

Concurs de admitere – 6 septembrie 2024
Proba scrisă la Informatică

NOTĂ IMPORTANTĂ:

În lipsa altor precizări:

- Toate operațiile aritmetice se efectuează pe tipuri de date nelimitate (nu există *overflow / underflow*).
- Numerotarea indicilor tuturor vectorilor, matricelor și a șirurilor de caractere începe de la 1.
- Toate restricțiile se referă la valorile parametrilor actuali la momentul apelului inițial.
- O subsecvență a unui vector este formată din elemente care ocupă poziții consecutive în vector.
- Un subșir al unui vector/șir este format din elemente situate nu neapărat pe poziții consecutive în vectorul/șirul respectiv, în ordinea în care acestea apar în vectorul/șirul dat.
- Dacă pe un același rând apar mai multe instrucțiuni de atribuire consecutive, acestea sunt delimitate prin ";".

1. Se consideră algoritmul $decide(n, x)$, unde n este număr natural ($1 \leq n \leq 10^4$), iar x este un vector cu n elemente numere întregi ($x[1], x[2], \dots, x[n]$, unde $-100 \leq x[i] \leq 100$, pentru $i = 1, 2, \dots, n$):

```
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
```

Pentru care din următoarele situații algoritmul returnează *True*?

- A. Dacă vectorul x este format din valorile 1, 2, 3, ..., 10
- B. Dacă vectorul x este strict crescător
- C. Dacă vectorul x nu are elemente negative
- D. Dacă vectorul x are elementele negative situate înaintea celor pozitive

2. Se consideră algoritmul $afiseaza(n, a)$, unde n este număr natural ($1 \leq n \leq 10^3$), iar a este un vector cu n elemente numere întregi ($a[1], a[2], \dots, a[n]$, unde $-100 \leq a[i] \leq 100$, pentru $i = 1, 2, \dots, n$):

```
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
```

Care dintre următoarele afirmații sunt adevărate?

- A. Dacă vectorul dat este sortat crescător, valorile din vector se afișează în ordine descrescătoare.
- B. Dacă vectorul dat este sortat descrescător, ultimul element afișat este elementul maxim.
- C. Dacă $n = 10$ și $a = [0, 2, 4, 6, 8, 10, 8, 6, 4, 2]$, valorile din vector se afișează în ordine crescătoare.
- D. Dacă elementul maxim este pe prima poziție, valorile din vector se afișează în ordine inversă.

3. Care este relația dintre numerele $X = 6543_{(8)}$ în baza 8 și $Y = CEF_{(16)}$ în baza 16?

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

4. Se consideră algoritmul $f(n)$, unde n este număr natural nenul ($1 \leq n \leq 15$).

```
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
```

Care dintre următoarele afirmații sunt adevărate?

- A. Algoritmul returnează aceeași valoare pentru orice număr natural $1 \leq n \leq 15$.
- B. Algoritmul returnează valori distincte pentru numerele naturale n cu proprietatea $1 \leq n \leq 10$.
- C. Dacă $n = 11$, algoritmul returnează 0.
- D. Dacă schimbăm instrucțiunea de pe linia 10 cu $x \leftarrow x - 1$, și cea de pe linia 11 cu $y \leftarrow y + 1$ algoritmul returnează aceeași valoare ca în varianta originală pentru orice număr natural $1 \leq n \leq 15$.

5. Se consideră algoritmul $\text{numere}(n, x)$, unde n este număr natural ($1 \leq n \leq 10^4$), iar x este un vector cu n elemente numere întregi ($x[1], x[2], \dots, x[n]$, unde $-100 \leq x[i] \leq 100$, pentru $i = 1, 2, \dots, n$):

```

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

```

Care dintre următoarele afirmații sunt adevărate?

- A. Apelul $\text{numere}(3, [1, 2, 3])$ returnează *True*.
- B. Apelul $\text{numere}(3, [1, -2, 3])$ returnează *False*.
- C. Apelul $\text{numere}(4, [1, 2, 3, -4])$ returnează *False*.
- D. Apelul $\text{numere}(4, [1, 2, 3, 4])$ returnează *True*.

