

Concurs Mate-Info – 12 aprilie 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 este formată din elemente care ocupă poziții consecutive în vector.
- Dacă pe un același rând apar mai multe instrucțiuni de atribuire consecutive, acestea sunt delimitate prin ";".

1. Se consideră algoritmul $\text{calcul}(n, c1, c2)$, unde n este număr natural ($1 \leq n \leq 10^4$), iar $c1$ și $c2$ sunt cifre ($0 \leq c1, c2 \leq 9$).

Algorithm $\text{calcul}(n, c1, c2)$:

If $n = 0$ then

Return 0

EndIf

If $n \text{ MOD } 10 = c1$ then

Return $\text{calcul}(n \text{ DIV } 10, c1, c2) * 10 + c2$

Else

Return $\text{calcul}(n \text{ DIV } 10, c1, c2) * 10 + n \text{ MOD } 10$

EndIf

EndAlgorithm

Ce returnează algoritmul pentru $n = 1999$, $c1 = 1$ și $c2 = 0$?

- A. 1000
- B. 999
- C. 1099
- D. 1990

2. Se consideră algoritmul $\text{ceFace}(m, n)$, unde m și n sunt numere naturale ($1 \leq m, n \leq 100$):

1. Algorithm $\text{ceFace}(m, n)$:

2. $c \leftarrow 1$; $i \leftarrow n$

3. While $i > 0$ execute

4. If $i \text{ MOD } 2 = 1$ then

5. $c \leftarrow c * m$

6. $i \leftarrow i - 1$

7. Else

8. $m \leftarrow m * m$

9. $i \leftarrow i \text{ DIV } 2$

10. EndIf

11. EndWhile

12. Return c

13. EndAlgorithm

Care dintre următoarele afirmații sunt adevărate?

- A. În urma apelului $\text{ceFace}(2, 5)$ algoritmul returnează 30.
- B. Dacă în urma apelului $\text{ceFace}(m, n)$ algoritmul returnează valoarea x , nu există altă pereche de valori $m1, n1$ ($m1 \neq m$ și $n1 \neq n$) pentru care apelul $\text{ceFace}(m1, n1)$ să returneze aceeași valoare x .
- C. Singura valoare a lui n pentru care linia 6 se execută de 2 ori în urma apelului algoritmului $\text{ceFace}(m, n)$ este 5.
- D. În urma apelului $\text{ceFace}(5, 8)$ linia 6 se execută o singură dată.

3. Se consideră algoritmul $\text{ceFace}(b, n, a)$, unde b și n sunt numere naturale ($2 \leq b, n \leq 100$), iar a este un vector cu n elemente, numere naturale ($a[1], a[2], \dots, a[n]$, $0 \leq a[i] < b$, pentru $i = 2, 3, \dots, n$ și $0 < a[1] < b$):

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

$v \leftarrow a[1]$

For $i \leftarrow 2, n$ execute

$v \leftarrow v * b + a[i]$

EndFor

Return v

EndAlgorithm

Care dintre următoarele afirmații sunt adevărate?

- A. Apelul $\text{ceFace}(2, 6, [1, 0, 1, 0, 1, 1])$ returnează valoarea 43.
- B. Apelul $\text{ceFace}(9, 3, [7, 6, 5])$ returnează valoarea 626.
- C. Dacă $a[n] = 0$, apelul $\text{ceFace}(b, n, a)$ returnează un număr par.
- D. Dacă $b1 > b2$, atunci apelul $\text{ceFace}(b1, n, a)$ returnează un număr mai mare decât apelul $\text{ceFace}(b2, n, a)$.

4. Se consideră numărul întreg n , ($-100 \leq n \leq 100$).

Care dintre următoarele expresii au valoarea *True* dacă și numai dacă n NU aparține mulțimii: $\{-8\} \cup \{-4, -3, \dots, 8\}$?

