

Felvételi verseny – 2026. július 17
Informatika írásbeli

FONTOS MEGJEGYZÉS:

Más pontosítások hiányában:

- Az aritmetikai műveleteket végtelen adattípusokon végezzük (nincs túlcsordulás és alulcsordulás).
- 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 ceface(n) algoritmus, ahol n egy természetes szám ($0 \leq n \leq 10^3$) és az $\&$ a bitenkénti **AND** műveletet jelöli.

```
Algorithm ceface(n):  
  If (n & 1)  $\neq$  0 then  
    Return 0  
  EndIf  
  Return 1  
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. Ha n kettőhatvány, az algoritmus 1-et térít vissza.
- B. Ha n páros szám, az algoritmus 1-et térít vissza.
- C. Az algoritmus akkor és csakis akkor térít vissza 0-át, ha n páratlan szám.
- D. Ha $n = 0$, az algoritmus 0-át térít vissza.

2. Adott a gramana(n, x) algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^3$), és x egy n elemű, kisbetűkből álló karakterlánc ($x[1], x[2], \dots, x[n], i = 1, 2, \dots, n$).

```
Algorithm gramana(n, x):  
  For i  $\leftarrow$  n, 1, -1 execute  
    For j  $\leftarrow$  2, i execute  
      If  $x[j - 1] > x[j]$  then  
        car  $\leftarrow$   $x[j]$   
         $x[j] \leftarrow x[j - 1]$   
         $x[j - 1] \leftarrow$  car  
      EndIf  
    EndFor  
  EndFor  
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. Az algoritmus időbonyolultsága $O(n^2)$.
- B. Az algoritmus az adott karakterláncnak egy körkörös permutációját valósítja meg.
- C. Az algoritmus az x karakterláncot lexikográfikusan növekvő sorrendbe rendezi.
- D. Az algoritmus az x karakterláncot lexikográfikusan csökkenő sorrendbe rendezi.

3. Adott az $101100110000_{(2)}$ bináris szám. Döntsék el, hogy a következő állítások közül melyek igazak?

- A. A szám osztható $16_{(10)}$ -tal.
- B. A szám kisebb, mint $10000_{(10)}$
- C. A szám páros.
- D. A szám osztható $32_{(10)}$ -vel.

4. Adott a test(n) algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^3$).

```
Algorithm test(n):  
  c  $\leftarrow$  1  
  While n  $>$  0 execute  
    n  $\leftarrow$  n - 2  
    c  $\leftarrow$  c * 2  
  EndWhile  
  Return c  
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. A test(n) hívás eredményeképpen az algoritmus mindig egy n -nél nagyobb értéket térít vissza.
- B. A test(n) hívás eredményeképpen a visszatérített érték mindig kisebb, mint a test($n + 1$) hívás eredményeképpen visszatérített érték.
- C. Ha két különböző természetes szám, n és m , párossága azonos, a test(n) és test(m) hívás eredményeképpen visszatérített értékek különbözők.
- D. A test(n) algoritmus időbonyolultsága $O(\log n)$.

5. Adott a $\text{mister}(x, n)$ algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^3$), és x egy n elemű, természetes számokat tartalmazó vektor ($x[1], x[2], \dots, x[n]$, $10 \leq x[i] \leq 10^5$, $i = 1, 2, \dots, n$).

```

Algorithm mister(x, n):
  For i ← 1, n - 1 execute
    For j ← i + 1, n execute
      a ← x[i]
      b ← x[j]
      While a > 10 execute
        a ← a DIV 10
      EndWhile
      While b > 10 execute
        b ← b DIV 10
      EndWhile
      If a < b then
        aux ← x[i]
        x[i] ← x[j]
        x[j] ← aux
      EndIf
    EndFor
  EndFor
  Return x
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. A $\text{mister}(x, n)$ algoritmus az elemek első számjegye alapján csökkenő sorrendbe rendezi az x vektort.
- B. A $\text{mister}([345, 21, 98, 412, 67], 5)$ hívás eredményeképpen az algoritmus a $[98, 67, 412, 345, 21]$ vektort téríti vissza.
- C. A $\text{mister}(x, n)$ algoritmus a számok utolsó számjegye alapján növekvő sorrendbe rendezi az x vektort.
- D. A $\text{mister}([10, 305, 47, 89], 4)$ hívás eredményeképpen az algoritmus a $[89, 47, 305, 10]$ vektort téríti vissza.

