

Felvételi verseny – 2025 szeptember 9
Informatika írásbeli

FONTOS MEGJEGYZÉS:

Más pontosítások hiányában:

- Az aritmetikai műveleteket végtelen adattípusokon végezzük (nincs túlsordulás és alulsordulás).
- Minden vektort, mátrixot és karakterláncot 1-től sorszámozunk (indexelünk).
- Az aktuális paraméterek értékeire vonatkozó megszorítások a kezdeti hívás pillanatában érvényesek.
- Egy vektor vagy karakterlánc tömbszakaszát a vektor vagy karakterlánc olyan elemei alkotják, amelyek egymás utáni pozíciókon találhatók.
- Ha ugyanabban a sorban több egymásutáni értékadó utasítás található, ezek ";" "-vel vannak elválasztva.

1. Adott a $\text{cauta}(a, n, b, m)$ algoritmus, ahol a és b két karakterlánc, n , illetve m karakterrel ($a[1], a[2], \dots, a[n], b[1], b[2], \dots, b[m], 1 \leq n, m \leq 100$ és $m \leq n$).

A $\text{cauta}(a, n, b, m)$ algoritmus következő implementációi közül melyik téríti vissza az a sornak azt a pozícióját, amelytől kezdődően a b sor először jelenik meg tömbszakaszként az a sorban, vagy -1-et, ha a b sor nem jelenik meg az a sorban?

A.

```
Algorithm cauta(a, n, b, m):  
  i ← 1  
  While i < n - m + 2 execute  
    j ← 1  
    While j ≤ m AND a[i + j - 1] = b[j] execute  
      j ← j + 1  
    EndWhile  
    If j > m then  
      Return i  
    EndIf  
    i ← i + 1  
  EndWhile  
  Return -1  
EndAlgorithm
```

C.

```
Algorithm cauta(a, n, b, m):  
  For i ← 1, n - m + 1 execute  
    k ← True; j ← 1  
    While k AND j ≤ m execute  
      If a[i + j - 1] ≠ b[j] then  
        k ← False  
      EndIf  
      j ← j + 1  
    EndWhile  
    If k then  
      Return i  
    EndIf  
  EndFor  
  Return -1  
EndAlgorithm
```

B.

```
Algorithm cauta(a, n, b, m):  
  If n = m then  
    Return 1  
  EndIf  
  For i ← 1, n - m + 1 execute  
    If a[i] = b[i] AND a[i + m - 1] = b[m] then  
      Return i  
    EndIf  
  EndFor  
  Return -1  
EndAlgorithm
```

D.

```
Algorithm cauta(a, n, b, m):  
  i ← 1  
  While i ≤ n - m execute  
    j ← 1  
    While j ≤ m AND a[i + j - 1] = b[j] execute  
      j ← j + 1  
    EndWhile  
    If j > m then  
      Return i  
    EndIf  
    i ← i + 1  
  EndWhile  
  Return -1  
EndAlgorithm
```

2. Adott az $S(n)$ algoritmus, ahol n egy egész szám ($0 \leq n \leq 10^4$).

```
Algorithm S(n):  
  If n = 0 then  
    Return 0  
  EndIf  
  Return n + S(n DIV 2)  
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. Az algoritmus időbonyolultsága $O(\log n)$.
- B. Az algoritmus az első n természetes szám összegét számítja ki.
- C. Az $S(4)$ hívás eredményeképpen az algoritmus 12-öt térít vissza.
- D. Az $S(2) + S(13) = S(7) + S(8)$ kifejezés értéke *True*.

3. A $\text{gcd}(a, b)$ algoritmus következő implementálásai közül melyik téríti vissza az a és b ($1 \leq a, b \leq 100$) természetes számok legnagyobb közös osztóját?

A.

```
Algorithm gcd(a, b):
  If b = 0 then
    Return b
  EndIf
  If a > b then
    Return gcd(a - b, b)
  Else
    Return gcd(a, b - a)
  EndIf
EndAlgorithm
```

C.

```
Algorithm gcd(a, b):
  While b > 0 execute
    c ← a; a ← b; b ← c MOD a
  EndWhile
  Return a
EndAlgorithm
```

B.

```
Algorithm gcd(a, b):
  If b = 0 then
    Return 0
  EndIf
  Return gcd(b, a MOD b)
EndAlgorithm
```

D.