- A. $(n \leq -8) \text{ AND } (n \geq -8) \text{ AND } (n \leq -4) \text{ AND } (n \geq 8)$
- B. $(n < -8) \text{ OR } ((n > -8) \text{ AND } (n < -4)) \text{ OR } (n > 8)$
- C. $(n < -8) \text{ OR } ((n > -8) \text{ OR } (n < -4)) \text{ AND } (n > 8)$
- D. $((n < -4) \text{ AND } (n \neq -8)) \text{ OR } (n > 8)$

5. Se consideră algoritmul $\text{cautBin}(st, dr, y, x, n)$, unde st și dr sunt numere naturale, x este un vector ordonat crescător cu n elemente numere întregi ($1 \leq st, dr, n \leq 10^4, x[1], x[2], \dots, x[n], -10^3 \leq x[i] \leq 10^3$ pentru $i = 1, 2, \dots, n$) și y este un număr întreg ($-10^3 \leq y \leq 10^3, x[1] < y$) care nu face parte din vector. Apelul algoritmului are forma: $\text{cautBin}(1, n, y, x, n)$.

```

Algorithm cautBin(st, dr, y, x, n):
  If st < dr then
    mij ← (st + dr) DIV 2
    If y < x[mij] then
      Return cautBin(st, mij, y, x, n)
    Else
      Return cautBin(mij + 1, dr, y, x, n)
    EndIf
  Else
    .....
  EndIf
EndAlgorithm

```

Cu ce trebuie înlocuite punctele de suspensie astfel încât algoritmul dat să returneze poziția celui mai apropiat număr din vector, mai mare decât y . Dacă un astfel de număr nu există, se returnează -1.

- | | | | |
|----|--|----|--|
| A. | <pre> If y > x[dr] then Return dr + 1 Else Return -1 EndIf </pre> | B. | <pre> If y < x[dr] then Return dr Else Return -1 EndIf </pre> |
| C. | <pre> If y > x[st] then Return st + 1 Else Return -1 EndIf </pre> | D. | <pre> If y < x[st] then Return st Else Return -1 EndIf </pre> |

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

```

Algorithm calculeaza(x, n):
  If n MOD 2 = 1 then
    s ← x[n]
  Else
    s ← 0
  EndIf
  For i ← 1, n - 2, 2 execute
    s ← s + x[i] + x[i + 1]
  EndFor
  Return s
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- În urma apelului $\text{calculeaza}([3, -8, -2, 15, -1, 0, 3, 1, 3], 9)$, algoritmul returnează 11.
- În urma apelului $\text{calculeaza}([2, -1, 7, 5, -9, 0, 3, 1, 12], 9)$, algoritmul returnează 4.
- În urma apelului $\text{calculeaza}([10, 2, 5, 78, 23, 4, 11], 7)$, algoritmul returnează 133.
- În urma apelului $\text{calculeaza}([-3, 8, -2, 15, -1, 10], 6)$, algoritmul returnează 27.

7. Se consideră algoritmul $f(n, a, p)$, unde n și p sunt numere naturale ($1 \leq n, p \leq 10^5$) și a este un vector ($a[1], a[2], \dots, a[n], 0 \leq a[i] \leq 9$, pentru $i = 1, 2, \dots, n$) de n cifre, unde cel puțin o cifră diferă de 0:

```

Algorithm f(n, a, p):
  s ← 0
  For i ← 1, n execute
    s ← s + a[i]
  EndFor
  For i ← 1, p execute
    If s MOD 3 = 0 then
      s ← s DIV 3
    Else
      Return False
    EndIf
  EndFor
  Return True
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- Algoritmul returnează *True* dacă și numai dacă suma elementelor vectorului a este multiplu de 3^p .
- Algoritmul returnează *True* dacă și numai dacă suma elementelor vectorului a este o putere a lui 3.
- Algoritmul returnează *False* dacă și numai dacă suma elementelor vectorului a nu este divizibil cu 3.
- În urma apelului $f(6, [9, 1, 8, 8, 4, 6], 2)$ algoritmul returnează *True*.