6. Egy sportversenyen n ($2 \leq n \leq 100$) játékos vesz részt. Minden mérkőzésen két sportoló játszik, és minden mérkőzés vesztesét kizárják a versenyből.

Hány mérkőzést kell lejátszani ahhoz, hogy megállapítható legyen a verseny győztese?

- A. $n - 1$
- B. n
- C. $n \text{ DIV } 2$
- D. $(n \text{ DIV } 2) + 1$

7. Egy logikai kifejezés, amely **OR** operátorral összekapcsolt tagokból áll ($t_1 \text{ OR } t_2 \text{ OR } \dots \text{ OR } t_n$) balról jobbra haladva lesz kiértékelve, és a kiértékelés csak addig tart, amíg megjelenik az első *True* értékű tag, ha létezik ilyen (ebben az esetben a kifejezés értéke *True* lesz és a többi tag nem lesz kiértékelve). Különben, ha minden tag értéke *False*, a kifejezés értéke *False*.

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

```

Algorithm test(n):
  Write '*'
  If n < 10 then
    Return False
  EndIf
  If n MOD 2 = 0 then
    Return True
  EndIf
  If n MOD 5 = 0 then
    Return True
  EndIf
  Return test(n DIV 2) OR test(n DIV 5) OR test(n DIV 10) //#
EndAlgorithm

```

Feltételezve, hogy a „#”-tel jelölt sor végrehajtásakor, ahol három **OR** operátorral összekötött logikai kifejezés található, a fent leírtak szerint járunk el, hány '*' karaktert ír ki az algoritmus a $\text{test}(159)$ hívás eredményeképpen?

- A. 10
- B. 15
- C. 3
- D. 8

8. Adott az $f(n)$ algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^3$).

```

Algorithm f(n):
  If n = 1 then
    Return 0
  EndIf
  Return f(n DIV 2) + 1
EndAlgorithm

```

Mit térít vissza az algoritmus az $f(n)$ hívás eredményeképpen?

- A. A $\log_2 n$ érték egész részét.
- B. Annak a minimális számú bitnek a számát, amely az n szám 2-es számrendszerbeli ábrázolásához szükséges.
- C. Az n szám 2-es számrendszerbeli ábrázolásában szereplő 1-es bitek számát.
- D. Egy n -től különböző értéket.

9. Adott az $ff(n, v)$ algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^4$), és v egy n elemű, egész számokat tartalmazó vektor ($v[1], v[2], \dots, v[n]$, $-100 \leq v[i] \leq 100$, $i = 1, 2, \dots, n$).

```

Algorithm ff(n, v):
  c ← 0
  For j ← 2, n execute
    a ← v[j]
    i ← j - 1
    While i > 0 AND v[i] ≥ a execute
      v[i + 1] ← v[i]
      i ← i - 1
      c ← c + 1
    EndWhile
    v[i + 1] ← a
  EndFor
  Return c
EndAlgorithm

```

Mely értéket téríti vissza az algoritmus az $ff(5, [-4, 1, 7, -8, 1])$ hívás eredményeképpen?

