok ($1 \leq a, b \leq 10^4$), amelyek nem tartalmazzák a 0-ás számjegyet.

Algorithm $ceFace(a, b)$:

```
 $p \leftarrow 0$ 
While  $a \neq 0$  execute
     $c \leftarrow a \bmod 10$ 
     $p \leftarrow p * 10 + c$ 
     $a \leftarrow a \text{ DIV } 10$ 
EndWhile
If  $p = b$  then
    Return True
Else
    Return False
EndIf
EndAlgorithm
```

A $ceFace(a, b)$ algoritmus akkor és csakis akkor térít vissza *True*-t, ha:

- A. az a és b számok egyenlők
- B. a és b palindromszámok
- C. az a szám a b szám fordítottja (tükrözöttje)
- D. az a szám utolsó számjegye egyenlő a b szám utolsó számjegyével

3. Legyen a $ceFace(n)$ algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^3$). A „/” operátor a valós osztást jelöli, például: $3 / 2 = 1.5$.

Algorithm $ceFace(n)$:

```
 $s \leftarrow 0$ 
For  $i \leftarrow 1, n$  execute
     $p \leftarrow (i + 1) * (i + 2)$ 
     $s \leftarrow s + (i / p)$ 
EndFor
Return  $s$ 
EndAlgorithm
```

Melyik kifejezésnek az értékét téríti vissza az algoritmus?

- A. $\frac{1}{1} + \frac{1}{1+2} + \dots + \frac{1}{1+2+\dots+n}$
- B. $\frac{1}{2*3} + \frac{2}{3*4} + \dots + \frac{n}{(n+1)*(n+2)}$
- C. $\frac{1}{1} + \frac{1}{1*2} + \dots + \frac{1}{1*2*\dots*n}$
- D. $\frac{1}{2*3} + \frac{2}{3*4} + \dots + \frac{n-1}{n*(n+1)}$

4. Legyen az $f(n, x)$ algoritmus, ahol n természetes szám ($3 \leq n \leq 10^4$), x egy n elemű, természetes számokat tároló vektor ($x[1], x[2], \dots, x[n]$, $1 \leq x[i] \leq 10^4$, ha $i = 1, 2, \dots, n$).

Algorithm $f(n, x)$:

```

k ← 0
For i ← 1, n - 1 execute
  If k = 0 then
    If  $x[i] = x[i + 1]$  then
      Return False
    EndIf
    If  $x[i] < x[i + 1]$  then
      k ← 1
    EndIf
  EndIf
EndFor
If  $x[n - 1] \geq x[n]$  then
  Return False
EndIf
Return True
EndAlgorithm

```

Mely hívások esetében térít vissza az algoritmus *True*-t?

- A. $f(6, [1000, 512, 23, 22, 1, 2])$
- B. $f(6, [6, 4, 1, 1, 2, 3])$
- C. $f(8, [3000, 2538, 799, 424, 255, 256, 299, 1001])$
- D. $f(3, [3, 2, 1])$

5. Legyen a $\text{calcul}(a, b, c, d)$ algoritmus, ahol a, b, c, d nemnulla természetes számok ($1 \leq a, b, c, d \leq 100$).

Algorithm $\text{calcul}(a, b, c, d)$:

```

x ← a * b
y ← c * d
While y ≠ 0 execute
  z ← x MOD y
  x ← y
  y ← z
EndWhile
Return x
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Az algoritmus az a, b, c, d számok legnagyobb közös osztóját téríti vissza.
- B. Az algoritmus az $a * b$ és $c * d$ számok legnagyobb közös osztóját téríti vissza.
- C. Az algoritmus az a, b, c, d számok legkisebb közös többszörösét téríti vissza.
- D. Az algoritmus az $a * b$ és $c * d$ számok legkisebb közös többszörösét térít vissza.

6. Legyen a $p(na, a, nb, b)$ algoritmus, ahol na és nb természetes számok ($0 \leq na, nb \leq 10^4$), a és b egy na , illetve egy nb elemű, természetes számokat tároló vektor ($a[1], a[2], \dots, a[na]$, $1 \leq a[i] \leq 10^4$, ha $i = 1, 2, \dots, na$ és $b[1], b[2], \dots, b[nb]$, $1 \leq b[i] \leq 10^4$, ha $i = 1, 2, \dots, nb$). A c lokális változó egy vektor.

Algorithm $p(na, a, nb, b)$:

```

i ← 1
j ← 1
nc ← 0
While i ≤ na AND j ≤ nb execute
  nc ← nc + 1
  If  $a[i] < b[j]$  then
     $c[nc] \leftarrow a[i]$ 
    i ← i + 1
  Else
     $c[nc] \leftarrow b[j]$ 
    j ← j + 1
  EndIf
EndWhile
Return nc
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Ha $na = 0$ és $nb = 0$, akkor az nc -ben visszatérített érték 0-val egyenlő.
- B. Ha az a és b vektorok elemei növekvően rendezettek, akkor a c vektorba mentett elemek növekvően rendezettek lesznek.
- C. Az nc -ben visszatérített érték mindig egyenlő $na + nb$ -vel.
- D. Ha $na, nb > 0$ és az a vektor legnagyobb elemének értéke kisebb, mint a b vektor összes elemének értéke, akkor c -nek ugyanazok az elemei lesznek, mint a -nak.

7. Legyen a $\text{suma}(n, a, m, b)$ algoritmus, ahol n és m természetes számok ($1 \leq n, m \leq 10^5$), a és b két növekvően rendezett sorozat. Az a sorozat n , a b sorozat m természetes számot tárol ($a[1], a[2], \dots, a[n]$ és $b[1], b[2], \dots, b[m]$):

```

Algorithm suma(n, a, m, b):
  s ← 0
  For i ← 1, n, 2 execute
    j ← 1
    While j ≤ a[i] AND j ≤ m execute
      s ← s + b[j]
      j ← j + 1
    EndWhile
  EndFor
  Return s
EndAlgorithm

```

Mit térít vissza az algoritmus, ha $n = 4$, $a = [1, 3, 4, 7]$, $m = 6$ és $b = [2, 4, 6, 8, 10, 12]$?

- A. 42
- B. 22
- C. 20
- D. Nem lehet meghatározni, hogy mely értéket fogja visszatéríteni

8. Legyen a $\text{verifica}(n, p1, p2)$ algoritmus, ahol $n, p1$ és $p2$ természetes számok ($1 \leq n, p1, p2 \leq 10^6$):

```

Algorithm verifica(n, p1, p2):
  bt ← (p1 + p2) DIV 2
  If p1 > p2 then
    Return False
  EndIf
  If bt * bt = n then
    Return True
  EndIf
  If bt * bt > n then
    Return verifica(n, p1, bt - 1)
  EndIf
  Return verifica(n, bt + 1, p2)
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Ha $p1, p2$ és n relatív prímek, akkor a $\text{verifica}(n, p1, p2)$ hívás *True*-t térít vissza.
- B. Az algoritmus a bináris keresés módszerét alkalmazza és ha n prímszám, a $\text{verifica}(n, 1, n)$ hívás *True*-t térít vissza.
- C. A $\text{verifica}(n, 1, n)$ hívás akkor és csakis akkor térít vissza *True*-t, ha az n négyzetszám.
- D. Ha $p1 \leq n \leq p2$ és a $[p1, n]$ és $[n, p2]$ intervallumokban létezik legalább egy-egy négyzetszám, akkor a $\text{verifica}(n, p1, p2)$ hívás *True*-t térít vissza.

9. Legyen a $\text{ceFace}(n)$ algoritmus, ahol n egy természetes szám ($1 \leq n \leq 3000$).