```
Algorithm gcd(a, b):
  While a MOD b > 0 execute
    c ← a; a ← b; b ← c MOD a
  EndWhile
  Return b
EndAlgorithm
```

4. Adott az $f(n, x, k)$ algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^4$), x egy n elemű, egész számokat tároló vektor ($x[1], x[2], \dots, x[n], -10^3 \leq x[i] \leq 10^3, i = 1, 2, \dots, n$) és k egy természetes szám ($1 \leq k \leq 10^4$):

```
Algorithm f(n, x, k):
  For i ← 1, k execute
    a ← x[1]
    j ← 1
    While j < n execute
      x[j] ← x[j + 1]
      j ← j + 1
    EndWhile
    x[n] ← a
  EndFor
  Return x[1]
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. Ha $n = 10, x = [1, 2, 3, \dots, 10]$ és $k = 3$, az algoritmus 6-ot térít vissza.
- B. Ha $n = 10, x = [1, 2, 3, \dots, 10]$ és $k = 117$, az algoritmus 8-at térít vissza.
- C. Ha $n = 5, x = [4, 2, 5, 11, 13]$ és $k = 117$, az algoritmus 5-öt térít vissza.
- D. Ha $n = 5, x = [1, 2, 3, 4, 5]$, az algoritmus ugyanazt az értéket téríti vissza, mint akkor, ha $k = 3$ vagy $k = 318$.

5. Adott a $\text{ceFace}(n)$ algoritmus, ahol n természetes szám ($1 \leq n \leq 10^9$).

```
Algorithm ceFace(n):
  a ← 0; b ← 0
  For i ← 0, 9 execute
    x ← n
    c ← 0
    While x > 0 execute
      If x MOD 10 = i then
        c ← c + 1
      EndIf
      x ← x DIV 10
    EndWhile
    If c ≠ 0 then
      a ← a + 1
      If i MOD 2 = c MOD 2 then
        b ← b + 1
      EndIf
    EndIf
  EndFor
  Return a = b
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. A $\text{ceFace}(n)$ algoritmus *True*-t térít vissza, ha az n szám egymástól különböző páros számjegyeinek darabszáma egyenlő az egymástól különböző páratlan számjegyek darabszámával.
- B. Ha $n = 12235$, a $\text{ceFace}(n)$ algoritmus *True*-t térít vissza.
- C. Ha n csak páratlan számjegyekből áll, a $\text{ceFace}(n)$ algoritmus *True*-t térít vissza.
- D. Ha $n = 10^k$, ahol k páros szám, a $\text{ceFace}(n)$ algoritmus *True*-t térít vissza.

6. Adott a $\text{numar}(n)$ algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^6$).

Algorithm $\text{numar}(n)$:

```
While  $n > 9$  execute
   $n \leftarrow n \text{ DIV } 10 + n \text{ MOD } 10$ 
EndWhile
Return  $n$ 
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. Bármely n szám esetében az algoritmus által visszatérített érték az $\{1, 2, \dots, 9\}$ halmaz eleme.
- B. Az algoritmus nem térít vissza 0-át n egyetlen értékére sem.
- C. Létezik 3 különböző v érték, amelyeknek esetében, ha $\text{numar}(n) = v$, akkor n osztható v -vel, függetlenül n értékétől.
- D. Az algoritmus az n szám számjegyeinek összegét téríti vissza.

7. Legyen a $b = 110101011_{(2)}$ bináris szám.

A következő állítások közül melyek igazak a b számra vonatkozóan?

- A. A 4-es számrendszerben az értéke $12223_{(4)}$.
- B. A 8-as számrendszerben az ábrázolás palindrom.
- C. Az értéke páratlan szám.
- D. Az értéke 3-mal osztható.