- A. 4
- B. 2
- C. 5
- D. 8

10. Adott az $f(a, b)$ algoritmus, ahol a és b természetes számok ($1 \leq a, b \leq 10^5$).

```

Algorithm f(a, b):
  If b = 0 then
    Return 0
  EndIf
  If b MOD 2 = 0 then
    Return f(a + a, b DIV 2)
  Else
    Return f(a + a, b DIV 2) + a
  EndIf
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Az algoritmus az $a * b$ szorzatot téríti vissza.
- B. Az algoritmus időbonyolultsága $O(\log b)$.
- C. Az $f(2, 3)$ hívás eredményeképpen az algoritmus 8-at térít vissza.
- D. Az algoritmus az a^b értéket téríti vissza.

11. Adott a $ceFace(a, n, s)$ algoritmus, ahol a egy n ($1 \leq n \leq 20$) elemű, természetes számokat tartalmazó, növekvően rendezett vektor ($a[1], a[2], \dots, a[n]$, $0 \leq a[i] \leq 10^3$, $1 \leq i \leq n$), és s egy természetes szám ($0 \leq s \leq 10^3$).

```

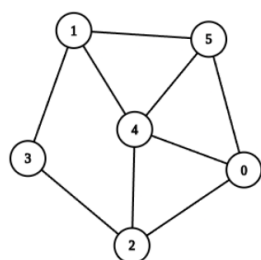
Algorithm ceFace(a, n, s):
  c ← 0
  For i ← 1, n - 2 execute
    j ← i + 1
    k ← n
    While j < k execute
      sum ← a[i] + a[j] + a[k]
      If sum = s then
        c ← c + 1
        j ← j + 1
        k ← k - 1
      Else
        If sum < s then
          j ← j + 1
        Else
          k ← k - 1
        EndIf
      EndIf
    EndWhile
  EndFor
  Return c
EndAlgorithm

```

Mely hívások esetében lesz a visszatérített érték 3?

- A. $ceFace([1, 2, 3, 4, 5, 6], 6, 10)$
- B. $ceFace([1, 2, 3, 4, 4, 5, 6], 7, 10)$
- C. $ceFace([1, 2, 3, 4, 4, 5, 6], 7, 8)$
- D. $ceFace([1, 1, 1, 2, 2, 2], 6, 4)$

12. Legyen a következő gráf:



A következő csomópontokat tartalmazó sorozatok közül melyik ábrázol Hamilton-kört?

- A. 0 1 2 3 4 5 0
- B. 1 3 2 0 4 5 1
- C. 1 3 2 4 2 0 5
- D. 4 2 3 1 5 0 4

13. Práslea cel Voinic-nak a kertjében van egy elvarázsolt gyümölcsfa, amelyen 27 aranyalma és 24 ezüstalma található. A sárkány minden nap ellop két almát, ugyanakkor a fán nő egy másik alma: ha a sárkány két azonos típusú almát vesz el (mindkettő aranyalma, vagy mindkettő ezüstalma), helyettük egy ezüstalma nő, ha a sárkány egy aranyalmát és egy ezüstalmát vesz el, akkor helyettük egy aranyalma nő.

A következő állítások közül melyek igazak?

- A. Az 50. napon a sárkány egy aranyalmát és egy ezüstalmát vett el.
- B. Nem lehet megállapítani, hogy milyen típusú almákat vett el a sárkány az 50. napon.
- C. Létezik a lopásoknak egy olyan sorozata, amelynek során egy adott pillanatban a fán kizárólag ezüstalmák vannak.
- D. A leírt események során a fán található összes alma száma pontosan 1-gyel csökken minden nap.

14. Adott az $f(X, n, k)$ algoritmus, ahol n és k természetes számok ($1 \leq k \leq n \leq 10$), és X egy n elemű, természetes számokat tartalmazó vektor ($X[1], X[2], \dots, X[n]$).

