

Concurs de admitere – 9 septembrie 2025
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 sau a unui șir de caractere este formată din elemente care ocupă poziții consecutive în vector, respectiv în șirul de caractere.
- Dacă pe un același rând apar mai multe instrucțiuni de atribuire consecutive, acestea sunt delimitate prin ";".

1. Se consideră algoritmul $\text{cauta}(a, n, b, m)$, unde a și b sunt două șiruri de caractere cu n , respectiv m caractere ($a[1], a[2], \dots, a[n], b[1], b[2], \dots, b[m]$, $1 \leq n, m \leq 100$ și $m \leq n$).

Care dintre următoarele implementări ale algoritmului $\text{cauta}(a, n, b, m)$ returnează poziția din șirul a începând de la care apare prima dată ca subsecvență șirul b în șirul a sau -1 dacă șirul b nu apare în șirul a ?

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. Se consideră algoritmul $S(n)$, unde n este un număr întreg ($0 \leq n \leq 10^4$).

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

Care dintre afirmațiile următoare sunt adevărate?

- A. Complexitatea timp a algoritmului este $O(\log n)$.
- B. Algoritmul calculează suma primelor n numere naturale.
- C. În urma apelului $S(4)$ se returnează valoarea 12.
- D. Expresia $S(2) + S(13) = S(7) + S(8)$ are valoarea *True*.

3. Care dintre următoarele implementări ale algoritmului $\text{gcd}(a, b)$ returnează cel mai mare divizor comun al numerelor naturale a și b ($1 \leq a, b \leq 100$)?

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

```
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
```

Care dintre următoarele afirmații sunt adevărate?

- A. Dacă $n = 10, x = [1, 2, 3, \dots, 10]$, iar $k = 3$, algoritmul returnează 6.
- B. Dacă $n = 10, x = [1, 2, 3, \dots, 10]$, iar $k = 117$, algoritmul returnează 8.
- C. Dacă $n = 5, x = [4, 2, 5, 11, 13]$, iar $k = 117$, algoritmul returnează 5.
- D. Dacă $n = 5, x = [1, 2, 3, 4, 5]$, algoritmul returnează aceeași valoare dacă $k = 3$ sau $k = 318$.

5. Se consideră algoritmul $\text{ceFace}(n)$, unde n este număr natural ($1 \leq n \leq 10^9$).

```
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
```

Care dintre următoarele afirmații sunt adevărate?

- A. Algoritmul $\text{ceFace}(n)$ returnează *True* dacă în numărul n numărul cifrelor pare distincte este egal cu numărul cifrelor impare distincte.
- B. Dacă $n = 12235$, algoritmul $\text{ceFace}(n)$ returnează *True*.
- C. Dacă n este format doar din cifre impare, algoritmul $\text{ceFace}(n)$ returnează *True*.
- D. Dacă $n = 10^k$ unde k este număr par, algoritmul $\text{ceFace}(n)$ returnează *True*.

6. Se consideră algoritmul $\text{numar}(n)$, unde n este număr natural ($1 \leq n \leq 10^6$).

```
Algorithm numar(n):  
  While n > 9 execute  
    n ← n DIV 10 + n MOD 10  
  EndWhile  
  Return n  
EndAlgorithm
```

Care dintre următoarele afirmații sunt adevărate?

- A. Valoarea returnată de algoritm este în mulțimea $\{1, 2, \dots, 9\}$, pentru orice valoare a lui n .
- B. Algoritmul nu returnează valoarea 0 pentru nicio valoare a numărului n .
- C. Există 3 valori distincte v pentru care, dacă $\text{numar}(n) = v$, atunci n este divizibil cu v , indiferent de valoarea lui n .
- D. Algoritmul returnează suma cifrelor numărului n .

7. Considerăm numărul binar $b = 110101011_{(2)}$.

Care dintre următoarele afirmații sunt adevărate referitoare la numărul b ?

- A. În baza 4 are valoarea $12223_{(4)}$.
- B. În baza 8 este un număr palindrom.
- C. Este un număr impar.
- D. Este un număr divizibil cu 3.