8. Adott az $f(a, b, n, d)$ algoritmus, ahol a és b nemnull természetes számok ($1 \leq a, b \leq 10^9$), $n = 3$ és $d = [5, 7, 11]$:

Algorithm $f(a, b, n, d)$:

```
 $c \leftarrow 0$ ;  $p \leftarrow 1$ 
 $g \leftarrow \text{False}$ 
While NOT  $g$  execute
  For  $i \leftarrow 1, n$  execute
    If  $(a \text{ MOD } d[i] = 0) \text{ OR } (b \text{ MOD } d[i] = 0)$  then
       $c \leftarrow c * 10 + 1$ 
       $p \leftarrow p * i$ 
    EndIf
  EndFor
   $g \leftarrow \text{NOT } g$ 
EndWhile
Return  $c * 100 + p$ 
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. Ha $a * b$ nem osztható 5-tel, vagy 7-tel, vagy 11-gyel, az $f(a, b, 3, [5, 7, 11])$ hívás eredményeképpen 1 térítődik vissza.
- B. Ha $a * b$ nem osztható 5-tel, vagy 7-tel, vagy 11-gyel, az $f(a, b, 3, [5, 7, 11])$ hívás esetében a c változó utolsó értéke 10.
- C. Ha $a * b$ osztható 5-tel, 7-tel és 11-gyel, az $f(a, b, 3, [5, 7, 11])$ hívás eredményeképpen 11106 térítődik vissza.
- D. A $f(112233, 331122, 3, [5, 7, 11])$ hívás eredményeképpen 1003 térítődik vissza.

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

Algorithm $f(x, n)$:

```
For  $i \leftarrow 1, n - 1$  execute
  If  $x[i + 1] < x[i]$  then
     $\text{tmp} \leftarrow x[i + 1]$ 
     $x[i + 1] \leftarrow x[i]$ 
     $x[i] \leftarrow \text{tmp}$ 
  EndIf
EndFor
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. Az $f(x, n)$ hívás eredményeképpen az x vektor legnagyobb eleme az n . pozíción lesz.
- B. Az $f(x, n)$ hívás eredményeképpen az x vektor legkisebb eleme az n . pozíción lesz.
- C. Az $f(x, n)$ hívás eredményeképpen az x vektor növekvően rendezett lesz.
- D. Az $f(x, n)$ hívás eredményeképpen az x vektor csökkenően rendezett lesz.

10. Adott az $F(n)$ algoritmus, ahol n egy egész szám ($0 < n < 10^5$).

Algorithm $F(n)$:

```
If  $n = 0$  then
  Return 0
Else
  If  $n \text{ MOD } 2 = 0$  then
    Return  $F(n \text{ DIV } 10) + 1$ 
  Else
    Return  $F(n \text{ DIV } 10) - 1$ 
  EndIf
EndIf
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. Ha $n = 543$, az algoritmus 2-őt térít vissza.
- B. Ha $n = 18$, az algoritmus 0-át térít vissza.
- C. Ha $n = 41173$, az algoritmus 3-at térít vissza.
- D. A $[111, 999]$ intervallumhoz tartozó egyetlen n számra sem térít vissza 0-át az $F(n)$ algoritmus.

11. Adott a $\text{ceFace}(x, n)$ algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^4$) és x egy n elemű, egész számokat tároló vektor ($x[1], x[2], \dots, x[n]$, $0 \leq x[i] \leq 10^4$, $i = 1, 2, \dots, n$):

```

Algorithm ceFace(x, n):
  i ← 1
  c ← 0
  d ← 0
  While i ≤ n execute
    If (x[i] MOD 10) MOD 2 = 0 then
      c ← c + 1
    Else
      If (x[i] MOD 10) MOD 3 = 0 then
        d ← d + 1
      EndIf
    EndIf
    i ← i + 1
  EndWhile
  Return c = d
EndAlgorithm

```

A következő esetek közül melyekben térít vissza *True*-t a $\text{ceFace}(x, n)$ algoritmus?

- A. Ha a vektorban található páros számok darabszáma egyenlő a vektorban található páratlan számok darabszámával.
- B. Ha a vektorban található 2-vel osztható számok darabszáma egyenlő a vektorban található 6 többszöröseinek darabszámával.
- C. Ha a vektorban található páros számok darabszáma egyenlő a vektorban található olyan páratlan számok darabszámával, amelyeknek utolsó számjegye 3-nak többszöröse.
- D. Ha a vektorban található 2-vel osztható számok darabszáma egyenlő a vektorban található olyan számok darabszámával, amelyek 3-nak többszörösei.

12. Adott az $f(n, x, c)$ algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^4$), x egy n elemű, egész számokat tároló vektor ($x[1], x[2], \dots, x[n]$, $-10^3 \leq x[i] \leq 10^3$, $i = 1, 2, \dots, n$) és c egy természetes szám ($1 \leq c \leq n$):