```

Algorithm ceFace(n):
  s ← 0
  i ← 1
  While s < n execute
    s ← s + i
    If s = n then
      Return True
    Else
      i ← i + 2
    EndIf
  EndWhile
  Return False
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Ha $n = 36$, az algoritmus *True*-t térít vissza.
- B. Ha n egymás után következő páratlan számok olyan összege, amely az 1-es-től kezdődik, az algoritmus *True*-t térít vissza.
- C. Ha n négyzetszám, az algoritmus *True*-t térít vissza, különben *False*-t.
- D. Ha $n = 64$, az algoritmus *False*-t térít vissza.

10. Legyen a $\text{ceFace}(a)$ algoritmus, ahol a természetes szám ($1 \leq a \leq 10^4$).

```

Algorithm ceFace(a):
  ok ← 0
  While ok = 0 execute
    b ← a
    c ← 0
    While b ≠ 0 execute
      c ← c * 10 + b MOD 10
      b ← b DIV 10
    EndWhile
    If c = a then
      ok ← 1
    Else
      a ← a + 1
    EndIf
  EndWhile
  Return a
EndAlgorithm

```

Állapítsátok meg az algoritmus hatását.

- A. Az algoritmus azt a legkisebb palindromszámot téríti vissza, amely a -nál nagyobb, vagy a -val egyenlő.
- B. Az algoritmus azt a legnagyobb palindromszámot téríti vissza, amely a -nál kisebb, vagy a -val egyenlő.
- C. Az algoritmus azt a legkisebb palindromszámot téríti vissza, amely a -nál nagyobb.
- D. Az algoritmus azt a legkisebb páros számot téríti vissza, amely a -nál nagyobb.

11. Legyen a $\text{calcul}(v, n)$ algoritmus, ahol n természetes szám ($1 \leq n \leq 10^4$) és v egy n elemű, természetes számokat tároló vektor ($v[1], v[2], \dots, v[n]$, $1 \leq v[i] \leq 10^4$, ha $i = 1, 2, \dots, n$):

```

Algorithm calcul(v, n):
  i ← 2
  x ← 0
  If v[1] MOD 2 ≠ 0 then
    Return False
  EndIf
  While i ≤ n execute
    If x = 0 AND v[i] MOD 2 = 0 then
      Return False
    Else
      If x = 1 AND v[i] MOD 2 = 1 then
        Return False
      Else
        i ← i + 1
        x ← (x + 1) MOD 2
      EndIf
    EndIf
  EndWhile
  Return True
EndAlgorithm

```

A következő esetek közül melyek esetében térít vissza *True*-t az algoritmus?

- A. Ha a v vektor a [2, 3, 10, 7, 20, 5, 18] számokat tartalmazza és $n = 7$
- B. Ha a v vektorban az értékek következő minta szerint találhatók: páratlan, páros, páratlan, páros...
- C. Ha a v vektor a [3, 8, 17, 20, 15, 10] számokat tartalmazza és $n = 6$
- D. Ha a v vektorban az értékek a következő minta szerint találhatók: páros, páratlan, páros, páratlan...

