

Felvételi verseny – 2024 szeptember 6
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 tömbszakaszát a vektor olyan elemei alkotják, amelyek egymás utáni pozíciókon találhatók.
- Egy vektor részsorozatát a vektor olyan elemei alkotják, amelyek nem kötelezően egymás utáni pozíciókon találhatók az adott vektorban, de az elemek eredeti sorrendje megmarad.
- Ha ugyanabban a sorban több egymásutáni értékadó utasítás található, ezek ";"-vel vannak elválasztva.

1. Legyen a $\text{decide}(n, x)$ algoritmus, ahol n természetes szám ($1 \leq n \leq 10^4$), x egy n elemű egész számokat tároló vektor ($x[1], x[2], \dots, x[n]$, ahol $-100 \leq x[i] \leq 100, i = 1, 2, \dots, n$):

```
Algorithm decide(n, x):  
  b ← True  
  i ← 1  
  While b AND (i < n) execute  
    b ← x[i] < x[i + 1]  
    i ← i + 1  
  EndWhile  
  Return b  
EndAlgorithm
```

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

- A. Ha az x vektor az 1, 2, 3, ..., 10 értékeket tárolja
- B. Ha az x vektor szigorúan növekvő
- C. Ha az x vektornak nincsenek negatív elemei
- D. Ha az x vektor negatív elemei a pozitív elemek előtt találhatók

2. Legyen az $\text{afiseaza}(n, a)$ algoritmus, ahol n természetes szám ($1 \leq n \leq 10^3$), a egy n elemű egész számokat tároló vektor ($a[1], a[2], \dots, a[n]$, ahol $-100 \leq a[i] \leq 100, i = 1, 2, \dots, n$):

```
Algorithm afiseaza(n, a):  
  i ← 1; j ← n  
  While i ≤ j execute  
    If a[i] < a[j] then  
      Write a[i], " "  
      i ← i + 1  
    Else  
      Write a[j], " "  
      j ← j - 1  
    EndIf  
  EndWhile  
EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. Ha az adott vektor növekvően rendezett, a vektor értékei csökkenő sorrendben lesznek kiírva.
- B. Ha az adott vektor csökkenően rendezett, az utolsóknak kiírt elem a maximális elem.
- C. Ha $n = 10$ és $a = [0, 2, 4, 6, 8, 10, 8, 6, 4, 2]$, a vektor értékei növekvő sorrendben lesznek kiírva.
- D. Ha a maximális elem az első pozíción található, a vektor értékei fordított sorrendben lesznek kiírva.

3. Mi a reláció a 8-as számrendszerben felírt $X = 6543_{(8)}$ és a 16-os számrendszerben felírt $Y = \text{CEF}_{(16)}$ szám között?

- A. $X > Y$
- B. $X \leq Y$
- C. $X \geq Y$
- D. $X = Y$

4. Legyen az $f(n)$ algoritmus, ahol n egy nullától különböző természetes szám ($1 \leq n \leq 15$).

```
1. Algorithm f(n):  
2.   x ← 10; y ← 13  
3.   While n ≠ 0 execute  
4.     z ← (x + y) MOD 2  
5.     n ← n DIV 2  
6.     If z MOD 2 = 0 then  
7.       x ← (x * 3 + y * 4) MOD z  
8.       y ← (y + x) * z  
9.     Else  
10.      x ← x + 1  
11.      y ← y - 1  
12.     EndIf  
13.   EndWhile  
14.   Return z  
15. EndAlgorithm
```

A következő állítások közül melyek igazak?

- A. Az algoritmus ugyanazt az értéket téríti vissza bármely $1 \leq n \leq 15$ természetes számra.
- B. Az algoritmus különböző értékeket térít vissza, ha $1 \leq n \leq 10$.
- C. Ha $n = 11$, az algoritmus 0-át térít vissza.
- D. Ha a 10. sorban található utasítást az $x \leftarrow x - 1$ utasítással helyettesítjük, és a 11. sorban található az $y \leftarrow y + 1$ utasítással, az algoritmus ugyanazt az értéket téríti vissza, mint az eredeti változatban, bármely $1 \leq n \leq 15$ természetes számra.

5. Legyen a `numere(n, x)` algoritmus, ahol n természetes szám ($1 \leq n \leq 10^4$), x egy n elemű egész számokat tároló vektor ($x[1], x[2], \dots, x[n]$, ahol $-100 \leq x[i] \leq 100, i = 1, 2, \dots, n$):