8. Numărul maxim de muchii dintr-un graf neorientat cu n noduri și p ($0 < p \leq n$) componente conexe este:

- | | | | | | | | |
|----|----------------------------------|----|------------------------|----|----------------------------------|----|----------------------------------|
| A. | $\frac{(n-p) \times (n-p+1)}{2}$ | B. | $(n-p) \times (n-p+1)$ | C. | $\frac{(n-p) \times (n-p+1)}{4}$ | D. | $\frac{(n-p) \times (n-p+1)}{2}$ |
|----|----------------------------------|----|------------------------|----|----------------------------------|----|----------------------------------|

9. Se dă un număr natural n ($10 \leq n \leq 10^4$).

Care dintre următoarele implementări ale algoritmului $f(n)$ returnează oglinditul numărului n ?

A.
Algorithm $f(n)$:
 If $n > 0$ **then**
 Return $n \text{ MOD } 10 + 10 * f(n \text{ DIV } 10)$
 EndIf
 Return 0
EndAlgorithm

B.
Algorithm $f1(n, \text{ogl})$:
 If $n > 0$ **then**
 Return $f1(n \text{ DIV } 10, n \text{ MOD } 10 + 10 * \text{ogl})$
 EndIf
 Return ogl
EndAlgorithm

C.
Algorithm $f(n)$:
 $\text{ogl} \leftarrow 0$
 While $n > 0$ **execute**
 $\text{ogl} \leftarrow (n \text{ MOD } 10) * 10 + \text{ogl}$
 $n \leftarrow n \text{ DIV } 10$
 EndWhile
 Return ogl
EndAlgorithm

Algorithm $f(n)$:
 Return $f1(n, 0)$
EndAlgorithm
D.
Algorithm $f(n)$:
 $\text{ogl} \leftarrow 0$
 While $n > 0$ **execute**
 $\text{ogl} \leftarrow \text{ogl} * 10 + n \text{ MOD } 10$
 $n \leftarrow n \text{ DIV } 10$
 EndWhile
 Return ogl
EndAlgorithm

10. Se consideră algoritmul $\text{ceFace}(x1, y1, x2, y2, x3, y3)$, unde $(x1, y1)$, $(x2, y2)$ și $(x3, y3)$ sunt coordonatele a trei puncte geometrice distincte.

Algorithm $\text{ceFace}(x1, y1, x2, y2, x3, y3)$:
 $t \leftarrow x1 * (y2 - y3)$
 $v \leftarrow x2 * (y1 - y3)$
 $z \leftarrow x3 * (y1 - y2)$
 Return $(t - v + z) \neq 0$
EndAlgorithm

Care dintre următoarele afirmații sunt adevărate pentru apelul algoritmului $\text{ceFace}(x1, y1, x2, y2, x3, y3)$?

- A. Returnează *True* dacă punctele date formează un triunghi nedegenerat.
- B. Returnează *False* dacă punctele date sunt coliniare.
- C. Returnează *False* dacă punctele date formează un triunghi nedegenerat.
- D. Returnează *True* dacă punctele date sunt coliniare.

11. Se consideră algoritmul $h(A, n)$, 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 $0 \leq A[i] \leq 100$, pentru $i = 1, 2, \dots, n$):

Algorithm $h(A, n)$:
 If $n = 0$ **then**
 Return 0
 EndIf
 Return $h(A, n - 1) + (A[n] \text{ MOD } 2) * (A[n] \text{ MOD } 10) * (n \text{ MOD } 2)$
EndAlgorithm

În urma căror apeluri se returnează valoarea 0?

- A. $h([25, 14, 35, 26, 2, 10], 6)$
- B. $h([14, 25, 26, 2, 10, 35], 6)$
- C. $h([12, 5, 22, 4, 32, 8, 46, 9, 54, 3], 10)$
- D. $h([3, 4, 7], 3)$

12. Se consideră algoritmul $\text{ceFace}(n)$, unde n este un număr natural ($0 \leq n \leq 10$).

1. **Algorithm** $\text{ceFace}(n)$:
2. $e \leftarrow 1$
3. **For** $f \leftarrow 1, n$ **execute**
4. $s \leftarrow 0$
5. **For** $j \leftarrow 1, f$ **execute**
6. $s \leftarrow s + j$
7. **EndFor**
8. $e \leftarrow e * s$
9. **EndFor**
10. **Return** e
11. **EndAlgorithm**

Care dintre următoarele afirmații sunt adevărate?