```

Algorithm f(X, n, k):
  If k = n + 1 then
    aux ← 0; ok ← True
    For j ← 1, n execute
      If X[j] = 1 then
        aux ← aux + 1
      EndIf
      If j > 1 AND X[j] = 1 AND X[j - 1] = 1 then
        ok ← False
      EndIf
    EndFor
    If ok AND aux = n DIV 2 then
      For j ← 1, n execute
        Write X[j]
      EndFor
      Write New Line
    EndIf
  Else
    For i ← 0, 1 execute
      X[k] ← i
      f(X, n, k + 1)
    EndFor
  EndIf
EndAlgorithm

```

Az $f([0, 0, 0, 0], 4, 1)$ hívás eredményeképpen az első kiírt sor: 0101. Mi lesz kiírva a következő két sorba?

- A.
 - 1001
 - 1100
- B.
 - 1010
 - 0110
- C.
 - 0110
 - 1010
- D.
 - 1001
 - 1010

15. Adott az $LU(A, n)$ algoritmus, ahol A egy n méretű, természetes számokat tartalmazó négyzetes mátrix ($A[1][1], A[1][2], \dots, A[1][n], \dots, A[n][n]$, $0 < A[i][j] < 10^3$, $i, j = 1, \dots, n$), és n egy természetes szám ($2 \leq n \leq 10$). A zerosMat(n) algoritmus egy n méretű négyzetes mátrixot térít vissza, amelyben minden elem értéke 0.

```

Algorithm LU(A, n):
  L ← zerosMat(n); U ← zerosMat(n)
  For i ← 1, n execute
    L[i][i] ← 1
    For j ← i, n execute
      sum ← 0
      For k ← 1, i - 1 execute
        sum ← sum + L[i][k] * U[k][j]
      EndFor
      U[i][j] ← A[i][j] - sum
    EndFor
    For j ← i + 1, n execute
      sum ← 0
      For k ← 1, i - 1 execute
        sum ← sum + L[j][k] * U[k][i]
      EndFor
      L[j][i] ← (A[j][i] - sum) DIV U[i][i]
    EndFor
  EndFor
  Return L, U
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Az $LU(A, n)$ hívás eredményeképpen a visszatérített L mátrixban a főátló fölött minden elem értéke 0 lesz.
- B. Az algoritmus időbonyolultsága $O(n^2)$
- C. Ha $A = \begin{pmatrix} 2 & 1 & 2 \\ 4 & 6 & 3 \\ 4 & 2 & 8 \end{pmatrix}$ és $n = 3$, az $LU(A, n)$ hívás eredményeképpen a visszatérített L és U mátrixok értékei $L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix}$ és $U = \begin{pmatrix} 2 & 1 & 2 \\ 0 & 4 & -1 \\ 0 & 0 & 4 \end{pmatrix}$ lesznek.
- D. Az $LU(\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, 2)$ hívás eredményeképpen a visszatérített U mátrix főátlóján található elemek szorzata egyenlő az A mátrix determinánsával.

16. Adott az $f(x, y, v)$ algoritmus, ahol x, y és v természetes számok ($1 \leq x, y, v \leq 100, x \leq v$).

```

Algorithm f(x, y, v):
  If x ≥ v then
    Write '1'
  Else
    If x MOD y ≠ 0 then
      f(x + 1, y, v - 1)
    Else
      If y MOD v ≠ 0 then
        f(x, y + 1, v - 1)
        Write '*'
      Else
        f(x + 1, y, v - 2)
        Write '2'
      EndIf
    EndIf
  EndIf
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Az $f(3, 5, 10)$, $f(5, 6, 10)$ és $f(2, 5, 9)$ hívások eredményeképpen az algoritmus azonos eredményt ír ki.
- B. Az $f(2, 2, 15)$ hívás eredményeképpen az algoritmus "1*****"-t ír ki.
- C. Az $f(2, 3, 14)$ hívás eredményeképpen az algoritmus egy olyan eredményt ír ki, amely tartalmazza a 2-es számjegyet.
- D. Az $f(2, 2, 11)$, $f(1, 2, 9)$ és $f(3, 7, 9)$ hívások eredményeképpen az algoritmus azonos eredményt ír ki.

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