12. Legyen a $\text{ceFace}(a, n)$ algoritmus, ahol n nemnulla természetes szám ($2 \leq n \leq 10^4$) és a egy n egész számot tároló vektor ($a[1], a[2], \dots, a[n]$, $-100 \leq a[i] \leq 100$, $i = 1, 2, \dots, n$). Az a vektor tartalmaz legalább egy pozitív számot.

```

Algorithm ceFace(a, n):
  b ← 0
  c ← b
  For i ← 1, n execute
    b ← b + a[i]
    If b < 0 then
      b ← 0
    EndIf
    If b > c then
      c ← b
    EndIf
  EndFor
  Return c
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Az algoritmus az a vektor összes elemének összegét téríti vissza.
- B. Az algoritmus az a vektor azon leghosszabb tömbszakaszának összegét téríti vissza, amely csak pozitív számokat tartalmaz.
- C. Az algoritmus az a vektor összes pozitív elemének összegét téríti vissza.
- D. Az algoritmus az a vektor egy maximális összegű tömbszakaszának összegét téríti vissza.

13. Legyen az n soros és m oszlopos ($1 \leq n, m \leq 10^4$), egész számokat tároló A mátrix. Ha $n * m = p * q$, át szeretnénk alakítani ezt a mátrixot egy p soros és q oszlopos, egész számokat tartalmazó B mátrixszá ($1 \leq p, q \leq 10^4$), az alábbi példának megfelelően, ahol $n = 4$, $m = 6$, $p = 3$ és $q = 8$. A sorok és oszlopok 1-től kezdődően vannak számozva.

A:

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24

B:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24

A következő változatok között melyik az az algoritmus, amely az i és j ($1 \leq i \leq n$, $1 \leq j \leq m$) természetes számpár esetében, amelyek indexek az A mátrixban, visszatéríti azt az indexpárt, amely megfelel az $A[i][j]$ értéknek a B mátrixban?

- A.


```

Algorithm reshape(i, j, n, m, p, q):
  Return (i * m + j) DIV q, (i * m + j) MOD q
EndAlgorithm

```
- B.


```

Algorithm reshape(i, j, n, m, p, q):
  i ← i - 1
  j ← j - 1
  Return (i * m + j) DIV q, (i * m + j) MOD q
EndAlgorithm

```
- C.


```

Algorithm reshape(i, j, n, m, p, q):
  i ← i - 1
  j ← j - 1
  Return (i * m + j) DIV q + 1,
        (i * m + j) MOD q + 1
EndAlgorithm

```
- D.


```

Algorithm reshape(i, j, n, m, p, q):
  Return (i * m + j - 1) DIV q + 1,
        (i * m + j - 1) MOD q + 1
EndAlgorithm

```

14. Legyen a $\text{ceFace}(n, m)$ algoritmus, ahol n természetes szám ($1 \leq n \leq 10^4$), m egy n soros és n oszlopos mátrix, amelynek elemei természetes számok ($m[1][1], \dots, m[1][n], m[2][1], \dots, m[2][n], \dots, m[n][1], \dots, m[n][n]$). Feltételezzük, hogy az m mátrix elemei eredetileg 0-val egyenlők.

```

Algorithm ceFace(n, m):
  a ← 0
  b ← 1
  For j ← 1, n execute
    i ← 1
    While i + j ≤ n - 1 execute
      If (i MOD 2 = 1) AND (j MOD 2 = 1) then
        m[i][j] ← b
        c ← a + b
        a ← b
        b ← c
      EndIf
      i ← i + 1
    EndWhile
  EndFor
EndAlgorithm

```

A következő állítások közül melyek **HAMISAK**?

- A. Ha $n = 11$, $m[6][4]$ értéke 21
- B. Ha $n = 7$, $m[3][5]$ értéke 4
- C. Ha $n = 10$, $m[6][4]$ értéke 21
- D. Ha $n = 7$, a mátrix maximális elemének értéke 8

15. Az alábbi algoritmusok egy n elemű, természetes számokat tároló, növekvően rendezett vektort dolgoznak fel ($1 \leq n \leq 10^4$, $x[1], x[2], \dots, x[n]$). A **first** és **last** paraméterek természetes számok ($1 \leq \text{first} \leq \text{last} \leq n$). Válasszátok ki azokat az algoritmusokat, amelyeknek időbonyolultsága a legkisebb, ha $A(x, 1, n, n)$ alakban hívjuk meg.

A.

```

Algorithm A(x, first, last, n):
  If first > last then
    Return 0
  EndIf
  m ← (first + last) DIV 2
  If x[m] = n then
    Return m
  Else
    If x[m] > n then
      Return A(x, first, m - 1, n)
    Else
      If x[m] < n then
        Return A(x, m + 1, last, n)
      EndIf
    EndIf
  EndIf
EndAlgorithm

```

B.