8. Se consideră algoritmul $f(a, b, n, d)$, unde a și b sunt numere naturale nenule ($1 \leq a, b \leq 10^9$), $n = 3$ și $d = [5, 7, 11]$:

```
Algorithm f(a, b, n, d):  
  c ← 0  
  p ← 1  
  g ← False  
  While NOT g execute  
    For i ← 1, n execute  
      If (a MOD d[i] = 0) OR (b MOD d[i] = 0) then  
        c ← c * 10 + 1  
        p ← p * i  
      EndIf  
    EndFor  
    g ← NOT g  
  EndWhile  
  Return c * 100 + p  
EndAlgorithm
```

Care dintre următoarele afirmații sunt adevărate?

- A. Dacă $a * b$ nu se divide cu 5 sau cu 7 sau cu 11, în urma apelului $f(a, b, 3, [5, 7, 11])$ se returnează 1.
- B. Dacă $a * b$ nu se divide cu 5 sau cu 7 sau cu 11, apelând algoritmul $f(a, b, 3, [5, 7, 11])$, ultima valoare atribuită lui c este 10.
- C. Dacă $a * b$ se divide cu 5 și cu 7 și cu 11, în urma apelului $f(a, b, 3, [5, 7, 11])$ se returnează 11106.
- D. În urma apelului $f(112233, 331122, 3, [5, 7, 11])$ se returnează 1003.

9. 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 $i = 1, 2, \dots, n$):

```
Algorithm f(x, n):  
  For i ← 1, n - 1 execute  
    If x[i + 1] < x[i] then  
      tmp ← x[i + 1]  
      x[i + 1] ← x[i]  
      x[i] ← tmp  
    EndIf  
  EndFor  
EndAlgorithm
```

Care din următoarele afirmații sunt adevărate?

- A. În urma apelului $f(x, n)$, cel mai mare element din vectorul x va fi pe poziția n .
- B. În urma apelului $f(x, n)$, cel mai mic element din vectorul x va fi pe poziția n .
- C. În urma apelului $f(x, n)$, vectorul x va fi sortat crescător.
- D. În urma apelului $f(x, n)$, vectorul x va fi sortat descrescător.

10. Se consideră algoritmul $F(n)$, unde n este număr întreg ($0 < n < 10^5$).

```
Algorithm F(n):  
  If n = 0 then  
    Return 0  
  Else  
    If n MOD 2 = 0 then  
      Return F(n DIV 10) + 1  
    Else  
      Return F(n DIV 10) - 1  
    EndIf  
  EndIf  
EndAlgorithm
```

Care dintre următoarele afirmații sunt adevărate?

- A. Dacă $n = 543$, algoritmul returnează 2.
- B. Dacă $n = 18$, algoritmul returnează 0.
- C. Dacă $n = 41173$, algoritmul returnează 3.
- D. Pentru niciun număr n din intervalul $[111, 999]$, algoritmul $F(n)$ nu returnează 0.

11. Se consideră algoritmul $\text{ceFace}(x, n)$, unde n este număr natural ($1 \leq n \leq 10^4$), iar x este un vector cu n elemente numere întregi ($x[1], x[2], \dots, x[n]$, $0 \leq x[i] \leq 10^4$, pentru $i = 1, 2, \dots, n$):

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

```

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

```

În care din următoarele condiții algoritmul $\text{ceFace}(x, n)$ returnează *True*?

- A. Dacă numărul de numere pare din vector este egal cu numărul de numere impare din vector.
- B. Dacă numărul de numere din vector divizibile cu 2 este egal cu numărul de numere din vector multipli ai lui 6.
- C. Dacă numărul de numere pare din vector este egal cu numărul de numere impare din vector care au ultima cifră multiplu de 3.
- D. Dacă numărul de numere din vector divizibile cu 2 este egal cu numărul de numere din vector multipli ai lui 3.