```

Algorithm f(n, x):
  If x[n] - x[n - 1] ≠ x[n - 2] then
    Return 0
  EndIf
  k ← 3
  For i ← n - 1, 3, -1 execute
    If x[i] - x[i - 1] ≠ x[i - 2] then
      Return k
    EndIf
    k ← k + 1
  EndFor
  Return k
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Az $f(4, [1, 1, 2, 3])$ hívás eredményeképpen az algoritmus 4-et térít vissza.
- B. Az algoritmus akkor és csakis akkor téríti vissza n értékét, ha az x vektor értékei a Fibonacci-sorozat egy tömbszakaszában találhatóak.
- C. Nem létezik olyan 3 elemű vektor, amelyre az algoritmus az 1 értéket térítse vissza.
- D. Ha az algoritmus az n értéket téríti vissza, akkor az x vektor értékei növekvően rendezettek.

18. Adott a $verifica(s1, n1, s2, n2)$ algoritmus, ahol $s1$ és $s2$ csak az angol ábécé kisbetűit tartalmazó karakterláncok, amelyeknek a hossza $n1$, illetve $n2$ ($1 \leq n1, n2 \leq 100$). A $zerosVec(n)$ algoritmus egy n elemű, természetes számokat tartalmazó vektort térít vissza, amelyben minden elem 0-val van inicializálva.

```

Algorithm verifica(s1, n1, s2, n2):
  If n1 ≠ n2 then
    Return False
  EndIf
  f ← zerosVec(26)
  For i ← 1, n1 execute
    p ← s1[i] - 'a' + 1
    f[p] ← f[p] + 1
  EndFor
  For i ← 1, n2 execute
    p ← s2[i] - 'a' + 1
    ...
    If f[p] < 0 then
      Return False
    EndIf
  EndFor
  Return True
EndAlgorithm

```

A következő utasítások közül melyikkel helyettesíthető a „...” ahhoz, hogy a $verifica(s1, n1, s2, n2)$ algoritmus akkor és csakis akkor térítsen vissza *True*-t, ha az $s2$ karakterlánc előállítható az $s1$ karakterláncból (átrendezve $s1$ karaktereit)?

- A. $f[p] \leftarrow f[p] - 1$
- B. $f[p] \leftarrow f[p] + 1$
- C. $f[p] \leftarrow f[p] - i$
- D. $f[p] \leftarrow (f[p] + 1) \text{ MOD } 2$

19. Adott a `produs_str(n1, n2)` algoritmus, ahol ***n1*** és ***n2*** olyan karakterláncok, amelyek nemnulla természetes számokat ábrázolnak (pl. ***n1*** = "123" a 123-as számot ábrázolja). Az algoritmus a két szám szorzatát téríti vissza karakterlánc formájában. Például, a `produs_str("2", "3")` hívás eredménye "6", a `produs_str("123", "456")` pedig "56088"-at térít vissza.

A `len(sir)` algoritmus a ***sir*** karakterlánc hosszát téríti vissza. Az `int(car)` algoritmus a ***car*** karakternek megfelelő számjegyet téríti vissza. A `str(nr)` algoritmus a ***nr*** számnak megfelelő karakterláncot téríti vissza (pl. `str(123)` a "123"-at téríti vissza). Az `add(n, m)` algoritmus az ***n*** és ***m***, karakterlánc formájában ábrázolt számok összegét téríti vissza (pl. `add("123", "457")` az 580-as számot téríti vissza). A `zero(k)` algoritmus egy ***k*** elemű vektort térít vissza, amelyben minden elem értéke 0. A + operátor a karakterláncok konkatenálását (összefűzését) jelöli.

A következő változatok közül melyek a `produs_str(n1, n2)` algoritmus helyes implementálásai?