```

Algorithm A(x, first, last, n):
  While first < last execute
    m ← (first + last) DIV 2
    If x[m] = n then
      Return m
    Else
      If x[m] > n then
        last ← m - 1
      Else
        If x[m] < n then
          first ← m + 1
        EndIf
      EndIf
    EndIf
  EndWhile
  Return 0
EndAlgorithm

```

C.

```

Algorithm A(x, first, last, n):
  For i ← first, last execute
    If x[i] = n then
      Return i
    EndIf
  EndFor
  Return 0
EndAlgorithm

```

D.

```

Algorithm A(x, first, last, n):
  For i ← first, last execute
    If x[i] = n then
      x[i] ← 3 * n
    EndIf
  EndFor
EndAlgorithm

```

16. Andrei a következő algoritmussal játszik, ahol n és m nemnulla természetes számok ($1 \leq n, m \leq 10^4$). Az $\text{abs}(x)$ algoritmus az x szám abszolútértékét téríti vissza.

```

Algorithm problema(n, m):
  b ← abs(m - n)
  c ← n - m
  If b - c = 0 then
    a ← n MOD m
  Else
    a ← (m + 2) MOD n
  EndIf
  Return a
EndAlgorithm

```

Andrei észreveszi, hogy függetlenül a specifikációnak megfelelő n értéktől, létezik legkevesebb két érték m számára, amikor a $\text{problema}(n, m)$ algoritmus 0-át térít vissza. Melyek m -nek ezek az értékei?

- A. 1 és n
- B. 1 és $n + 2$
- C. n és $n + 2$
- D. 1 és $n - 2$

17. Egy tanuló a backtracking módszerrel szeretné generálni az összes háromjegyű páratlan számot, amelyhez a számjegyeket a [4, 3, 8, 5, 7, 6] vektorból kell vennie az adott sorrendben. Tudva, hogy a generált első 5 szám ebben a sorrendben: 443, 445, 447, 433, 435, melyik lesz a 10-edik generált szám?

A. 487

B. 453

C. 457

D. 455

18. Legyen az $f(k, n, x)$ algoritmus, ahol k, n természetes számok ($1 \leq k, n \leq 10^3$) és x egy n elemű, természetes számokat tároló vektor ($x[1], x[2], \dots, x[n], 1 \leq x[i] \leq 10^4$, ha $i = 1, 2, \dots, n$).

```

Algorithm f(k, n, x):
  If n = 0 then
    Return 0
  Else
    d ← 0
    For i ← 2, x[n] DIV 2 execute
      If (x[n] MOD i) = 0 then
        d ← d + 1
      EndIf
    EndFor
    If d = k then
      Return 1 + f(k, n - 1, x)
    Else
      Return f(k, n - 1, x)
    EndIf
  EndIf
EndAlgorithm

```

A következő állítások közül melyek igazak?

A. Ha $x = [4, 9, 26, 121]$, az $f(1, 4, x)$ hívás eredménye 3.

B. Ha $x = [4, 8, 6, 144]$ az $f(2, 4, x)$ hívás eredménye 3.

C. Ha $x = [4, 9, 25, 144]$ az $f(1, 4, x)$ hívás eredménye 3.

D. Ha $x = [8, 27, 25, 121]$ az $f(2, 4, x)$ hívás eredménye 3.

19. Legyen a $check(n)$ algoritmus, ahol n természetes szám ($1 \leq n \leq 10^5$).