6. Se consideră algoritmul $\text{ceFace}(v, n)$, unde v este un vector de n ($1 \leq n \leq 10^4$) numere naturale ($v[1], v[2], \dots, v[n]$, unde $1 \leq v[i] \leq 10^4$ pentru $i = 1, 2, \dots, n$).

```

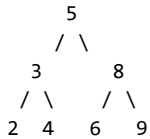
Algorithm ceFace(v, n):
  a ← 0; b ← 1
  For i ← n, 2, -1 execute
    If v[i] = v[i - 1] + 1 then
      b ← b + 1
    Else
      b ← 1
    EndIf
    If b > a then
      a ← b
    EndIf
  EndFor
  Return a
EndAlgorithm

```

Ce returnează algoritmul $\text{ceFace}(v, n)$?

- A. Lungimea celei mai lungi subsecvențe formată din numere consecutive crescătoare din vectorul v .
- B. Lungimea celei mai lungi subsecvențe formată din numere consecutive descrescătoare din vectorul v .
- C. Numărul subsecvențelor crescătoare din vectorul v .
- D. Lungimea celui mai lung subșir format din numere consecutive crescătoare din vectorul v .

7. Se consideră următorul arbore binar:



Care dintre următoarele șiruri de noduri corespund traversării arborelui în postordine?

- 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. Se consideră algoritmul $\text{prelucrare}(n, m, x)$, unde n și m sunt numere naturale ($1 \leq n \leq 100, 1 < m \leq 100$), iar x este o matrice cu $n * m$ elemente numere naturale ($x[1][1], x[1][2], \dots, x[n][m]$, unde inițial $x[i][j] = 0$, pentru $i = 1, 2, \dots, n; j = 1, 2, \dots, m$):

```

Algorithm prelucrare(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

```

Ce afișează acest algoritmul?

- A. Un șir de n valori.
- B. Dacă m este par, un șir de valori în care valoarea 0 alternează cu valori care reprezintă pătrate perfecte pare, iar prima și ultima valoare sunt 0.
- C. Un șir de $m - 1$ valori.
- D. Un șir de valori în care valoarea 0 alternează cu valori care reprezintă pătrate perfecte impare.

9. Se consideră doi vectori de biți, x cu n , și y cu m elemente, unde n și m sunt numere naturale nenule ($0 < n, m \leq 64$).

Elementele vectorilor sunt 0 sau 1. Fie $b1$ și $b2$ doi biți pentru care definim operația $op(b1, b2) = \begin{cases} 0 & \text{dacă } b1 = b2 \\ 1 & \text{altfel} \end{cases}$

Definim operația os ca aplicare a operației op pe elementele din x și y , dar pornind de la finalul vectorilor (deci prima dată aplicăm op pe $x[n]$ și $y[m]$). Dacă cei doi vectori au număr diferit de elemente, elementele de la începutul vectorului cu lungime mai mare, care nu au pereche în celălalt vector, rămân nemodificate. De exemplu, pentru vectorii $[1, 1, 1, 0, 1, 0]$ și $[1, 1, 1, 0]$, rezultatul operației os va fi $[1, 1, 0, 1, 0, 0]$. Algoritmul creează un vector r cu $\max(n, m)$ elemente.

Scopul algoritmului OperatieSpeciala(x, n, y, m) este să implementeze operația *os*, descrisă mai sus, și să returneze vectorul rezultat și lungimea lui. Algoritmul Zero(k) returnează un vector cu *k* elemente, toate egale cu zero.

```

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

```

Care dintre afirmațiile de mai jos sunt adevărate?

- A. Algoritmul OperatieSpeciala(x, n, y, m) implementează corect operația *os* și returnează vectorul rezultat și lungimea lui.
- B. Pentru acele date de intrare pentru care instrucțiunile de pe rândul 18 și 21 se execută de același număr de ori, rezultatul returnat este corect.
- C. Implementarea ar deveni corectă, dacă înlocuim instrucțiunea de pe rândul 11 cu `If (x[n - i] + y[m - i]) MOD 2 = 0 then`
- D. Rezultatul algoritmului OperatieSpeciala(x, n, y, m) nu este corect, iar vectorul returnat conține elementele în ordine inversă.