```

Algorithm f(n, x, c):
  For i ← 1, c execute
    For j ← 1, n - 1 execute
      If x[j] > x[j + 1] then
        aux ← x[j]
        x[j] ← x[j + 1]
        x[j + 1] ← aux
      EndIf
    EndFor
  EndFor
  Return x[n + 1 - c]
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Az $f(5, [5, 4, 30, 5, 1], 3)$ hívás eredményeképpen az algoritmus 5-öt térít vissza.
- B. Ha $n = 100$ és x a 100, 99, 98, ..., 1 értékeket tárolja, valamint $c = 50$, az $f(n, x, c)$ algoritmus 51-et térít vissza.
- C. Ha $c = n$, az $f(n, x, c)$ algoritmus az x vektor legkisebb elemét téríti vissza.
- D. Ha $c = 1$, az $f(n, x, c)$ algoritmus az x vektor legkisebb elemét téríti vissza.

13. Adott az $\text{afla}(n, x)$ algoritmus, ahol n egy természetes szám ($3 \leq n \leq 10^4$) és x egy n elemű, egész számokat tároló vektor ($x[1], x[2], \dots, x[n]$, $-100 \leq x[i] \leq 100$, $i = 1, 2, \dots, n$).

```

Algorithm afla(n, x):
  M ← x[1]
  For i ← 1, n - 2 execute
    For j ← i + 1, n - 1 execute
      For k ← j + 1, n execute
        If M < x[i] + x[j] + x[k] then
          M ← x[i] + x[j] + x[k]
        EndIf
      EndFor
    EndFor
  EndFor
  Return M
EndAlgorithm

```

A következő állítások közül melyek igazak az $\text{afla}(n, x)$ meghívása után?

- A. Ha az x vektornak minden eleme negatív, az algoritmus az $x[1]$ elem értékét téríti vissza.
- B. Ha az x vektornak minden eleme pozitív, de kisebb vagy egyenlő $x[1]$ -gyel, az algoritmus az $x[1]$ elem értékét téríti vissza.
- C. Ha $x[1], x[2]$ és $x[3]$ pozitív számok, az algoritmus ezeknek az összegét téríti vissza.
- D. Bármely n elemű x vektor esetében, ahol az összes elem szorzata páratlan szám, az $\text{afla}(n, x)$ algoritmus egy páratlan számot térít vissza.

14. Adjátok meg, hogy hány olyan különböző irányítatlan gráfot lehet építeni, amelyeknek 5 darab, 1-től 5-ig számozott csomópontjuk van, és a 3-as csomópont foka 1. Két gráf különböző, ha az adjacencia (szomszédsági) mátrixuk különböző.

- A. 2^8
- B. $5!$
- C. $5 * C_5^4$
- D. 2^5

15. Adott a $\text{ceFace}(A, n)$ algoritmus, ahol n egy természetes szám ($1 < n \leq 10$) és A egy $n \times n$ elemű, természetes számokat tároló mátrix ($A[1][1], A[1][2], \dots, A[n][n]$, $1 \leq A[i][j] \leq 10, i, j = 1, 2, \dots, n$).

Algorithm $\text{ceFace}(A, n)$:

```

c ← 0
For i ← 2, 3 execute
    first ← modifica(A, n, i - 1, i, 2, 4)
    second ← modifica(A, n, i, i - 1, 3, 2)
    If first AND second then
        c ← c + 1
    EndIf
EndFor
Return c
EndAlgorithm

```

Algorithm $\text{modifica}(A, n, r, c, x, nr)$:

```

If (r + x - 1 > n) OR (c + x - 1 > n) then
    Return False
EndIf
For i ← r, r + x - 1 execute
    For j ← c, c + x - 1 execute
        A[i][j] ← A[i][j] * nr
    EndFor
EndFor
Return True
EndAlgorithm

```

Ha $n = 4$ és az eredeti A mátrix $\begin{pmatrix} 4 & 8 & 3 & 4 \\ 2 & 5 & 6 & 8 \\ 4 & 3 & 2 & 9 \\ 5 & 3 & 1 & 7 \end{pmatrix}$, a következő állítások közül melyek igazak a $\text{ceFace}(A, n)$ algoritmus meghívása után?