- A.
- ```

Algorithm produs_str(n1, n2):
 rez ← ""
 For i ← 1, len(n1) execute
 For j ← 1, len(n2) execute
 produs ← int(n1[i]) * int(n2[j])
 rez ← rez + str(produs)
 EndFor
 EndFor
 Return rez
EndAlgorithm

```
- B.
- ```

Algorithm produs_str(n1, n2):
    rez ← "0"
    For i ← len(n1), 1, -1 execute
        cif ← int(n1[i])
        For j ← 1, cif execute
            rez ← str(add(rez, n2))
        EndFor
    EndFor
    Return rez
EndAlgorithm

```
- C.
- ```

Algorithm produs_str(n1, n2):
 rez ← zero(len(n1) + len(n2))
 For i ← len(n1), 1, -1 execute
 For j ← len(n2), 1, -1 execute
 produs ← int(n1[i]) * int(n2[j])
 poz1 ← i + j - 1
 poz2 ← i + j
 sum ← produs + rez[poz2]
 rez[poz2] ← sum MOD 10
 rez[poz1] ← rez[poz1] + (sum DIV 10)
 EndFor
 EndFor
 rez_str ← ""
 For i ← 1, len(n1) + len(n2) execute
 digit ← rez[i]
 If NOT ((len(rez_str) = 0) AND (digit = 0))
 then
 rez_str ← rez_str + str(digit)
 EndIf
 EndFor
 Return rez_str
EndAlgorithm

```
- D.
- ```

Algorithm produs_str(n1, n2):
    rez ← zero(len(n1) + len(n2))
    For i ← len(n1), 1, -1 execute
        cifTransp ← 0
        For j ← len(n2), 1, -1 execute
            produs ← int(n1[i]) * int(n2[j]) +
                cifTransp + rez[i + j]
            rez[i + j] ← produs MOD 10
            cifTransp ← produs DIV 10
        EndFor
        rez[i] ← rez[i] + cifTransp
    EndFor
    rez_str ← ""
    For i ← 1, len(n1) + len(n2) execute
        digit ← rez[i]
        If NOT ((len(rez_str) = 0) AND (digit = 0))
            then
                rez_str ← rez_str + str(digit)
        EndIf
    EndFor
    Return rez_str
EndAlgorithm

```

20. Adott a `P(s, n)` algoritmus, ahol ***s*** egy ***n*** hosszúságú karakterlánc ($3 \leq n \leq 255$). A `copiaza(s, p, q)` algoritmus az ***s*** karakterláncnak a ***p*** pozíción kezdődő és a ***q*** pozíción végződő tömbszakaszát ($1 \leq p \leq q \leq n$) téríti vissza. A + operátor a karakterláncok konkatenálását (összefűzését) jelöli.

```

Algorithm P(s, n):
    i ← n DIV 3
    j ← 2 * (n DIV 3)
    k ← n
    If n MOD 2 = 0 then
        Return copiaza(s, i, i + 1) + copiaza(s, j - 1, j + 1) + copiaza(s, k, k)
    Else
        Return copiaza(s, 1, 1) + copiaza(s, i, i + 1) + copiaza(s, j - 1, j + 1)
    EndIf
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. A $P(\text{"calculator"}, 10)$ hívás eredményeképpen az algoritmus az "lcular" karakterláncot téríti vissza.
- B. A $P(s, n)$ algoritmus által visszatérített karakterláncnak mindig kevesebb, mint n karaktere van.
- C. A $P(s_1, n_1)$ hívás eredményeképpen akkor és csakis akkor téríthető vissza az "ubbubb" karakterlánc, ha n_1 nagyobb, vagy egyenlő 8-cal, és a $P(s_2, n_2)$ hívás eredményeképpen akkor és csakis akkor téríthető vissza az "ubbfmi" karakterlánc, ha n_2 nagyobb, vagy egyenlő 9-cel.
- D. A $P(s, n)$ hívás eredményeképpen akkor és csakis akkor téríthető vissza különböző karakterekből álló karakterlánc, ha n nagyobb, vagy egyenlő 9-cel és az s karakterlánc különböző karaktereket tartalmaz.

21. Adott az $f(n, x)$ algoritmus, ahol n egy természetes szám ($1 \leq n \leq 10^3$), x egy n elemű, egész számokat tartalmazó vektor ($x[1], x[2], \dots, x[n], -100 \leq x[i] \leq 100, i = 1, 2, \dots, n$), és az $\text{abs}(k)$ algoritmus, amely a k egész szám abszolútértékét téríti vissza.