10. Se consideră algoritmul ceFace(x, m, y, n), unde *x* este un șir de caractere de lungime *m* ($1 \leq m \leq 100$) iar *y* un șir de caractere de lungime *n* ($1 \leq n \leq 100$), astfel încât $m < n$.

```

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

```

Ce instrucțiune ar trebui să conțină linia 11, astfel încât algoritmul să returneze *True* în cazul în care șirul *x* este prefix al șirului *y*? *Exemplu:* dacă *x* = "abc" și *y* = "abcd", *x* este prefix al lui *y* și algoritmul returnează *True*.

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

11. Se consideră o matrice *A* cu *m* linii și *n* coloane ($A[1][1], A[1][2], \dots, A[m][n]$), unde *m* și *n* sunt numere naturale ($1 < m \leq 25, 1 < n \leq 25$), și $1 \leq A[i][j] \leq 10^3$, pentru $i = 1, 2, \dots, m; j = 1, 2, \dots, n$.

Care dintre următorii algoritmi returnează suma elementelor de pe coloana *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. Se consideră algoritmul $F(n)$, unde n este număr natural nenul ($1 \leq n \leq 10^6$). Algoritmul $\text{sqrt}(n)$ returnează radicalul lui n și are complexitatea $O(1)$. Notăția $[a]$ reprezintă partea întreagă a valorii lui a . Operatorul „/” reprezintă împărțirea reală, de exemplu: $3 / 2 = 1.5$.

```

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

```

Care dintre următoarele afirmații sunt adevărate?

- A. Algoritmul $F(n)$ are complexitatea timp $O(\log \log n)$.
- B. În urma apelului $F(200)$ se obține valoarea 4.
- C. În urma apelului $F(250)$ se obține valoarea 5.
- D. Algoritmul $F(n)$ are complexitatea timp $O(1)$.

13. Se consideră algoritmul $\text{check}(n, x)$, unde n este număr natural ($1 \leq n \leq 10^4$), iar x este un vector cu n elemente numere întregi ($x[1], x[2], \dots, x[n], -100 \leq x[i] \leq 100$, pentru $i = 1, 2, \dots, n$):

```

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

```

Care dintre următoarele afirmații sunt adevărate?

- A. Dacă vectorul x este ordonat descrescător și are cel puțin 3 elemente, algoritmul $\text{check}(n, x)$ returnează *True*.
- B. Dacă vectorul $x = [12, 10, 8, 5, 9, 11, 15, 18]$ și $n = 8$ algoritmul $\text{check}(n, x)$ returnează *True*.
- C. Dacă vectorul $x = [20, 10, 5, 1, 2, 4, 6, 10, 8]$ și $n = 9$ algoritmul $\text{check}(n, x)$ returnează *False*.
- D. Dacă vectorul x este ordonat strict crescător și are cel puțin 3 elemente, algoritmul $\text{check}(n, x)$ returnează *True*.

14. Se consideră algoritmul $f(x, n)$, unde n este număr natural ($3 \leq n \leq 10^4$), iar x este un vector de n numere naturale ($x[1], x[2], \dots, x[n], 1 \leq x[i] \leq 10^4$, pentru $i = 1, 2, \dots, n$). Prin $[\]$ s-a notat un vector vid, iar prin $[a, b]$ s-a notat un vector cu 2 elemente a și b .

```

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

```

Ce va returna algoritmul pentru apelul $f([4, 15, 5, 8, 10, 18, 16, 19, 1, 12], 10)$?

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

15. Se consideră algoritmul $\text{numere}(x, n, e)$, unde n este număr natural ($1 \leq n \leq 10^4$), x este un vector cu n elemente numere întregi ($x[1], x[2], \dots, x[n]$, unde $-100 \leq x[i] \leq 100$, pentru $i = 1, 2, \dots, n$), iar e este valoarea unui element din vector:

```

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

```

În care din următoarele situații returnează algoritmul *True*?