```

Algorithm check(n):
  While n > 0 execute
    If n MOD 3 > 1 then
      Return False
    EndIf
    n ← n DIV 3
  EndWhile
  Return True
EndAlgorithm

```

Állapítsátok meg az algoritmus hatását.

A. Ha n 3-nak hatványa, az algoritmus *True*-t térít vissza, különben *False*-t.

B. Ha n felírva 3-as számrendszerben csak 0-ás és 1-es számjegyeket tartalmaz, az algoritmus *True*-t térít vissza, különben *False*-t.

C. Ha n felírható 3-hatványként, vagy különböző 3-hatványok összegeként, az algoritmus *True*-t térít vissza, különben *False*-t.

D. Ha n felírva 3-as számrendszerben csak 2-es számjegyeket tartalmaz, az algoritmus *True*-t térít vissza, különben *False*-t.

20. Egy előadást egy adott I-es teremben kellett volna megtartani, de át kellett költöznie a II-es terembe, ahol a székek számozása különbözik. Mindkét teremben L széksor található ($2 \leq L \leq 50$), minden sort kettőbe oszt a sor közepén található folyosó, melynek mindkét oldalán K szék található ($2 \leq K \leq 50$) (tehát a teremben összesen $2 * K * L$ szék van).

Az I-es teremben minden helyet egyetlen szám azonosít be. A folyosó bal oldalán levő helyek páros számúak, és a székek számozása a színpad előtti sorban kezdődik. Tehát az első sorban levő székek számai (a folyosótól a terem széle fele haladva) 2, 4, 6 stb. Miután, egy adott sorban megszámoztak minden széket, a következő sorban folytatódik a számozás a folyosó melletti székekkel és a következő páros számmal. A folyosó jobb oldalán található helyeket hasonlóan számozták, de páratlan számokkal. Tehát, az első sorban levő székek számai (a folyosótól a terem széle fele haladva) 1, 3, 5 stb.

A II-es teremben minden helyhez három érték van rendelve. A sor sorszáma (egy 1 és L közötti szám, ahol az 1-es sor a színpad előtti), a folyosótól nézve a hely iránya („stanga” vagy „dreapta”) és a szék sorszáma a soron belül (egy 1 és K közötti szám, ahol az 1-es szék a folyosó melletti).

Mivel az előadásnak át kell költöznie, az I-es teremben érvényes jegyeken található egyetlen számot át kell alakítani úgy, hogy a II-es teremben érvényes helyeknek feleljen meg (ezeket a *rând*, *loc*, *direcție* jelöli).

Melyek azok az algoritmusok, amelyek az $L, K, nrLoc$ bemeneti adatok alapján elvégzik a kijelentésnek megfelelően helyesen az átalakítást? Egy átalakítás akkor helyes, ha minden néző helye egyedi a II-es teremben.

A.

```

Algorithm transforma(L, K, nrLoc):
  If nrLoc MOD 2 = 1 then
    directie ← "dreapta"
    nrLoc ← nrLoc + 1
  Else
    directie ← "stanga"
  EndIf
  If nrLoc MOD (2 * K) = 0 then
    rand ← nrLoc DIV (2 * K)
  Else
    rand ← nrLoc DIV (2 * K) + 1
  EndIf
  loc ← (nrLoc - (rand - 1) * 2 * K) DIV 2
  Return rand, loc, directie
EndAlgorithm

```

C.

```

Algorithm transforma(L, K, nrLoc):
  If nrLoc MOD 2 = 1 then
    directie ← "dreapta"
    nrLoc ← nrLoc + 1
  Else
    directie ← "stanga"
  EndIf
  rand ← nrLoc DIV (2 * K) + 1
  loc ← (nrLoc - (rand - 1) * 2 * K) DIV 2
  Return rand, loc, directie
EndAlgorithm

```

B.

```

Algorithm transforma(L, K, nrLoc):
  If nrLoc MOD 2 = 1 then
    directie ← "dreapta"
  Else
    directie ← "stanga"
  EndIf
  If nrLoc MOD (2 * K) = 0 then
    rand ← nrLoc DIV (2 * K)
  Else
    rand ← nrLoc DIV (2 * K) + 1
  EndIf
  loc ← (nrLoc - (rand - 1) * 2 * K) DIV 2
  Return rand, loc, directie
EndAlgorithm

```

D.