```

1.  Algorithm f(n, x):
2.      m ← n
3.      a ← 1
4.      b ← -1
5.      For i ← 1, n execute
6.          p ← -1
7.          j ← n
8.          While p = -1 AND j > i execute
9.              If abs(x[i] - x[j]) = 1 then
10.                  p ← j
11.              EndIf
12.              j ← j - 1
13.          EndWhile
14.          t ← i - 1 + n - p
15.          If t < m then
16.              m ← t
17.              a ← i
18.              b ← p
19.          EndIf
20.      EndFor
21.      Return m
22. EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. A 20. sorban levő utasítás végrehajtása után $m = n$ vagy, ha $b \neq -1$, akkor $\text{abs}(x[a] - x[b]) = 1$.
- B. Ha az x vektor a Fibonacci-sorozat első n ($n \geq 3$) elemét tartalmazza, függetlenül a vektorban levő elhelyezésüktől, az algoritmus egy 0-nál nagyobb számot térít vissza.
- C. Egy n elemű x vektor és ugyanazokat az elemeket tartalmazó y vektor esetében, amelynek az elemei az eredeti x vektorból kiindulva k ($1 \leq k < n$) pozícióval körkörösén permutálva vannak jobbra, az $f(n, x)$ és az $f(n, y)$ hívások ugyanazt az értéket térítik vissza, függetlenül k értékétől.
- D. Az $f(9, [8, 2, 4, 5, 2, 5, 3, 4, 6])$ hívás eredményeképpen az algoritmus 6-ot térít vissza.

22. Adott az $f(v, n, i, a, j)$ algoritmus, ahol i, j, n és a természetes számok ($0 \leq j, 1 \leq i, n \leq 20, 1 \leq a \leq 10^4$), v egy n elemű, nemnulla természetes számokat tartalmazó vektor ($v[1], v[2], \dots, v[n], 1 \leq v[k] \leq 10^4, k = 1, 2, \dots, n$).

```

Algorithm f(v, n, i, a, j):
    If a = 0 AND j MOD 2 = 1 then
        Return 1
    EndIf
    If a = 0 then
        Return 0
    EndIf
    If a < 0 OR i = n + 1 then
        Return 0
    EndIf
    If v[i] MOD 2 = 0 then
        Return f(v, n, i + 1, a, j)
    EndIf
    Return f(v, n, i + 1, a - v[i], j + 1) +
           f(v, n, i + 1, a, j)
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Az $f([2, 4, 6], 3, 1, 11, 0)$ hívás eredményeképpen az algoritmus 4-et térít vissza.
- B. Az $f([1, 7, 11, 2, 3], 5, 2, 11, 0)$ hívás eredményeképpen az algoritmus 2-t térít vissza.
- C. Az $f(v, n, 1, 1000, 0)$ hívás eredményeképpen az algoritmus 0-át térít vissza, függetlenül n értékétől és a v vektor értékeitől.
- D. Az $f([1, 2, 3, 4, 5, 6, 3, 7], 8, 1, 7, 0)$ hívás eredményeképpen az algoritmus 2-t térít vissza.

23. Adott az $f(x)$ algoritmus, ahol x egy természetes szám ($0 \leq x \leq 10^9$).