- A. A módosított mátrix főátlóján levő elemek összege 95.
- B. A módosított mátrixban kétszer több 2-vel osztható elem létezik, mint az eredeti mátrixban.
- C. A visszatérített érték 1.
- D. A módosított mátrix főátlóján levő elemek összege 67.

16. Adott a $\text{buildMatrix}(n, x, m, y)$ algoritmus, ahol n és m természetes számok ($1 \leq n, m \leq 100$), x és y n , illetve m elemű, egész számokat tároló vektorok ($x[1], x[2], \dots, x[n]$, ahol $-10^3 \leq x[i] \leq 10^3, i = 1, 2, \dots, n$ és $y[1], y[2], \dots, y[m]$, ahol $-10^3 \leq y[j] \leq 10^3, j = 1, 2, \dots, m$). A $\max(a, b)$ algoritmus az a és b számok közül a nagyobbát téríti vissza. A $\text{zeros}(n, m)$ algoritmus egy n sorral és m oszloppal rendelkező mátrixot térít vissza, ahol minden elem egyenlő 0-val.

Algorithm $\text{buildMatrix}(n, x, m, y)$:

```

A ← zeros(n + 1, m + 1)
For i ← 1, n execute
    For j ← 1, m execute
        If x[i] = y[j] then
            A[i + 1][j + 1] ← A[i][j] + 1
        Else
            A[i + 1][j + 1] ← max(A[i][j + 1], A[i + 1][j])
        EndIf
    EndFor
EndFor
Return A[n + 1][m + 1]
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. A $\text{buildMatrix}(3, [3, 2, 3], 3, [2, 3, 3])$ hívás eredményeképpen az algoritmus 2-t térít vissza.
- B. A $\text{buildMatrix}(n, x, m, y)$ algoritmus akkor és csakis akkor térít vissza 0-át, ha x és y hossza azonos és nincs egyetlen közös elemük sem.
- C. Ha $n = m$, és az y vektor elemei az x vektor elemeinek egy permutációját ábrázolja, akkor a $\text{buildMatrix}(n, x, m, y)$ algoritmus n -et térít vissza.
- D. Léteznek olyan bemeneti adatok, amelyekre az algoritmus $n + m$ értékét téríti vissza.

17. Adott a $h(A, n, p)$ algoritmus, ahol n és p természetes számok ($1 \leq n, p \leq 10^3$) és A egy n elemű, természetes számokat tároló vektor ($A[1], A[2], \dots, A[n]$, ahol $1 \leq A[i] \leq 100, i = 1, 2, \dots, n$):

Algorithm $h(A, n, p)$:

```

t ← n + 1; j ← 1; s ← 0
For i ← 1, n execute
    s ← s + A[i]
    While s > p execute
        If t > i - j + 1 then
            t ← i - j + 1
        EndIf
        s ← s - A[j]
        j ← j + 1
    EndWhile
EndFor
Return t
EndAlgorithm

```

Mely hívások eredményeképpen térít vissza az algoritmus 3-at?

- A. $h([2, 1, 5, 6, 2, 5], 6, 11)$
- B. $h([2, 1, 5, 6, 2, 5], 6, 10)$
- C. $h([7, 8, 1, 2], 4, 15)$
- D. $h([7, 8, 1, 2], 4, 16)$

18. Adott a `matrice(mat, n)` algoritmus, ahol *mat* egy *n* sorral és *n* oszloppal rendelkező ($3 \leq n \leq 200$), természetes számokat tároló mátrix ($mat[1][1], mat[1][2], \dots, mat[n][n], 1 \leq mat[i][j] \leq 10^4, i = 1, 2, \dots, n, j = 1, 2, \dots, n$). A "/" operátor a valós számok osztását jelöli; például $7 / 2 = 3.5$.