- A. Dacă vectorul are număr par de elemente și este ordonat descrescător până la elementul cu valoarea e inclusiv, care se află pe poziția $n \text{ DIV } 2$.
- B. Dacă vectorul are număr impar de elemente și este ordonat strict crescător până la elementul cu valoarea e inclusiv, care se află pe poziția $n \text{ DIV } 2 + 1$.
- C. Dacă vectorul are număr impar de elemente și este ordonat descrescător până la elementul cu valoarea e inclusiv, care se află pe poziția $n \text{ DIV } 2 + 1$.
- D. Dacă vectorul are număr impar de elemente, iar valoarea e se găsește pe poziția $n \text{ DIV } 2 + 1$ și înainte de e sunt doar valori mai mici, iar după e sunt doar valori mai mari.

16. Care dintre algoritmi următori afișează reprezentarea numărului a în baza b , unde a, b sunt numere naturale date în baza 10 ($1 \leq a \leq 10^4, 2 \leq b \leq 9, a > b$)?

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. Se consideră algoritmul $f(x, y)$, unde x și y sunt două numere naturale ($1 \leq x \leq 100, 1 \leq y \leq 100$).

```

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

```

Care dintre următoarele afirmații sunt adevărate?

- A. Ca urmare a apelurilor $f(12, 15)$ și $f(8, 12)$ nu se afișează același șir de caractere
- B. Ca urmare a apelurilor $f(15, 12)$ și $f(12, 8)$ se afișează același șir de caractere
- C. Ca urmare a apelului $f(17, 23)$ nu se afișează niciun caracter "#"
- D. Ca urmare a apelului $f(23, 17)$ șirul de caractere afișat conține cel puțin un caracter "#"

18. Se consideră algoritmul $decide(n, x, t)$, unde x este un vector de n numere naturale ($2 \leq n \leq 10^4$, $1 \leq x[i] \leq 10^4$, $i = 1, 2, \dots, n$) iar t este un număr natural ($1 \leq t \leq 10^4$).

```

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

```

În care din următoarele situații algoritmul $decide(n, x, t)$ determină indicii $left, right$ ($1 \leq left < right \leq n$) astfel încât $x[left] + x[right] = t$?

- A. Dacă și numai dacă vectorul x conține numere distincte.
- B. Dacă și numai dacă vectorul x este ordonat crescător.
- C. Dacă vectorul x este ordonat descrescător și conține numere distincte.
- D. Dacă vectorul x este ordonat crescător și în vector există cel puțin o pereche de elemente având suma t .

19. Se consideră algoritmul $perechi(x, y)$, unde x și y sunt numere naturale nenule ($1 \leq x, y \leq 100$):

```

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

```

În care dintre următoarele variante de răspuns avem doar perechi de numere (x, y) pentru care algoritmul $perechi(x, y)$ afișează valorile 1 7 11?

- 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. Se consideră algoritmul $first(x, n)$, unde n este un număr natural nenul ($2 \leq n \leq 10^4$), iar x este un vector de n numere naturale ($x[1], x[2], \dots, x[n]$, unde $1 \leq x[i] \leq 10^4$, pentru $i = 1, 2, \dots, n$).

```

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

```

Cu ce ar trebui înlocuite liniile 19 și 25, astfel încât algoritmul să returneze cel mai mic număr natural nenul care nu se află în vectorul x ?

- 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. Se consideră algoritmul $ceFace(arr, n)$ unde arr este un vector cu n ($1 \leq n \leq 100$) elemente numere întregi ($arr[1]$, $arr[2]$, ..., $arr[n]$), unde $-10^5 \leq arr[i] \leq 10^5$ pentru $i = 1, 2, \dots, n$).

```

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

```

Care din următoarele afirmații sunt adevărate?

- Apelul $ceFace([11, 5, 6, 22, 0, 7, 6, 13], 8)$ returnează *True*.
- Apelul $ceFace([-5, -6, -22, -7, -6, -13], 6)$ **NU** returnează *True*.
- Dacă vectorul arr conține doar valori negative, algoritmul $auxiliar(arr, n, sum)$ va intra în ciclul infinit.
- Dacă și numai dacă elementele din arr pot fi partiționate în două mulțimi astfel încât media elementelor din cele două mulțimi să fie egală, algoritmul $ceFace(arr, n)$ returnează *True*.