- A. În urma apelului $\text{ceFace}(5)$, algoritmul returnează valoarea 2700.
- B. Indiferent de valoarea lui n , algoritmul $\text{ceFace}(n)$ nu returnează niciodată valoarea 0.
- C. Valoarea returnată în urma apelului $\text{ceFace}(9)$ are același număr de zerouri la final ca valoarea returnată în urma apelului $\text{ceFace}(10)$.
- D. În urma apelului $\text{ceFace}(10)$ rândul 6 se execută de 45 de ori.

13. Se consideră algoritmul $ceva(n)$ unde n este un număr natural ($1 \leq n \leq 10^9$).

```

Algorithm aux(n):
  v1 ← 1
  v2 ← 1
  While v1 < n execute
    v3 ← v1 + v2
    v1 ← v2
    v2 ← v3
  EndWhile
  Return v1 = n
EndAlgorithm

```

```

Algorithm ceva(n):
  If aux(n) then
    Return True
  EndIf
  p ← 10
  gata ← False
  While (n DIV p ≠ 0) AND (NOT gata) execute
    nr1 ← n MOD p
    nr2 ← (n - nr1) DIV p
    If aux(nr1) then
      gata ← ceva(nr2)
    EndIf
    p ← p * 10
  EndWhile
  Return gata
EndAlgorithm

```

Considerând că primele șase numere din șirul *Fibonacci* sunt 1, 1, 2, 3, 5, 8, care dintre următoarele afirmații sunt adevărate?

- A. Algoritmul $ceva(n)$ returnează *True* dacă și numai dacă n este un număr *Fibonacci*.
- B. Algoritmul $ceva(n)$ verifică dacă n poate fi scris ca sumă de numere *Fibonacci*.
- C. Algoritmul $ceva(n)$ verifică dacă n poate fi scris ca produs de numere *Fibonacci*.
- D. Dacă $n = 1234589$, atunci algoritmul $ceva(n)$ returnează *True*.

14. Se consideră algoritmul $ceFace(n, f, p)$, unde n este număr natural ($0 \leq n \leq 10^{10}$), p este număr natural ($0 \leq p \leq 100$) și f este număr întreg ($-1 \leq f \leq 1$).

```

Algorithm ceFace(n, f, p):
  If n = 0 then
    Return f = 1
  EndIf
  c ← n MOD 10
  n ← n DIV 10
  If f = -1 then
    If c < p then
      Return ceFace(n, 0, c)
    Else
      Return False
    EndIf
  EndIf
  If f = 0 then
    If c < p then
      Return ceFace(n, 0, c)
    Else
      If c > p then
        Return ceFace(n, 1, c)
      Else
        Return False
      EndIf
    EndIf
  EndIf
  If f = 1 then
    If c > p then
      Return ceFace(n, 1, c)
    Else
      Return False
    EndIf
  EndIf
EndAlgorithm

```

Care dintre următoarele afirmații referitoare la rezultatul apelului $ceFace(n \text{ DIV } 10, -1, n \text{ MOD } 10)$ sunt adevărate?

- A. Pentru orice valoare a lui $n < 101$ se returnează *False*.
- B. Pentru $n = 8976532014$ se returnează *True*.
- C. Dacă n conține cel puțin două cifre egale se returnează *False*.
- D. Dacă n nu conține cifra 0 și apelul returnează *True*, atunci apelul va returna *True* și pentru oglinditul lui n .