```

Algorithm numere(n, x):
  i ← 1; nr ← n
  While i ≤ n execute
    If (x[i] MOD 10) MOD 2 = 0 then
      nr ← nr + 1
    Else
      nr ← nr - 1
    EndIf
    i ← i + 1
  EndWhile
  Return nr = n
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. A `numere(3, [1, 2, 3])` hívás *True*-t térít vissza.
- B. A `numere(3, [1, -2, 3])` hívás *False*-t térít vissza.
- C. A `numere(4, [1, 2, 3, -4])` hívás *False*-t térít vissza.
- D. A `numere(4, [1, 2, 3, 4])` hívás *True*-t térít vissza.

6. Legyen a `ceFace(v, n)` algoritmus, ahol v egy n elemű ($1 \leq n \leq 10^4$), természetes számokat tároló vektor ($v[1], v[2], \dots, v[n]$, ahol $1 \leq v[i] \leq 10^4, i = 1, 2, \dots, n$).

```

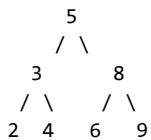
Algorithm ceFace(v, n):
  a ← 0; b ← 1
  For i ← n, 2, -1 execute
    If v[i] = v[i - 1] + 1 then
      b ← b + 1
    Else
      b ← 1
    EndIf
    If b > a then
      a ← b
    EndIf
  EndFor
  Return a
EndAlgorithm

```

Mit térít vissza a `ceFace(v, n)` algoritmus?

- A. A v vektor egymásutáni számokból álló leghosszabb növekvő tömbszakaszának a hosszát
- B. A v vektor egymásutáni számokból álló leghosszabb csökkenő tömbszakaszának a hosszát
- C. A v vektor növekvő tömbszakaszainak a darabszámát
- D. A v vektor egymásutáni számokból álló leghosszabb növekvő részsorozatának a hosszát

7. Legyen a következő bináris fa:



A következő csomópont-sorozatok közül melyik felel meg a fa posztorderben történő bejárásának?

- A. 2, 3, 4, 5, 6, 8, 9
- B. 4, 3, 2, 9, 8, 6, 5
- C. 2, 4, 3, 6, 9, 8, 5
- D. 9, 6, 8, 5, 3, 2, 4

8. Legyen a `prelucrare(n, m, x)` algoritmus, ahol n és m természetes számok ($1 \leq n \leq 100, 1 < m \leq 100$), x egy $n * m$ természetes számot tároló mátrix ($x[1][1], x[1][2], \dots, x[n][m]$, ahol kezdetben $x[i][j] = 0, i = 1, 2, \dots, n; j = 1, 2, \dots, m$):

```

Algorithm creareTablou(n, m, x):
  k ← 1; i ← k
  While i ≤ n execute
    j ← k + 1
    While j ≤ m execute
      If k MOD 2 = 0 then
        x[i][j] ← k * k
      EndIf
      Write x[i][j], " "
      k ← k + 1; j ← j + 1
    EndWhile
    i ← i + 1
  EndWhile
EndAlgorithm

```

Mit ír ki ez az algoritmus?

- A. Egy n értékből álló sorozatot.
- B. Ha m páros szám, egy olyan sorozatot, amelyben a 0 érték páros számok négyzetével váltakozik, az első és utolsó érték pedig 0.
- C. Egy $m - 1$ értékből álló sorozatot.
- D. Egy olyan sorozatot, amelyben a 0 érték páratlan számok négyzetével váltakozik.

9. Adott az n elemű x és az m elemű y vektor, amelyeknek elemei bitek, vagyis 0 vagy 1 értékű számok. n és m nemnulla természetes számok ($0 < n, m \leq 64$). Legyen $b1$ és $b2$ két bit, amelyekre definiáljuk az $op(b1, b2) = \begin{cases} 0 & \text{ha } b1 = b2 \\ 1 & \text{különb} \end{cases}$ műveletet. Definiáljuk az os műveletet, amely az op műveletet alkalmazza az x és y elemeire, de a vektorok végétől indul (először az op műveletet $x[n]$ -re és $y[m]$ -re alkalmazza). Ha a két vektor hossza különbözik, a hosszabb vektor elején található elemek, amelyeknek nincs párjuk a másik vektorban, nem módosulnak. Például, az $[1, 1, 1, 0, 1, 0]$ és $[1, 1, 1, 0]$ vektorok esetében az os művelet eredménye $[1, 1, 0, 1, 0, 0]$. Az algoritmus létrehozza a $maxim(n, m)$ elemű r vektort.