```

Algorithm transforma(L, K, nrLoc):
  If nrLoc MOD 2 = 1 then
    directie ← "dreapta"
    nrLoc ← nrLoc + 1
  Else
    directie ← "stanga"
  EndIf
  If nrLoc MOD (2 * K) = 0 then
    rand ← nrLoc DIV (2 * K)
  Else
    rand ← nrLoc DIV (2 * K) + 1
  EndIf
  loc ← (nrLoc - (rand - 1) * 2 * K) DIV 2 + 1
  Return rand, loc, directie
EndAlgorithm

```

21. Legyen a $p(x, n, k, final)$ algoritmus, ahol x egy $n + 1$ természetes számból álló vektor ($x[0], x[1], \dots, x[n]$). Eredetileg $x[i] = 0$, ha $i = 0, 1, \dots, n$. Az n és k változók nemnulla természetes számok ($1 \leq n, k \leq 20$) és **final** egy boolean típusú változó. Az $Afis(x, 1, n)$ algoritmus kiírja az $x[1], x[2], \dots, x[n]$ elemeket.

```

Algorithm p(x, n, k, final):
  While final = False execute
    While x[k] < n execute
      x[k] ← x[k] + 1
      If OK(x, k) = True then
        If k = n then
          Afis(x, 1, n)
        Else
          k ← k + 1
          x[k] ← 0
        EndIf
      EndIf
    EndWhile
  EndWhile
  _____ // itt kell kiegészíteni az algoritmust
EndAlgorithm

```

```

Algorithm OK(x, k):
  For i ← 1, k - 1 execute
    If x[k] = x[i] then
      Return False
    EndIf
  EndFor
  Return True
EndAlgorithm

```

Mely kódrészlettel kell kiegészíteni az algoritmust ahhoz, hogy ha $p(x, n, 1, False)$ alakban hívjuk meg, írjon ki minden n -ed rendű permutációt, mindegyiket egyszer.

A.

```

If k > 1 then
  k ← k - 1
Else
  final ← True
EndIf

```

B.

```

If k > 0 then
  k ← k - 1
Else
  final ← True
EndIf

```

C.

```

final ← True

```

D.

```

If k > 1 then
  k ← k - 1
  final ← True
EndIf

```

22. Adottak a problema(n) és a calcul(a, b) algoritmusok, ahol n, a, b természetes számok ($0 \leq n, a, b \leq 9$).

```

Algorithm problema(n):
    rezultat  $\leftarrow$  0
    For  $k \leftarrow 0, n$  execute
        For  $p \leftarrow 0, n$  execute
            For  $j \leftarrow 0, n$  execute
                If  $p \bmod 2 = 0$  then
                    rezultat  $\leftarrow$  rezultat + 1
                EndIf
            EndFor
        EndFor
    EndFor
    Return rezultat
EndAlgorithm

```

```

Algorithm calcul(a, b):
    t  $\leftarrow$  0
    For cifra  $\leftarrow a, b$  execute
        t  $\leftarrow$  t + problema(cifra)
    EndFor
    Write t
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. A calcul(1, 8) hívás eredményeként 1095 íródik ki.
- B. A calcul(1, 8) hívás eredményeként 1094 íródik ki.
- C. A calcul(0, 9) hívás eredményeként 1095 íródik ki.
- D. A calcul(0, 9) hívás eredményeként 1595 íródik ki.