15. Pentru a determina o cifră care apare de cele mai multe ori într-un număr, implementăm trei algoritmi: cifreA(n), cifreB(n) și cifreC(n) unde n este un număr natural ($1 \leq n \leq 10^{12}$).

```

Algorithm cifreA(n):
  c ← n
  maxf ← -1; maxd ← -1
  While c > 0 execute // (*)
    d ← c MOD 10
    copie ← n
    cnt ← 0
    While copie > 0 execute
      If copie MOD 10 = d then
        cnt ← cnt + 1
      EndIf
      copie ← copie DIV 10
    EndWhile
    If cnt > maxf then
      maxf ← cnt
      maxd ← d
    EndIf
    c ← c DIV 10
  EndWhile
  Return maxd
EndAlgorithm

```

```

Algorithm cifreC(n):
  maxf ← -1; maxd ← -1
  For i ← 9, 0, -1 execute
    c ← n; cnt ← 0
    While c > 0 execute
      If c MOD 10 = i then
        cnt ← cnt + 1
      EndIf
      c ← c DIV 10
    EndWhile
    If cnt > maxf then
      maxf ← cnt
      maxd ← i
    EndIf
  EndFor
  Return maxd
EndAlgorithm

```

16. Se consideră algoritmul getSomeMax(n , x), unde n este număr natural ($1 \leq n \leq 10^3$), iar x este un vector cu n elemente numere întregi ($x[1], x[2], \dots, x[n]$, unde $-10^3 \leq x[i] \leq 10^3$, pentru $i = 1, 2, \dots, n$). Algoritmul zero(k) returnează un vector cu k elemente, toate egale cu zero.

```

Algorithm getSomeMax(n, x):
  y ← zero(n + 1)
  For i ← 1, n execute
    y[i + 1] ← y[i] + x[i]
  EndFor
  sm ← y[2]
  For i ← 2, n execute
    For j ← i, n execute
      s ← y[j] - y[i - 1]
      If s > sm then
        sm ← s
      EndIf
    EndFor
  EndFor
  Return sm
EndAlgorithm

```

```

Algorithm cifreB(n):
  maxf ← -1
  maxd ← -1
  For i ← 0, 9 execute // (*)
    c ← n
    cnt ← 0
    While c > 0 execute
      If c MOD 10 = i then
        cnt ← cnt + 1
      EndIf
      c ← c DIV 10
    EndWhile
    If cnt > maxf then
      maxf ← cnt
      maxd ← i
    EndIf
  EndFor
  Return maxd
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- cifreA(123453) = cifreB(123453) = cifreC(123453)
- cifreA(123456) = cifreB(123456) = cifreC(123456)
- Există cel puțin un număr n pentru care cei trei algoritmi returnează trei valori diferite.
- Pentru orice număr n , ciclul while marcat cu (*) din algoritmul cifreA(n) se execută de mai puține ori decât ciclul For marcat cu (*) din algoritmul cifreB(n).

Care dintre următoarele afirmații sunt adevărate?

- Dacă $n = 1$, valoarea returnată de algoritmul getSomeMax(n , x) este valoarea lui $x[1]$.
- Valoarea returnată de algoritmul, în cazul apelului getSomeMax(8, [5, 7, -4, 6, -3, -2, 6, -7]) este 10.
- Dacă $n = 100$ și $x = [1, 2, 3, \dots, 99, 100]$, valoarea returnată de algoritmul getSomeMax(n , x) este 4950.
- Dacă toate valorile din vectorul x sunt strict negative, algoritmul getSomeMax(n , x) returnează cel mai mare element din vector.

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