Az OperatieSpeciala(x, n, y, m) algoritmus célja, hogy implementálja a definiált *os* műveletet és térítse vissza az eredményvektort, valamint ennek a hosszát. A Zero(k) algoritmus egy k elemű vektort térít vissza, amelyeknek értéke nulla.

```

1. Algorithm OperatieSpeciala(x, n, y, m):
2.   length ← n
3.   lenF ← m
4.   r ← Zero(m)
5.   If m < n then
6.     length ← m
7.     lenF ← n
8.     r ← Zero(n)
9.   EndIf
10.  For i ← 1, length execute
11.    If (x[i] + y[i]) MOD 2 = 0 then
12.      r[i] ← 0
13.    Else
14.      r[i] ← 1
15.    EndIf
16.  EndFor
17.  For i ← length + 1, m execute
18.    r[i] ← y[i]
19.  EndFor
20.  For i ← length + 1, n execute
21.    r[i] ← x[i]
22.  EndFor
23.  Return r, lenF
24. EndAlgorithm

```

Melyek igazak az alábbi állítások közül?

- A. Az OperatieSpeciala(x, n, y, m) algoritmus helyesen implementálja az *os* műveletet és visszatéríti az eredményvektort, valamint ennek a hosszát.
- B. Azoknak a bemeneti adatoknak esetében, amelyekre a 18. és 21. sorban levő utasítások azonos számszor hajtódnak végre, a visszatérített eredmény helyes.
- C. Az implementáció helyessé válna, ha helyettesítjük a 11. sorban levő utasítást az alábbival:
If (x[n - i] + y[m - i]) MOD 2 = 0 then
- D. Az OperatieSpeciala(x, n, y, m) algoritmus által visszatérített eredmény nem helyes, és a visszatérített vektor az elemeket fordított sorrendben tartalmazza.

10. Legyen a ceFace(x, m, y, n) algoritmus, ahol x egy m elemű karakterlánc ($1 \leq m \leq 100$), y egy n elemű karakterlánc ($1 \leq n \leq 100$), és $m < n$.

```

1. Algorithm ceFace(x, m, y, n):
2.   i ← 1
3.   ok ← True
4.   While ok AND i ≤ m execute
5.     If i ≤ m AND x[i] ≠ y[i] then
6.       ok ← False
7.     Else
8.       i ← i + 1
9.     EndIf
10.  EndWhile
11.  _____
12. EndAlgorithm

```

Mely utasítást kellene tartalmaznia a 11. sornak, ahhoz, hogy az algoritmus *True*-t térítsen vissza, ha az x karakterlánc előszelete az y karakterláncnak? Például, ha $x = "abc"$ és $y = "abcd"$, x előszelete y -nak és az algoritmus *True*-t térít vissza.

- A. Return (i = m) OR (i = n)
- B. Return i = m
- C. Return i > m
- D. Return ok

11. Legyen az m soros és n oszlopos A mátrix ($A[1][1], A[1][2], \dots, A[m][n]$), ahol m és n természetes számok ($1 < m \leq 25$, $1 < n \leq 25$), és $1 \leq A[i][j] \leq 10^3$, $i = 1, 2, \dots, m; j = 1, 2, \dots, n$.

A következő algoritmusok közül melyek térítik vissza a mátrix k -dik oszlopa elemeinek összegét k ($1 < k \leq n$)?