22. Se consideră algoritmul $f(n, x)$, unde n este număr natural ($3 \leq n \leq 10^4$), iar x este un vector de n numere naturale ($x[1], x[2], \dots, x[n]$), unde $1 \leq x[i] \leq 10^4$, pentru $i = 1, 2, \dots, n$):

```

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

```

Care din următoarele afirmații sunt adevărate?

- În cazul apelului $f(6, [12, 16, 80, 40, 28, 144])$ algoritmul returnează valoarea 4.
- Pentru orice vector de intrare, valoarea $s1$ (calculată pe rândul 2 din algoritmul $f(n, x)$) va fi strict mai mare ca valoarea $s2$ (calculată pe rândul 6 din algoritmul $f(n, x)$).
- Dacă în algoritmul $f(n, x)$ înlocuim instrucțiunile de pe liniile 3, 4 și 5 cu secvența de mai jos, la finalul executării algoritmului $f(n, x)$ vectorul x va avea același conținut ca în algoritmul original.


```

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

```
- Există vector de intrare cu n elemente pentru care complexitatea timp a algoritmului $f(n, x)$ este $O(n)$.

23. Se consideră un număr natural nenul par n ($2 \leq n \leq 12$). Dorim să generăm în șirul de caractere x toate șirurile formate din n paranteze rotunde care se deschid și se închid corect. Algoritmul $paranteze(i, desc, inc, x, n)$ se apelează sub forma $paranteze(2, 1, 0, x, n)$, știind că au avut loc inițializările $x[1] \leftarrow '('$ și $x[n] \leftarrow ')''$. Algoritmul $afisare(n, x)$ afișează șirul de caractere x de lungime n .

```

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

```

Cu ce ar trebui completate liniile precizate mai jos, astfel încât algoritmul să afișeze doar șirurile de caractere corecte conform cerinței?

- A. Linia 5 trebuie completată cu $desc < n$, iar linia 9 trebuie completată cu $inc < desc$
- B. Linia 5 trebuie completată cu $desc < n \text{ DIV } 2$, iar linia 9 trebuie completată cu $inc < desc$
- C. Linia 5 trebuie completată cu $desc < n$, iar linia 9 trebuie completată cu $inc < n \text{ DIV } 2$
- D. Indiferent de comparațiile cu care se completează liniile 5 și 9, algoritmul nu va afișa toate șirurile de caractere conform cerinței.

24. La ora de educație fizică n copii stau unul lângă altul, cu fața către profesorul lor când acesta le cere să se întoarcă toți la stânga. Unii copii se întorc la stânga, iar alții, din greșeală, se întorc la dreapta. Într-o unitate de timp, toți copiii care se văd față în față fac stânga împrejur (adică se rotesc cu 180°), fiecare copil făcând cel mult o întoarcere. Mișcarea copiilor continuă până nu mai sunt copii situați față în față. Precizați care dintre următorii algoritmi $intoarceri(n, c)$ determină numărul unităților de timp t care trec până când nu mai sunt copii situați față în față. Variabila n este număr natural nenul ($1 \leq n \leq 100$), iar șirul de caractere c are n elemente, unde $c[i]$ este fie 's' (reprezentând „stânga”) fie 'd' (reprezentând „dreapta”) în funcție de direcția în care s-a întors al i -lea copil după comanda profesorului. *Exemple:* dacă $n = 6$ și $c = "sdsssd"$ atunci $t = 3$; dacă $n = 3$ și $c = "sdd"$ atunci $t = 0$. Algoritmul $copiaza(a, n)$ returnează o copie a vectorului a cu n elemente.

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

```


UNIVERSITATEA BABEȘ-BOLYAI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

Concurs de admitere – 6 septembrie 2024
Proba scrisă la INFORMATICĂ
BAREM ȘI REZOLVARE

OFICIU: 10 puncte

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