23. Legyen a checkAcc(n, f, w, lw) algoritmus, ahol n egy nemnulla természetes szám ($1 \leq n \leq 10^4$), f természetes szám, w egy lw ($1 \leq lw \leq 10^4$) elemű, természetes számokat tároló sorozat ($w[1], w[2], \dots, w[lw]$, ahol $0 \leq w[p] \leq 10^4$, ha $p = 1, 2, \dots, lw$). A checkAcc(n, f, w, lw) algoritmus meghívja a $t(i, j, k)$ algoritmust, ahol i, j és k természetes számok. A $t(i, j, k)$ algoritmus boolean típusú eredményt térít vissza.

```

Algorithm checkAcc(n, f, w, lw):
    acc  $\leftarrow$  True
    If  $lw = 0$  AND  $f \neq 1$  then
        acc  $\leftarrow$  False
    Else
        index  $\leftarrow$  1
        q  $\leftarrow$  1
        While (acc = True) AND (index  $\leq$  lw) execute
            crt  $\leftarrow$  1
            changed  $\leftarrow$  False
            While (changed = False) AND (crt  $\leq$  n) execute
                If  $t(q, w[index], crt)$  then
                    q  $\leftarrow$  crt
                    changed  $\leftarrow$  True
                Else
                    crt  $\leftarrow$  crt + 1
                EndIf
            EndWhile
            If changed = False then
                acc  $\leftarrow$  False
            Else
                index  $\leftarrow$  index + 1
            EndIf
        EndWhile
        If (index > lw) AND (acc = True) AND (q  $\neq$  f) then
            acc  $\leftarrow$  False
        EndIf
    EndIf
    Return acc
EndAlgorithm

```

Az alábbi esetek közül melyikben térít vissza a checkAcc(2, f, w, lw) algoritmus True-t, tudva, hogy a $t(i, j, k)$ algoritmus True-t térít vissza a táblázatban megadott esetekben, különben False-t?

i	j	k
1	0	1
1	1	2
2	1	2

- A. $w = [0, 0, 1, 1]$, $lw = 4$ és $f = 1$
- B. $w = [1, 1, 1, 0]$, $lw = 4$ és $f = 2$
- C. $w = [0, 0, 1, 1]$, $lw = 4$ és $f = 2$
- D. $w = [0, 0, 0, 0]$, $lw = 4$ és $f = 1$

24. Legyen az $a = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$ számjegyekvektor. Abból a célból, hogy az a vektor elemeit egy más sorrendben írjuk ki, felépítjük az eredetileg üres b vektort. Minden lépésben a következő két művelet közül lehet egyet választani:

- *Ragaszt* – az a vektor első elemét a b vektor végére helyezi és törli azt az a vektorból.
- *Töröl* – kiírja, majd törli a b vektor utolsó elemét.

Megjegyzések:

- Az a vektor elemeit az adott sorrendben dolgozzuk fel.
- A *Ragaszt* művelet nem alkalmazható, ha az a vektor üres és a *Töröl*, ha a b vektor üres.
- A feldolgozás befejeződik, amikor az a és b vektorok üreseké váltak.

A fenti szabályoknak megfelelően, mely sorrendben **NEM** lehetséges kiírni a számjegyeket?

- A. 0 1 2 3 4 5 6 7 8 9
- B. 9 8 7 6 5 4 3 2 1 0
- C. 2 4 6 5 3 7 0 1 9 8
- D. 2 3 1 4 5 0 8 9 7 6

Felvételi verseny – 2023 július 19.

Informatika írásbeli

JAVÍTÁSI KULCS & MEGOLDÁSOK

HIVATALBÓL: 10 pont

1.	AB	3.75 pont
2.	C	3.75 pont
3.	B	3.75 pont
4.	AC	3.75 pont
5.	B	3.75 pont
6.	ABD	3.75 pont
7.	B	3.75 pont
8.	C	3.75 pont
9.	ABC	3.75 pont
10.	A	3.75 pont
11.	AD	3.75 pont
12.	D	3.75 pont
13.	C	3.75 pont
14.	ABC	3.75 pont
15.	AB	3.75 pont
16.	A	3.75 pont
17.	B	3.75 pont
18.	AC	3.75 pont
19.	BC	3.75 pont
20.	A	3.75 pont
21.	A	3.75 pont
22.	BD	3.75 pont
23.	CD	3.75 pont
24.	C	3.75 pont