```

1. Algorithm af1a(n, x):
2.   M1 ← x[1]; M2 ← x[2]; M3 ← x[3]
3.   For i ← 1, n execute
4.     If x[i] > M1 then
5.       M3 ← M2
6.       M2 ← M1
7.       M1 ← x[i]
8.     Else
9.       If x[i] > M2 then
10.        M3 ← M2
11.        M2 ← x[i]
12.      Else
13.        If x[i] > M3 then
14.          M3 ← x[i]
15.        EndIf
16.      EndIf
17.    EndIf
18.  EndFor
19.  Return M1, M2, M3
20. EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- A. În urma apelului $\text{af1a}(6, [1, 2, 3, 4, 5, 6])$ algoritmul returnează 6, 5, 4.
- B. Dacă instrucțiunile de pe liniile 8 și 12 ar fi înlocuite cu **EndIf**, iar instrucțiunile de pe liniile 16 și 17 ar fi șterse, atunci algoritmul ar returna același rezultat ca algoritmul inițial.
- C. Dacă la început **M1**, **M2** și **M3** ar lua valorile $x[3]$, $x[2]$, respectiv $x[1]$, atunci algoritmul ar returna același rezultat ca algoritmul inițial.
- D. Dacă în linia 3 în loc de **For i ← 1, n execute** am avea **For i ← 4, n execute**, atunci algoritmul ar returna același rezultat ca algoritmul inițial.

18. Se consideră algoritmul $\text{ceFace}(A, n)$, unde n este un număr natural ($1 \leq n \leq 20$), iar A este o matrice pătratică cu n linii și n coloane, care conține numere naturale: ($A[1][1], A[1][2], \dots, A[n][n]$, unde $0 \leq A[i][j] \leq 200$, pentru $i = 1, 2, \dots, n$ și $j = 1, 2, \dots, n$).

```

Algorithm ceFace(A, n):
// Început partea 1
For i ← 1, n execute
  For j ← i + 1, n execute
    temp ← A[i][j]
    A[i][j] ← A[j][i]
    A[j][i] ← temp
  EndFor
EndFor
// Sfârșit partea 1

// Început partea 2
For i ← 1, n execute
  For j ← 1, n DIV 2 execute
    temp ← A[i][j]
    A[i][j] ← A[i][n - j + 1]
    A[i][n - j + 1] ← temp
  EndFor
EndFor
// Sfârșit partea 2
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- A. În urma executării algoritmului $\text{ceFace}(A, 3)$,

$$\begin{matrix} 1 & 2 & 3 & & 7 & 4 & 1 \\ \text{matricea } A = & 4 & 5 & 6, & \text{va deveni} & 8 & 5 & 2. \\ & 7 & 8 & 9 & & 9 & 6 & 3 \end{matrix}$$
- B. Dacă matricea de intrare A este matricea de identitate de ordinul 3, atunci ea nu se modifică în urma executării algoritmului $\text{ceFace}(A, 3)$
- C. Algoritmul $\text{ceFace}(A, n)$ aplică o rotație cu 90° la dreapta asupra matricei date, pe care o modifică corespunzător.
- D. Dacă interschimbăm partea din algoritm cuprinsă între Început partea 1 și Sfârșit partea 1 cu cea cuprinsă între Început partea 2 și Sfârșit partea 2, algoritmul $\text{ceFace}(A, n)$ returnează același rezultat ca algoritmul inițial.

19. Se consideră algoritmul $\text{ceFace}(n)$, unde n este un număr natural ($0 \leq n \leq 200$).

```

1. Algorithm ceFace(n):
2.   e ← 0
3.   For i ← 1, n execute
4.     If i MOD 2 = 0 then
5.       e ← e - 2 * i * i
6.     Else
7.       e ← e + 2 * i * i
8.     EndIf
9.   EndFor
10.  Return e
11. EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- A. Pentru orice număr n par, algoritmul va returna o valoare negativă.
- B. Algoritmul calculează valoarea expresiei
 $0 + 1 * 2 - 2 * 4 + 3 * 6 - 4 * 8 + \dots + (-1)^{n-1} * n * 2 * n$
- C. Dacă algoritmul $\text{ceFace}(n)$ returnează o valoare negativă, n este un număr par.
- D. Există o singură valoare pentru n , pentru care instrucțiunea de pe linia 7 se execută de exact 7 ori.