```

Algorithm f(x):
  If x = 0 then
    Return 0
  EndIf
  If x MOD 2 = 0 then
    Return f(x DIV 2) + ((x + 1) MOD 2)
  EndIf
  Return f(x DIV 2) - (x MOD 2)
EndAlgorithm

```

```

Algorithm f2(x):
  If x = 0 then
    Return 0
  EndIf
  v ← f2(x DIV 2)
  If x MOD 2 = 0 then
    Return v + ((x + 1) MOD 2)
  EndIf
  Return v - (x MOD 2)
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Az $f(100)$ és $f(24)$ hívások eredményeképpen az algoritmus ugyanazt az értéket téríti vissza.
- B. Az $f(1024)$ hívás eredményeképpen az algoritmus 10-et térít vissza.
- C. A $[128, 255]$ intervallumban 36 olyan érték létezik, amelyekre az algoritmus 0-át térít vissza.
- D. Bármely x számra, $f2(x) = f(x)$; a két algoritmus időbonyolultsága azonos.

24. Adott az $\text{nrd}(M, r, c, n)$ algoritmus, ahol n ($2 \leq n \leq 10$), r, c ($1 \leq r, c \leq n$) természetes számok, és M egy $n \times n$ elemű négyzetes mátrix ($M[1][1], \dots, M[1][n], \dots, M[n][1], \dots, M[n][n]$, ahol $M[i][j] \in \{0, 1\}$, $i, j = 1, \dots, n$). Egy út az M mátrixban olyan egymásutáni pozíciókból áll, amelyek szomszédosak sorosan vagy oszloposan, mindegyiknek az értéke 1, és amelyben egyetlen pozíció sem ismétlődik. Két út különböző, ha legalább egy pozíciójuk különbözik. A 0 értékek falakat ábrázolnak, amelyeken nem lehet átmenni.

```

Algorithm nrd(M, r, c, n):
  If r = n AND c = n then
    Return 1
  EndIf
  If r < 1 OR r > n OR c < 1 OR c > n OR M[r][c] = 0 then
    Return 0
  EndIf
  total ← 0
  ...
  Return total
EndAlgorithm

```

A következő állítások közül melyek igazak, ha a „...” helyére beírjuk az egyes variánsokban megadott utasításcsoportot?

- A.


```

M[r][c] ← 0
total ← nrd(M, r + 1, c, n) + nrd(M, r - 1, c, n) + nrd(M, r, c + 1, n) + nrd(M, r, c - 1, n)
M[r][c] ← 1

```

 Az algoritmus az $M[r][c]$ pozícióból induló, és az $M[n][n]$ pozícióig tartó különböző utak számát téríti vissza (az eredeti hívás pillanatában $M[r][c] = M[n][n] = 1$).
- B.


```

M[r][c] ← 0
total ← nrd(M, r + 1, c, n) + nrd(M, r - 1, c, n) + nrd(M, r, c + 1, n) + nrd(M, r, c - 1, n)
M[r][c] ← 1

```

 Az algoritmus időbonyolultsága $O(n^4)$.
- C.


```

total ← nrd(M, r + 1, c, n) + nrd(M, r - 1, c, n) + nrd(M, r, c + 1, n) + nrd(M, r, c - 1, n)

```

 Az algoritmus az $M[r][c]$ pozícióból induló, és az $M[n][n]$ pozícióig tartó különböző utak számát téríti vissza (az eredeti hívás pillanatában $M[r][c] = M[n][n] = 1$).
- D.


```

total ← nrd(M, r + 1, c, n) + nrd(M, r, c + 1, n)

```

 Az $\text{nrd}(M, 1, 1, 3)$ hívás eredményeképpen, ahol az M mátrix csak 1-es értékeket tartalmaz, az algoritmus 6-ot térít vissza.

BABEŞ-BOLYAI TUDOMÁNYEGYETEM
MATEMATIKA ÉS INFORMATIKA KAR

Felvételi verseny – 2026 július 17

Informatika írásbeli

JAVÍTÁSI KULCS & MEGOLDÁSOK

HIVATALBÓL: 10 pont

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