A. Algorithm suma(A, m, n, k):
s ← 0
For i ← n, 1, -1 execute
s ← s + A[i][k]
EndFor
Return s
EndAlgorithm

B. Algorithm suma(A, m, n, k):
s ← 0; i ← 1
While i ≤ m execute
s ← s + A[i][k]
i ← i + 1
EndWhile
Return s
EndAlgorithm

C. Algorithm suma(A, m, n, k):
s ← 0
For i ← 1, m execute
s ← s + A[k][i]
EndFor
Return s
EndAlgorithm

D. Algorithm suma(A, m, n, k):
s ← 0; k ← 1
While k ≤ n execute
s ← s + A[k][k]
k ← k + 1
EndWhile
Return s
EndAlgorithm

12. Legyen az $F(n)$ algoritmus, ahol n nullától különböző természetes szám ($1 \leq n \leq 10^6$). Az $\text{sqrt}(n)$ algoritmus az n szám négyzetgyökét téríti vissza és bonyolultsága $O(1)$. Az $[a]$ jelölés az a érték egész részét jelenti. A „/” művelettel a valós osztást jelöljük, például: $3 / 2 = 1.5$.

```

Algorithm F(n):
  If n = 1 then
    Return 1
  EndIf
  i ← [n / sqrt(n)]
  Return 1 + F(i)
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Az $F(n)$ algoritmus időbonyolultsága $O(\log \log n)$.
- B. Az $F(200)$ hívás eredményeként 4-et kapunk.
- C. Az $F(250)$ hívás eredményeként 5-öt kapunk.
- D. Az $F(n)$ algoritmus időbonyolultsága $O(1)$.

13. Legyen a $\text{check}(n, x)$ algoritmus, ahol n természetes szám ($1 \leq n \leq 10^4$), 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 check(n, x):
  If n < 3 then
    Return False
  EndIf
  p ← select(n, x)
  If p = 1 OR p = n then
    Return False
  EndIf
  For i ← 2, p execute
    If x[i] ≥ x[i - 1] then
      Return False
    EndIf
  EndFor
  For i ← p + 1, n - 1 execute
    If x[i] ≥ x[i + 1] then
      Return False
    EndIf
  EndFor
  Return True
EndAlgorithm

```

```

Algorithm select(n, x):
  r ← 0
  v ← x[1]
  For i ← 2, n execute
    If x[i] < v then
      r ← i
      v ← x[i]
    EndIf
  EndFor
  Return r
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Ha az x vektor növekvően rendezett és van legalább 3 eleme, a $\text{check}(n, x)$ algoritmus *True*-t térít vissza.
- B. Ha $x = [12, 10, 8, 5, 9, 11, 15, 18]$ és $n = 8$, a $\text{check}(n, x)$ algoritmus *True*-t térít vissza.
- C. Ha $x = [20, 10, 5, 1, 2, 4, 6, 10, 8]$ és $n = 9$, a $\text{check}(n, x)$ algoritmus *False*-t térít vissza.
- D. Ha az x vektor szigorúan növekvően rendezett és van legalább 3 eleme, a $\text{check}(n, x)$ algoritmus *True*-t térít vissza.

14. Legyen az $f(x, n)$ algoritmus, ahol n természetes szám ($3 \leq n \leq 10^4$), x egy n elemű természetes számokat tároló vektor ($x[1], x[2], \dots, x[n], 1 \leq x[i] \leq 10^4, i = 1, 2, \dots, n$). A $[\]$ jelölés az üres vektort jelenti, az $[a, b]$ az a -t és b -t tároló 2 elemű vektort.

```

Algorithm f(x, n):
  If n < 2 then
    Return []
  EndIf
  If n = 2 then
    If x[1] > x[2] then
      Return [x[1], x[2]]
    Else
      Return [x[2], x[1]]
    EndIf
  EndIf
  y ← f(x, n - 1)
  If x[n] > y[1] then
    Return [x[n], y[1]]
  Else
    If x[n] > y[2] then
      Return [y[1], x[n]]
    Else
      Return y
    EndIf
  EndIf
EndAlgorithm

```

Mit fog visszatéríteni az algoritmus ha $f([4, 15, 5, 8, 10, 18, 16, 19, 1, 12], 10)$ alakban hívjuk meg?

- A. [19, 18]
- B. [18, 19]
- C. [16, 19]
- D. [19, 16]

15. Legyen a $\text{numere}(x, n, e)$ algoritmus, ahol n természetes szám ($1 \leq n \leq 10^4$), x egy n elemű egész számokat tároló vektor ($x[1], x[2], \dots, x[n]$, ahol $-100 \leq x[i] \leq 100, i = 1, 2, \dots, n$) és e a vektor egy elemének értéke:

```

Algorithm numere(x, n, e):
  i ← 1
  c ← 0
  b ← True
  If n MOD 2 = 0 then
    Return False
  EndIf
  While (i ≤ n) AND b execute
    If x[i] < e then
      c ← c + 1
    Else
      b ← False
    EndIf
    i ← i + 1
  EndWhile
  Return c = (n - i + 1)
EndAlgorithm

```

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

- A. Ha a vektornak páros darabszámú eleme van és csökkenően rendezett az e értékű elemmel bezárólag, amely az $n \text{ DIV } 2$ pozícióban található.
- B. Ha vektornak páratlan darabszámú eleme van és szigorúan növekvően rendezett az e értékű elemmel bezárólag, amely az $n \text{ DIV } 2 + 1$ pozícióban található.
- C. Ha vektornak páratlan darabszámú eleme van és csökkenően rendezett az e értékű elemmel bezárólag, amely az $n \text{ DIV } 2 + 1$ pozícióban található.
- D. Ha vektornak páratlan darabszámú eleme van, az e értékű elem az $n \text{ DIV } 2 + 1$ pozícióban található és e előtt csak kisebb értékű elemek, e után pedig csak nagyobb értékűek találhatók.

16. A következő algoritmusok közül melyek írják ki az a számot a b számrendszerben, ahol a és b a 10-es számrendszerben megadott természetes számok ($1 \leq a \leq 10^4, 2 \leq b \leq 9, a > b$)?

A.

```

Algorithm afiseaza(a, b):
  If a ≠ 0 then
    Write a MOD b
    afiseaza(a DIV b, b)
  EndIf
EndAlgorithm

```

B.

```

Algorithm afiseaza(a, b):
  If a ≠ 0 then
    afiseaza(a DIV b, b)
    Write a MOD b
  EndIf
EndAlgorithm

```

C.

```

Algorithm afiseaza(a, b):
  While a > 0 execute
    Write a MOD b
    a ← a DIV b
  EndWhile
EndAlgorithm

```

D.

```

Algorithm afiseaza(a, b):
  nrNou ← 0
  putere ← 1
  While a > 0 execute
    nrNou ← nrNou + (a MOD b) * putere
    a ← a DIV b
    putere ← putere * b
  EndWhile
  Write nrNou
EndAlgorithm

```

17. Legyen az $f(x, y)$ algoritmus, ahol x és y természetes számok ($1 \leq x \leq 100, 1 \leq y \leq 100$).

```

Algorithm f(x, y):
  If x = y then
    Write "start: "
  Else
    If x MOD y = 0 then
      f(x + 1, y + 2)
    Else
      If (x DIV y) MOD 2 = 0 then
        f(x + 2, y + 1)
        Write "*"
      Else
        f(x - 1, y + 1)
        Write "#"
      EndIf
    EndIf
  EndIf
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Az $f(12, 15)$ és $f(8, 12)$ hívások eredményeként a kiírt karakterláncok nem azonosak
- B. Az $f(15, 12)$ és $f(12, 8)$ hívások eredményeként a kiírt karakterláncok azonosak
- C. Az $f(17, 23)$ hívás eredményeként az algoritmus nem ír ki egyetlen '#' karaktert sem
- D. Az $f(23, 17)$ hívás eredményeként a kiírt karakterláncban legalább egy '#' karakter található

18. Legyen a $\text{decide}(n, x, t)$ algoritmus, ahol x egy n elemű természetes számokat tároló vektor ($2 \leq n \leq 10^4$, $1 \leq x[i] \leq 10^4$, $i = 1, 2, \dots, n$) és t egy természetes szám ($1 \leq t \leq 10^4$).

```

Algorithm decide(n, x, t):
  left ← 1; right ← n
  While x[left] + x[right] ≠ t execute
    If x[left] + x[right] < t then
      left ← left + 1
    Else
      right ← right - 1
    EndIf
  EndWhile
  Return left, right
EndAlgorithm

```

Mely esetekben határozza meg a $\text{decide}(n, x, t)$ algoritmus azokat a $left$ és $right$ indexeket ($1 \leq left < right \leq n$) amelyekre érvényes, hogy $x[left] + x[right] = t$?