20. Se consideră algoritmul $\text{ceFace}(n, x)$, unde n este număr natural ($2 \leq n \leq 10^3$), iar x este un vector cu n elemente numere întregi ($x[1], x[2], \dots, x[n]$, $-100 \leq x[i] \leq 100$, pentru $i = 1, 2, \dots, n$). Algoritmul $\text{zero}(k)$ returnează un vector cu k elemente, toate egale cu zero. Algoritmii $\text{minim}(n, x)$, $\text{maxim}(n, x)$ returnează valoarea minimă, respectiv maximă din vectorul x cu n elemente.

```

01. Algorithm ceFace(n, x):
02.   min ← minim(n, x)
03.   max ← maxim(n, x)
04.   r ← max - min + 1
05.   y ← zero(r)
06.   For i ← 1, n execute
07.     y[x[i] - min + 1] ← y[x[i] - min + 1] + 1
08.   EndFor
09.   idx ← 1
10.   For i ← 1, r execute
11.     While y[i] > 0 execute
12.       x[idx] ← i + min - 1
13.       idx ← idx + 1
14.       y[i] ← y[i] - 1
15.     EndWhile
16.   EndFor
17. EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- A. Dacă vectorul x conține și numere negative, algoritmul va încerca să acceseze poziții inexistente în vectorul y .
- B. Dacă am înlocui instrucțiunile de la rândurile 9 și 10 cu secvența de instrucțiuni de mai jos, algoritmul $\text{ceFace}(n, x)$ ar returna același rezultat ca algoritmul inițial.


```

x[1] ← min
idx ← 2
y[1] ← y[1] - 1
For i ← 2, r execute

```
- C. În urma apelului $\text{ceFace}(2, [5, 8])$ vectorul x devine: $x = [6, 9]$.
- D. După executarea algoritmului $\text{ceFace}(n, x)$ elementele vectorului x vor reprezenta o permutare a elementelor inițiale ale vectorului.

21. Se consideră algoritmul $p(x, n, a, b, c, d)$, unde x este un vector cu n ($0 \leq n \leq 100$) elemente întregi ($x[1], x[2], \dots, x[n]$, unde $-100 \leq x[i] \leq 100$, pentru $i = 1, 2, \dots, n$), iar a, b, c și d sunt numere întregi ($0 \leq a, b, c, d \leq 100$).

```

Algorithm p(x, n, a, b, c, d):
  If n = 0 then
    Return a = b AND c = d
  EndIf
  p1 ← p(x, n - 1, a + x[n], b, c * x[n], d)
  p2 ← p(x, n - 1, a, b + x[n], c, d * x[n])
  Return p1 OR p2
EndAlgorithm

```

Știind că $x = [2, 9, 5, 6, 8, 4, 1, 2, 5, 3, 4, 1, 9, 6, 8, 3]$, care dintre următoarele afirmații sunt adevărate?

- A. În urma apelului $p(x, 16, 0, 0, 1, 1)$, algoritmul returnează *True*.
- B. În urma apelului $p(x, 16, 0, 0, 1, 1)$, algoritmul returnează *False*.
- C. Corespunzător apelului $p(x, 16, 0, 0, 1, 1)$, algoritmul intră în ciclul infinit.
- D. În urma apelului $p(x, 16, 0, 0, 1, 1)$, algoritmul returnează același rezultat pentru oricare permutare a elementelor vectorului x .

22. Se consideră algoritmii $\text{rec}(n, x, i, j)$ și $\text{ceFace}(n, x)$, unde n este număr natural ($1 \leq n \leq 10^3$), iar x este un vector cu n elemente numere întregi ($x[1], x[2], \dots, x[n]$, $-100 \leq x[k] \leq 100$, pentru $k = 1, 2, \dots, n$), iar i și j sunt numere întregi în intervalul $[0, n]$. Algoritmul $\text{maxim}(a, b)$ returnează valoarea mai mare dintre a și b .