```
Algorithm matrice(mat, n):
  c ← 0
  For i ← 1, n execute
    For j ← 1, n execute
      ... // (1)
      c ← c + mat[i][j]
    EndIf
  EndFor
  ... // (2)
EndAlgorithm
```

Mely utasításokkal kell helyettesítenünk a `matrice(mat, n)` algoritmus (1)-gyel és (2)-vel jelölt sorait, ahhoz, hogy ez azoknak a számoknak az átlagát térítse vissza, amelyek nincsenek a mátrix átlóján?

A.

(1) If $i \neq j$ AND $i \neq n - j - 1$ then
(2) Return $c / (n * n - 2 * n + 1)$

B.

(1) If $i \neq j$ OR $i + j \neq n + 1$ then
(2) Return $c / (n * n - 2 * n)$

C.

(1) If $i \neq j$ AND $i + j \neq n + 1$ then
(2) Return $c / (n * n - 2 * n + n \text{ MOD } 2)$

D.

(1) If $i \neq j$ AND $i \neq n - j - 1$ then
(2) Return $c / (n * n - 2 * n + n \text{ MOD } 2)$

19. Adott az `f(x, n)` algoritmus, ahol *n* egy természetes szám ($2 \leq n \leq 10^3$), *x* egy *n* elemű, természetes számokat tároló vektor ($x[1], x[2], \dots, x[n], 0 \leq x[i] \leq 100, i = 1, 2, \dots, n$).

```
Algorithm f(x, n):
  left ← 1
  right ← n
  g ← True
  While left < right execute
    max ← -1
    For i ← left, right execute
      If  $x[i] > \text{max}$  then
        max ←  $x[i]$ 
        poz1 ← i
      EndIf
    EndFor
    If g then
      poz2 ← left
      left ← left + 1
    Else
      poz2 ← right
      right ← right - 1
    EndIf
    If poz1 ≠ poz2 then
       $x[\text{poz1}] \leftarrow x[\text{poz2}]$  // (*)
       $x[\text{poz2}] \leftarrow \text{max}$ 
    EndIf
    g ← NOT g
  EndWhile
  Return x
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. Egy adott *x*, *n* elemű vektor esetében az `f(x, n)` hívás ugyanazt az eredményt téríti vissza, mint a vektor elemeinek bármely permutációjára.
- B. Ha $n = 10$ és $x = [1, 2, 3, \dots, 10]$, az `f(x, n)` hívás következtében a (*)-gal jelölt sor 9-szer hajtódik végre.
- C. Ha az *x* vektor növekvően rendezett, az `f(x, n)` hívás következtében a (*)-gal jelölt sor ugyanannyiszor hajtódik végre, mint amikor az *x* vektor csökkenően rendezett.
- D. Ha $n = 100$ és $x = [1, 2, 3, \dots, 100]$, az *x* vektor elemeinek egyetlen olyan permutációja létezik, amelynek esetében a (*)-gal jelölt sor az `f(x, n)` hívás esetében nem hajtódik végre.

20. Adott a `rezultat(azi, zile)` algoritmus, ahol *azi* ($1 \leq azi \leq 5$) egy természetes szám, amely a hét egy munkanapját jelöli (az 1 hétfőnek, a 2 keddeknek, ..., illetve az 5 pénteknek felel meg) és *zile* ($1 \leq zile \leq 100$) egy természetes szám.

A `rezultat(azi, zile)` algoritmus a hétnak azt a munkanapját téríti vissza, amelyen közlik egy vizsgálat eredményét, tudva azt, hogy az adott vizsgálatot *zile* naptári nap alatt végzik el. Vizsgálatokat a hét minden napján végeznek, de az eredményt az ezt követő első munkanapon közlik. Ha a vizsgálat a hétvégén készült el (szombaton vagy vasárnap), az eredményt a következő hétfőn közlik.

Például, a `rezultat(2, 1)` hívás eredményeképpen az algoritmus 2-öt térít vissza (az eredményt aznap közlik, amikor elkészült), a `rezultat(3, 4)` hívás eredményeképpen az algoritmus 1-et térít vissza (az eredményt hétfőn közlik).