- A. Akkor és csakis akkor, ha az x vektor különböző számokat tárol.
- B. Akkor és csakis akkor, ha az x vektor növekvően rendezett.
- C. Ha az x vektor csökkenően rendezett és különböző számokat tárol.
- D. Ha az x vektor növekvően rendezett és a vektorban létezik legalább egy elempár, amelyeknek összege t .

19. Legyen a $\text{perechi}(x, y)$ algoritmus, ahol x és y nullától különböző természetes szám ($1 \leq x, y \leq 100$):

```

Algorithm perechi(x, y):
  nr ← 0; d ← 2
  While d ≤ x AND d ≤ y execute
    If (x MOD d = 0) AND (y MOD d = 0) then
      nr ← nr + 1
      x ← x DIV d
      y ← y DIV d
    Else
      d ← d + 1
    EndIf
  EndWhile
  Write nr, " ", x, " ", y
EndAlgorithm

```

A következő válaszok között melyikben található csak olyan (x, y) számpárok, amelyekre a $\text{perechi}(x, y)$ algoritmus az 1 7 11 számokat írja ki?

- A. (14, 22), (21, 33), (35, 55), (49, 77)
- B. (7, 11), (14, 22), (21, 33), (28, 44)
- C. (1, 7), (1, 11)
- D. (2, 2), (3, 3), (4, 4), (5, 5)

20. Legyen a $\text{first}(x, n)$ algoritmus, ahol n egy nullától különböző természetes szám ($2 \leq n \leq 10^4$), x egy n elemű természetes számokat tároló vektor ($x[1], x[2], \dots, x[n]$, ahol $1 \leq x[i] \leq 10^4$, $i = 1, 2, \dots, n$).

```

01. Algorithm first(x, n):
02.   f1 ← False
03.   f2 ← False
04.   For i ← 1, n execute
05.     If x[i] = 1 then
06.       f1 ← True
07.     EndIf
08.     If x[i] = n then
09.       f2 ← True
10.     EndIf
11.     If x[i] ≥ n then
12.       x[i] ← 1
13.     EndIf
14.   EndFor
15.   If NOT f1 then
16.     Return 1
17.   EndIf
18.   For i ← 1, n execute
19.     ...
20.   EndFor
21.   If f2 then
22.     x[n] ← n
23.   EndIf
24.   For i ← 1, n execute
25.     ...
26.     Return i
27.   EndIf
28. EndFor
29. Return n + 1
30. EndAlgorithm

```

Mivel kellene helyettesíteni a 19-edik és 25-dik sort, ahhoz, hogy az algoritmus azt a legkisebb, nullától különböző természetes számot térítse vissza, amely nem található meg az x vektorban?

- A.
 - 19: $x[x[i] \text{ MOD } (n + 1)] \leftarrow x[x[i] \text{ MOD } (n + 1)] + n$
 - 25: **If** $x[i] \text{ DIV } (n + 1) = 0$ **then**
- B.
 - 19: $x[x[i] \text{ MOD } n] \leftarrow x[x[i] \text{ MOD } n] + n$
 - 25: **If** $x[i] \text{ DIV } n = 0$ **then**
- C.
 - 19: $x[x[i] \text{ MOD } n] \leftarrow 1$
 - 25: **If** $x[i] = 1$ **then**
- D.
 - 19: $x[x[i] \text{ MOD } n] \leftarrow x[x[i] \text{ MOD } n] + n$
 - 25: **If** $x[i] \text{ MOD } n = 0$ **then**

21. Legyen a $\text{ceFace}(\text{arr}, n)$ algoritmus, ahol arr egy n ($1 \leq n \leq 100$) elemű egész számokat tároló vektor ($\text{arr}[1], \text{arr}[2], \dots, \text{arr}[n]$), ahol $-10^5 \leq \text{arr}[i] \leq 10^5$ $i = 1, 2, \dots, n$).

```

Algorithm ceFace(arr, n):
    sum ← 0
    For i ← 1, n execute
        sum ← sum + arr[i]
    EndFor
    If sum MOD 2 ≠ 0 then
        Return False
    EndIf
    Return auxiliar(arr, n, sum DIV 2)
EndAlgorithm

```

```

Algorithm auxiliar(arr, n, sum):
    If sum = 0 then
        Return True
    EndIf
    If n = 1 AND sum ≠ 0 then
        Return False
    EndIf
    If arr[n - 1] > sum then
        Return auxiliar(arr, n - 1, sum)
    EndIf
    Return auxiliar(arr, n - 1, sum) OR
        auxiliar(arr, n - 1, sum - arr[n - 1])
EndAlgorithm

```

A következő állítások közül melyek helyesek?