12. Se consideră algoritmul $f(n, x, c)$, unde n este număr natural ($1 \leq n \leq 10^4$), x este un vector cu n elemente numere întregi ($x[1], x[2], \dots, x[n]$, $-10^3 \leq x[i] \leq 10^3$, pentru $i = 1, 2, \dots, n$), iar c este un număr natural ($1 \leq c \leq n$):

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

```

Care dintre următoarele afirmații sunt adevărate?

- A. În urma apelului $f(5, [5, 4, 30, 5, 1], 3)$ algoritmul va returna 5.
- B. Dacă x conține valorile 100, 99, 98, ..., 1, $n = 100$, iar $c = 50$, algoritmul $f(n, x, c)$ va returna 51.
- C. Dacă $c = n$, algoritmul $f(n, x, c)$ va returna elementul minim din vectorul x .
- D. Dacă $c = 1$, algoritmul $f(n, x, c)$ va returna elementul minim din vectorul x .

13. Se consideră algoritmul $\text{af1a}(n, x)$, unde n este număr natural ($3 \leq n \leq 10^4$), 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 $\text{af1a}(n, x)$:

```

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

```

Care dintre următoarele afirmații sunt adevărate în urma apelului $\text{af1a}(n, x)$?

- A. Dacă toate elementele vectorului x sunt negative, se returnează valoarea elementului $x[1]$.
- B. Dacă toate elementele vectorului x sunt pozitive, dar mai mici sau egale cu $x[1]$, se returnează valoarea elementului $x[1]$.
- C. Dacă $x[1], x[2]$ și $x[3]$ sunt numere pozitive, se returnează suma lor.
- D. Pentru orice vector x cu n elemente, unde produsul tuturor elementelor este un număr impar, algoritmul $\text{af1a}(n, x)$ returnează un număr impar.

14. Precizați câte grafuri neorientate distincte cu 5 noduri, numerotate de la 1 la 5, se pot construi, astfel încât nodul 3 să aibă gradul 1. Două grafuri sunt distincte dacă matricele lor de adiacență sunt diferite.

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

15. Se consideră algoritmul $\text{ceFace}(A, n)$, unde n este număr natural ($1 < n \leq 10$), A este o matrice cu $n \times n$ elemente numere naturale ($A[1][1], A[1][2], \dots, A[n][n]$, $1 \leq A[i][j] \leq 10$, pentru $i, j = 1, 2, \dots, n$).

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

```

Dacă $n = 4$ și matricea inițială A este $\begin{pmatrix} 4 & 8 & 3 & 4 \\ 2 & 5 & 6 & 8 \\ 4 & 3 & 2 & 9 \\ 5 & 3 & 1 & 7 \end{pmatrix}$, care dintre următoarele afirmații sunt adevărate în urma apelului

$\text{ceFace}(A, n)$?

- A. Suma elementelor de pe diagonala principală în matricea modificată este 95.
- B. În matricea modificată există de două ori mai multe elemente divizibile cu 2 decât în matricea inițială.
- C. Valoarea returnată este 1.
- D. Suma elementelor de pe diagonala principală în matricea modificată este 67.

16. Se consideră algoritmul $\text{buildMatrix}(n, x, m, y)$, unde n și m sunt numere naturale ($1 \leq n, m \leq 100$), x și y sunt vectori cu n , respectiv m elemente numere întregi ($x[1], x[2], \dots, x[n]$, unde $-10^3 \leq x[i] \leq 10^3$, pentru $i = 1, 2, \dots, n$ și $y[1], y[2], \dots, y[m]$, unde $-10^3 \leq y[j] \leq 10^3$, pentru $j = 1, 2, \dots, m$). Algoritmul $\text{max}(a, b)$ returnează valoarea maximă dintre numerele a și b . Algoritmul $\text{zeros}(n, m)$ returnează o matrice cu n linii și m coloane, cu toate elementele egale cu 0.

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

```

Care dintre următoarele afirmații sunt adevărate?

- A. În urma apelului $\text{buildMatrix}(3, [3, 2, 3], 3, [2, 3, 3])$, se returnează valoarea 2.
- B. Algoritmul $\text{buildMatrix}(n, x, m, y)$ returnează valoarea 0 dacă și numai dacă x și y au aceeași lungime și nu au nici un element comun.
- C. Dacă $n = m$, iar elementele vectorului y reprezintă o permutare a elementelor vectorului x , atunci algoritmul $\text{buildMatrix}(n, x, m, y)$ va returna valoarea n .
- D. Există date de intrare pentru care valoarea returnată de algoritm este egală cu $n + m$.

17. Se consideră algoritmul $h(A, n, p)$, unde n și p sunt numere naturale ($1 \leq n, p \leq 10^3$), iar A este un vector cu n elemente numere naturale ($A[1], A[2], \dots, A[n]$, unde $1 \leq A[i] \leq 100$, pentru $i = 1, 2, \dots, n$):

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

```

În urma căror apeluri se va returna valoarea 3?