Algorithm rezultat(azi, zile): A következő állítások közül melyek igazak?

```

For i ← 2, zile execute
    azi ← azi + 1
    If azi > 7 then
        azi ← 1
    EndIf
EndFor
If azi ≥ 6 then
    azi ← 1
EndIf
Return azi
EndAlgorithm

```

A. A rezultatRec(azi, zile) algoritmus ugyanazt az eredményt téríti vissza, mint a rezultat(azi, zile) algoritmus, bármely bemeneti adatra.

```

Algorithm rezultatRec(azi, zile):
    If zile = 0 then
        If azi > 5 then
            Return 1
        EndIf
        Return azi
    EndIf
    Return rezultatRec(azi + 1, zile - 1)
EndAlgorithm

```

B. A rezultat(4, 25) hívás eredményeként az algoritmus 1-et térít vissza.

C. A rezultat2(azi, zile) algoritmus ugyanazt az eredményt téríti vissza, mint a rezultat(azi, zile) algoritmus, bármely bemeneti adatra, és az időbonyolultsága kisebb:

```

Algorithm rezultat2(azi, zile):
    v ← ((azi + zile - 2) MOD 7) + 1
    If v > 5 then
        v ← 1
    EndIf
    Return v
EndAlgorithm

```

D. A rezultat(azi, zile) algoritmus időbonyolultsága $O(1)$.

21. Adott a suma(A , n , m , x_1 , y_1 , x_2 , y_2) algoritmus, amely az A mátrix egy adott tartománya elemeinek összegét számítja ki. A egy $n \times m$ dimenziójú mátrix ($A[1][1]$, $A[1][2]$, ..., $A[1][m]$, $A[2][1]$, ..., $A[n][m]$, ahol $2 < n, m < 100$, $0 \leq A[i][j] \leq 10^3$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$). Az érdeklét téglalap alakú tartományt a bal felső sarkának koordinátái (x_1, y_1) és a jobb alsó sarkának koordinátái (x_2, y_2) határozzák meg, ahol $1 \leq x_1 < x_2 \leq n$, $1 \leq y_1 < y_2 \leq m$. A zeros(a , b) algoritmus egy a sorral és b oszloppal rendelkező mátrixot téríti vissza, ahol minden elem kezdőértéke 0.

```

Algorithm compute( $A$ ,  $n$ ,  $m$ ):
    aux ← zeros( $n$ ,  $m$ )
    For i ← 1,  $n$  execute
        For j ← 1,  $m$  execute
            aux[i][j] ←  $A[i][j]$ 
            If j > 1 then
                aux[i][j] ← aux[i][j] + aux[i][j - 1]
            EndIf
            If i > 1 then
                aux[i][j] ← aux[i][j] + aux[i - 1][j]
            EndIf
            If j > 1 AND i > 1 then
                aux[i][j] ← aux[i][j] - aux[i - 1][j - 1]
            EndIf
        EndFor
    EndFor
    Return aux
EndAlgorithm

```

```

1. Algorithm suma( $A$ ,  $n$ ,  $m$ ,  $x_1$ ,  $y_1$ ,  $x_2$ ,  $y_2$ ):
2.     aux ← compute( $A$ ,  $n$ ,  $m$ )
3.     S1 ← 0; S2 ← 0; S3 ← 0; S4 ← 0
4.     If  $x_1 > 1$  AND  $y_1 > 1$  then
5.         S1 ← aux[ $x_1 - 1$ ][ $y_1 - 1$ ]
6.     EndIf
7.     If  $x_1 > 1$  then
8.         S2 ← aux[ $x_1 - 1$ ][ $y_2$ ]
9.     EndIf
10.    If  $y_1 > 1$  then
11.        S3 ← aux[ $x_2$ ][ $y_1 - 1$ ]
12.    EndIf
13.    S4 ← aux[ $x_2$ ][ $y_2$ ]
14.    .....
15. EndAlgorithm

```

Mely utasítást kell beszúrunk a 14. sorba, ahhoz, hogy a suma(A , n , m , x_1 , y_1 , x_2 , y_2) algoritmus a kért összeget térítse vissza?