- A. A $\text{ceFace}([11, 5, 6, 22, 0, 7, 6, 13], 8)$ hívás *True*-t térít vissza.
- B. A $\text{ceFace}([-5, -6, -22, -7, -6, -13], 6)$ hívás **NEM** térít vissza *True*-t.
- C. Ha az arr vektor csak negatív értékeket tárol, az $\text{auxiliar}(\text{arr}, n, \text{sum})$ algoritmus végtelen ciklusba kerül.
- D. Akkor és csakis akkor, ha az arr vektor elemei szétoszthatók két olyan halmazba, amelyek elemeinek számtani középárányosa egyenlő, a $\text{ceFace}(\text{arr}, n)$ algoritmus *True*-t térít vissza.

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

```

1. Algorithm f(n, x):
2.     s1 ← h(n, x)
3.     For i ← 1, 2 * n execute
4.         x[((i + 1) MOD n) + 1] ← g(x[(i MOD n) + 1], x[((i + 1) MOD n) + 1])
5.     EndFor
6.     s2 ← h(n, x)
7.     Return x[n]
8. EndAlgorithm

```

```

Algorithm g(a, b):
    If a * b = 0 then
        Return a + b
    EndIf
    If a = b then
        Return a
    EndIf
    If a > b then
        Return g(a - b, b)
    EndIf
    Return g(a, b - a)
EndAlgorithm

```

```

Algorithm h(n, x):
    s ← 0
    For i ← 1, n execute
        s ← s + x[i]
    EndFor
    Return s
EndAlgorithm

```

A következő állítások közül melyek igazak?

- A. Az $f(6, [12, 16, 80, 40, 28, 144])$ hívás eredményeként az algoritmus 4-et térít vissza.
- B. Bármely bemeneti vektor esetében az $s1$ érték (amelyet az $f(n, x)$ algoritmus 2-dik sorában számítunk ki) szigorúan nagyobb lesz, mint $s2$ (amelyet az $f(n, x)$ algoritmus 6-dik sorában számítunk ki).
- C. Ha az $f(n, x)$ algoritmusban a 3-dik, 4-dik és 5-dik sorban található utasításokat az alábbi algoritmrészlettel helyettesítjük, az $f(n, x)$ algoritmus végrehajtása után az x vektor tartalma ugyanaz lesz, mint az eredeti algoritmus esetében.


```

For j ← 1, 2 execute
    For i ← 1, n - 1 execute
        x[i + 1] ← g(x[i], x[i + 1])
    EndFor
EndFor

```
- D. Létezik olyan bemeneti n elemű vektor, amelynek esetében az $f(n, x)$ algoritmus időbonyolultsága $O(n)$.

23. Legyen a nullától különböző n páros természetes szám ($2 \leq n \leq 12$). Szeretnénk generálni mindazokat az n kerek zárójelet tartalmazó x karakterláncokat, amelyekben a zárójelek helyesen nyitódnak és csukódnak. A $\text{paranteze}(i, \text{desc}, \text{inc}, x, n)$ algoritmust $\text{paranteze}(2, 1, 0, x, n)$ alakban hívjuk meg, miután megtörténtek az $x[1] \leftarrow '('$ és $x[n] \leftarrow ')'$ kezdőértékdadások. Az $\text{afisare}(n, x)$ algoritmus kiírja az n elemű x karakterláncot.

```

1. Algorithm paranteze(i, desc, inc, x, n):
2.   If i = n then
3.     afisare(n, x)
4.   Else
5.     If ____ then
6.       x[i] = '('
7.       paranteze(i + 1, desc + 1, inc, x, n)
8.     EndIf
9.     If ____ then
10.      x[i] = ')'
11.      paranteze(i + 1, desc, inc + 1, x, n)
12.    EndIf
13.  EndIf
14. EndAlgorithm

```

Mivel kell helyettesítenünk az alább megadott sorokat, ahhoz, hogy az algoritmus csak a specifikációnak megfelelő helyes karakterláncokat írja ki?