- 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. Se consideră algoritmul `matrice(mat, n)`, unde *mat* este o matrice cu *n* linii și *n* coloane ($3 \leq n \leq 200$) cu elemente numere naturale (*mat*[1][1], *mat*[1][2], ..., *mat*[*n*][*n*], $1 \leq \text{mat}[i][j] \leq 10^4$, pentru $i = 1, 2, \dots, n, j = 1, 2, \dots, n$). Operatorul "/" reprezintă împărțirea numerelor reale; de exemplu $7 / 2 = 3.5$.

```
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
```

Cu ce instrucțiuni trebuie înlocuite liniile marcate (1) și (2) în algoritmul `matrice(mat, n)` astfel încât acesta să returneze media elementelor care nu se află pe diagonalele matricei?

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. Se consideră algoritmul `f(x, n)`, unde *n* este număr natural ($2 \leq n \leq 10^3$), iar *x* este un vector cu *n* elemente numere naturale (*x*[1], *x*[2], ..., *x*[*n*], $0 \leq x[i] \leq 100$, pentru $i = 1, 2, \dots, n$).

```
Algorithm f(x, n):
  left ← 1
  right ← n
  g ← True
  While left < right execute
    max ← -1
    For i ← left, right execute
      If x[i] > 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[poz1] ← x[poz2] // (*)
      x[poz2] ← max
    EndIf
    g ← NOT g
  EndWhile
  Return x
EndAlgorithm
```

Care dintre următoarele afirmații sunt adevărate?

- A. Fiind dat vectorul *x* cu *n* elemente, apelul `f(x, n)` returnează același rezultat pentru oricare permutare a elementelor vectorului.
- B. Dacă $n = 10$ și $x = [1, 2, 3, \dots, 10]$, în urma apelului `f(x, n)` linia marcată cu (*) se execută de 9 ori.
- C. Dacă vectorul *x* este ordonat crescător, în urma apelului `f(x, n)` linia marcată cu (*) se execută de același număr de ori ca și în cazul în care vectorul *x* este ordonat descrescător.
- D. Dacă $n = 100$ și $x = [1, 2, 3, \dots, 100]$, există o singură permutare a elementelor vectorului *x*, pentru care în urma apelului `f(x, n)` linia marcată cu (*) nu este executată.

20. Se consideră algoritmul `rezultat(azi, zile)`, unde *azi* ($1 \leq \text{azi} \leq 5$), este un număr natural care reprezintă o zi lucrătoare a săptămânii (valoarea 1 corespunde zilei de luni, valoarea 2 corespunde zilei de marți, ..., valoarea 5 corespunde zilei de vineri, respectiv), iar *zile* ($1 \leq \text{zile} \leq 100$) este un număr natural.

Algoritmul `rezultat(azi, zile)` returnează ziua lucrătoare în care este eliberat rezultatul unei analize, luând în considerare timpul de procesare necesar, exprimat în zile calendaristice prin valoarea variabilei *zile*. Analizele sunt procesate în fiecare zi a săptămânii, iar rezultatele sunt eliberate în prima zi lucrătoare în care sunt disponibile. Dacă procesarea se încheie într-o zi de weekend (sâmbătă sau duminică), rezultatul este eliberat în următoarea zi de luni.

De exemplu, în urma apelului `rezultat(2, 1)` se va returna valoarea 2 (rezultatul se eliberează în aceeași zi), iar în urma apelului `rezultat(3, 4)` se va returna valoarea 1 (rezultatul se eliberează luni).

Algorithm rezultat(azi, zile): Care dintre următoarele afirmații sunt adevărate?

```

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. Algoritmul rezultatRec(azi, zile) returnează aceeași valoare ca și algoritmul rezultat(azi, zile) pentru orice date de intrare.

```

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. În urma apelului rezultat(4, 25) se returnează valoarea 1.