- A. **Return** S4 - S2 - S3 + S1
- B. **Return** S4 + S2 + S3 + S1
- C. **Return** S4 - S2 + S3 + S1
- D. **Return** S4 - S2 - S3 - S1

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

```
Algorithm f(n, x):
  For i ← 1, n - 1 execute
    x[i + 1] ← (x[i + 1] * x[i]) DIV g(x[i], x[i + 1])
  EndFor
  Return x[n]
EndAlgorithm
```

```
Algorithm g(a, b):
  If a = b then Return a
  EndIf
  If a > b then Return g(a - b, b)
  EndIf
  Return g(a, b - a)
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. Az $f(5, [12, 16, 8, 40, 24])$ hívás eredményeképpen az algoritmus 240-et térít vissza.
- B. A $g(a, b)$ algoritmus összes hívásainak száma nagyobb az $f(5, [8, 12, 16, 24, 40])$ hívás esetében, mint az $f(5, [40, 24, 16, 12, 8])$ hívás esetében.
- C. Az algoritmus az x vektor összes elemének legnagyobb közös osztóját számítja ki.
- D. Ha az $n = 10$ elemű x vektor egyetlen 17-es értéket tartalmaz, a többi értéke pedig egyenlő 1-gyel, a $g(a, b)$ algoritmus hívásainak maximális száma, beleértve az eredeti hívást is, 153.

23. Az **A** és **B** csapatok közötti labdarúgó mérkőzésen, amelyen 3 gól születik és a mérkőzés az **A** csapat győzelmével 2 - 1-re zárul, a gólok 3 különböző sorrendben születhetnek:

1. 0 - 0, 0 - 1, 1 - 1, 2 - 1
2. 0 - 0, 1 - 0, 1 - 1, 2 - 1
3. 0 - 0, 1 - 0, 2 - 0, 2 - 1

Például, az 1.-es lehetőség szerint a mérkőzés 0 - 0-ás állásról indul, a **B** csapat gólt szerez (0 - 1), az **A** csapat egyenlít (1 - 1), majd az **A** csapat újabb gólt szerez (2 - 1).

Hány lehetséges sorrend létezik egy 4 gólos mérkőzés esetében?

- A. 4!
- B. 16
- C. 64
- D. $1! + 2! + 3! + 2! + 1!$

24. Adott a $litere(v, n)$ algoritmus, ahol v egy n elemű karakterlánc ($1 \leq n \leq 100$), amelynek elemei az angol ábécé nagybetűi. A karakterlánc minden i . ($1 \leq i \leq n$) pozícióján levő karakter esetében kiszámítódik egy $S[i]$ érték, amely azoknak a $j \in \{1, 2, \dots, i\}$ pozícióknak a számát jelenti, amelyekben olyan betűk találhatók, amelyek az ábécében a $v[i]$ betű előtt vannak. Például, ha $v = ['E', 'X', 'A', 'M', 'E', 'N']$, $S[6] = 4$, mivel az 'E', 'A' és 'M' betűk az 'N' betű előtt vannak az ábécében. Karakter típus esetében az összehasonlítás és a kivonás az operandusok ASCII kódjával történik.

A következő algoritmusok közül melyik téríti vissza az összes $S[i]$ ($i = 1, 2, \dots, n$) érték összegét?

A.

```
Algorithm litere(v, n):
  sum ← 0
  For i ← 2, n execute
    For j ← 1, i execute
      If v[j] - 'A' > v[i] - 'A' then
        sum ← sum + 1
      EndIf
    EndFor
  EndFor
  Return n * (n - 1) DIV 2 - sum
EndAlgorithm
```

C.

```
Algorithm litere(v, n):
  sum ← 0
  For i ← 2, n - 1 execute
    For j ← i + 1, n execute
      If v[j] - 'A' < v[i] - 'A' then
        sum ← sum + 1
      EndIf
    EndFor
  EndFor
  Return sum
EndAlgorithm
```

B.

```
Algorithm litere(v, n):
  sum ← 0
  For i ← 2, n execute
    For j ← 1, i execute
      If v[j] < v[i] then
        sum ← sum + 1
      EndIf
    EndFor
  EndFor
  Return sum
EndAlgorithm
```

D.

```
Algorithm litere(v, n):
  sum ← 0
  For i ← 1, n execute
    For j ← 1, i execute
      If v[j] < v[i] - 'A' then
        sum ← sum + (v[i] - v[j])
      EndIf
    EndFor
  EndFor
  Return sum
EndAlgorithm
```

BABEŞ-BOLYAI TUDOMÁNYEGYETEM
MATEMATIKA ÉS INFORMATIKA KAR

Felvételi verseny – 2025 szeptember 9

Informatika írásbeli

JAVÍTÁSI KULCS & MEGOLDÁSOK

HIVATALBÓL: 10 pont

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