- A. Az 5-dik sort a $desc < n$ relációval, a 9-dik sort az $inc < desc$ relációval
- B. Az 5-dik sort a $desc < n \text{ DIV } 2$ relációval, a 9-dik sort az $inc < desc$ relációval
- C. Az 5-dik sort a $desc < n$ relációval, a 9-dik sort az $inc < n \text{ DIV } 2$ relációval
- D. Függetlenül attól, hogy mivel egészítjük ki az 5-dik és 9-dik sort, az algoritmus nem fogja kiírni a specifikációnak megfelelő összes karakterláncot.

24. A testnevelés órán n gyerek áll egymás mellett, arccal a tanár felé, amikor ez kéri, hogy forduljanak balra. Egyesek balra fordulnak, mások – tévedésből – jobbra. Egy időegység alatt minden gyerek, aki szemben találja magát a mellette állóval, sarkon fordul (elfordul 180° -kal). Egy időegység alatt minden gyerek legtöbb egy fordulatot végez. A gyerekek mozgása folytatódik, amíg nincs több gyerek, akik szemtől szembe állnának. Állapítsátok meg, hogy az $intoarceri(n, c)$ algoritmusok közül melyek határozzák meg azoknak az időegységeknek a t darabszámát, amelyek eltelnek míg nincs több gyerek, akik szemtől szemben állnának. Az n változó nemnulla természetes szám ($1 \leq n \leq 100$), a c karakterláncnak n eleme van, ahol $c[i]$ vagy 's' (jelentése „bal”) vagy 'd' (jelentése „jobb”) annak megfelelően, hogy merre fordult az i -dik gyerek a tanár kérése után. *Példák:* ha $n = 6$ és $c = "sdsssd"$ akkor $t = 3$; ha $n = 3$ és $c = "sdd"$ akkor t értéke 0 lesz. A $copiaza(a, n)$ algoritmus az n elemű a vektor másolatát téríti vissza.

A.

```

Algorithm intoarceri(n, c):
  t ← 0; aux ← copiaza(c, n); ok ← False
  While NOT ok execute
    ok ← True
    For i ← 1, n - 1 execute
      If (aux[i] = 'd') AND (aux[i + 1] = 's') then
        c[i] ← 's'; c[i + 1] ← 'd'
        ok ← False
      EndIf
    EndFor
    aux ← copiaza(c, n)
    If NOT ok then
      t ← t + 1
    EndIf
  EndWhile
  Return t
EndAlgorithm

```

C.

```

Algorithm intoarceri(n, c):
  stop ← False; t ← 0
  While NOT stop execute
    i ← 1
    stop ← True
    While i < n execute
      If (c[i] = 'd') AND (c[i + 1] = 's') then
        c[i] ← 's'; c[i + 1] ← 'd'
        i ← i + 2
        stop ← False
      Else
        i ← i + 1
      EndIf
    Endwhile
    If NOT stop then
      t ← t + 1
    EndIf
  Endwhile
  Return t
EndAlgorithm

```

B.

```

Algorithm intoarceri(n, c):
  t ← 0; dr ← 0; st ← 0
  For i ← 1, n execute
    If c[i] = 'd' then
      dr ← dr + 1
      If st > 0 then
        st ← st - 1
      EndIf
    Else
      If dr > 0 then
        t ← dr + st
        st ← st + 1
      EndIf
    EndIf
  EndFor
  Return t
EndAlgorithm

```

D.

```

Algorithm intoarceri(n, c):
  t ← 0; dr ← 0; st ← 0
  For i ← 1, n execute
    If c[i] = 'd' then
      dr ← dr + 1
      If st > 0 then
        t ← dr + st
        st ← st - 1
      EndIf
    Else
      If dr > 0 then
        t ← dr + st
        dr ← dr + 1
      EndIf
    EndIf
  EndFor
  Return t
EndAlgorithm

```


Felvételi verseny – 2024 szeptember 6.

Informatika írásbeli

JAVÍTÁSI KULCS & MEGOLDÁSOK

HIVATALBÓL: 10 pont

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