C. Algoritmul rezultat2(azi, zile) returnează aceeași valoare ca algoritmul rezultat(azi, zile) pentru orice date de intrare, și are o complexitate timp mai mică:

```

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

```

D. Complexitatea timp a algoritmului rezultat(azi, zile) este $O(1)$.

21. Se consideră algoritmul suma(A, n, m, x1, y1, x2, y2) care calculează suma elementelor dintr-o regiune a matricei A. A este o matrice de dimensiuni $n \times m$ ($A[1][1], A[1][2], \dots, A[1][m], A[2][1], \dots, A[n][m]$, unde $2 < n, m < 100$, $0 \leq A[i][j] \leq 10^3$ pentru $i = 1, 2, \dots, n, j = 1, 2, \dots, m$). Regiunea de interes dreptunghiulară este definită de coordonatele colțului din stânga sus ($x1, y1$) și ale colțului din dreapta jos ($x2, y2$), unde $1 \leq x1 < x2 \leq n, 1 \leq y1 < y2 \leq m$. Algoritmul zeros(a, b) returnează o matrice cu a linii și b coloane, unde toate elementele sunt inițializate cu 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, x1, y1, x2, y2):
2.     aux ← compute(A, n, m)
3.     S1 ← 0; S2 ← 0; S3 ← 0; S4 ← 0
4.     If x1 > 1 AND y1 > 1 then
5.         S1 ← aux[x1 - 1][y1 - 1]
6.     EndIf
7.     If x1 > 1 then
8.         S2 ← aux[x1 - 1][y2]
9.     EndIf
10.    If y1 > 1 then
11.        S3 ← aux[x2][y1 - 1]
12.    EndIf
13.    S4 ← aux[x2][y2]
14.    .....
15. EndAlgorithm

```

Ce instrucțiune trebuie inserată la linia 14 pentru ca algoritmul suma(A, n, m, x1, y1, x2, y2) să returneze suma cerută?

- 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. 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 ($x[1], x[2], \dots, x[n]$, $1 \leq x[i] \leq 10^4$, pentru $i = 1, 2, \dots, n$).

```
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
```

Care din următoarele afirmații sunt adevărate?

- A. În urma apelului $f(5, [12, 16, 8, 40, 24])$ se returnează valoarea 240.
- B. Numărul total de apeluri al algoritmului $g(a, b)$ este mai mare în cazul apelului $f(5, [8, 12, 16, 24, 40])$ decât în cazul apelului $f(5, [40, 24, 16, 12, 8])$.
- C. Algoritmul calculează cel mai mare divizor comun al tuturor elementelor din vectorul x .
- D. Pentru un vector x cu $n = 10$ elemente care conține o singură valoare egală cu 17, restul valorilor fiind egale cu 1, numărul maxim de apeluri ale algoritmului $g(a, b)$, incluzând și apelul inițial, este 153.

23. Pentru o partidă de fotbal între echipele **A** și **B** în care se înscriu 3 goluri și care se termină cu scorul de 2 - 1 pentru echipa **A**, golurile pot fi înscrise în 3 succesiuni:

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

De exemplu, în succesiunea 1., partida începe de la scorul de 0 - 0, înscrie echipa **B** (0 - 1), egalează echipa **A** (1 - 1), și înscrie echipa **A** (2 - 1).

Câte succesiuni posibile există pentru o partidă cu 4 goluri?

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

24. Se consideră algoritmul $litere(v, n)$, unde v este un șir de caractere cu n elemente ($1 \leq n \leq 100$) care sunt majuscule din alfabetul englezesc. Pentru fiecare element de pe poziția i ($1 \leq i \leq n$) din șirul de caractere se calculează o valoare $S[i]$ care reprezintă numărul pozițiilor $j \in \{1, 2, \dots, i\}$ pe care se află litere care sunt în fața literei $v[i]$ în alfabet. De exemplu, dacă $v = ['E', 'X', 'A', 'M', 'E', 'N']$, valoarea $S[6]$ este 4, deoarece doar literele 'E', 'A' și 'M' sunt înaintea literei 'N' în alfabet. Pentru tipul caracter, operațiile de comparație și scădere se efectuează pe codul ASCII al operanzilor.

Care dintre următorii algoritmi returnează suma tuturor valorilor $S[i]$ ($i = 1, 2, \dots, n$)?

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
```

UNIVERSITATEA BABEȘ-BOLYAI CLUJ-NAPOCA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

Concurs de admitere – 9 septembrie 2025
Proba scrisă la INFORMATICĂ

BAREM ȘI REZOLVARE

OFICIU: 10 puncte

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