```

Algorithm rec(n, x, i, j):
  If i = n then
    Return 0
  EndIf
  a ← rec(n, x, i + 1, j)
  b ← 0
  If j = 0 then
    b ← 1 + rec(n, x, i + 1, i)
  Else
    If x[i] > x[j] then
      b ← 1 + rec(n, x, i + 1, i)
    EndIf
  EndIf
  Return maxim(a, b)
EndAlgorithm

```

```

Algorithm ceFace(n, x):
  Return rec(n, x, 1, 0)
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- A. Pentru un vector x ordonat strict crescător algoritmul $\text{ceFace}(n, x)$ va returna valoarea n .
- B. Complexitatea de timp a algoritmului în cazul cel mai defavorabil este $O(n^2)$.
- C. În urma apelului $\text{ceFace}(8, [10, 15, 9, 30, 21, 50, 42, 60])$ algoritmul returnează valoarea 5.
- D. În urma apelului $\text{ceFace}(2, [3, 2])$ algoritmul returnează valoarea 1.

23. Un număr n se numește *special* dacă are ca divizori primi doar numerele 2, 3 și 5. De exemplu, numere speciale sunt 1 ($1 = 2^0 * 3^0 * 5^0$), 12 ($12 = 2^2 * 3$) sau 30 ($30 = 2 * 3 * 5$). Algoritmul `zero(k)` returnează un vector cu k elemente egale cu 0. Care dintre secvențele de instrucțiuni din răspunsurile A, B, C, D trebuie inserate în algoritmul `special(n)` în locul punctelor de suspensie, astfel încât algoritmul să returneze al n -lea număr special, unde n este un număr natural ($1 \leq n \leq 10^5$)?

Algorithm `special(n)`:

```

v ← zero(n)
v[1] ← 1; c2 ← 1; c3 ← 1; c5 ← 1
nr ← 1
While nr < n execute
    val1 ← v[c2] * 2
    val2 ← v[c3] * 3
    val3 ← v[c5] * 5
    If val1 ≤ val2 AND val1 ≤ val3 then
        elem ← val1
        c2 ← c2 + 1
    Else
        If val2 ≤ val1 AND val2 ≤ val3 then
            elem ← val2; c3 ← c3 + 1
        Else
            elem ← val3
            c5 ← c5 + 1
        EndIf
    EndIf
    .....
EndWhile
Return v[n]
EndAlgorithm

```

A.

```

v[nr] ← elem
nr ← nr + 1

```

B.

```

If v[nr] < elem then
    v[nr + 1] ← elem
    nr ← nr + 1
EndIf

```

C.

```

nr ← nr + 1
v[nr] ← elem

```

D.

```

tmp ← nr
While elem < v[tmp] AND tmp ≥ 1 execute
    v[tmp + 1] ← v[tmp]
    tmp ← tmp - 1
EndWhile
v[tmp + 1] ← elem
nr ← nr + 1

```

24. Se dă un număr natural n ($0 \leq n \leq 2^{31}$) și se dorește determinarea numărului de biți din reprezentarea în baza 2 a numărului n care este reprezentat folosind exact 32 de biți, care au valoarea $k \in \{0, 1\}$. În algoritmi se utilizează operațiile pe biți: `&` (AND), `<<` (deplasare la stânga a reprezentării) și `>>` (deplasare la dreapta a reprezentării) având următoarele semnificații:

- Dacă x și y sunt două numere naturale, atunci $x \& y$ aplică operația AND pe biți în reprezentarea lor binară: fiecare bit din rezultat este 1 doar dacă ambii biți corespunzători din x și y sunt 1; în caz contrar este 0.
- Dacă x este un număr natural, operația $x \ll i$ este echivalentă cu înmulțirea cu 2 de i ori a numărului x , iar operația $x \gg i$ este echivalentă cu împărțirea întreagă cu 2 de i ori a numărului x .

Care dintre variantele de algoritmi de mai jos returnează valoarea cerută?

A. Algorithm `countBits_A(n, k)`:

```

count ← 0
For i ← 0, 31 execute
    If ((n & (1 << i)) >> i) = k then
        count ← count + 1
    EndIf
EndFor
Return count
EndAlgorithm

```

C. Algorithm `countBits_C(n, k)`:

```

If n = 0 then
    If k = 0 then
        Return 32
    Else
        Return 0
    EndIf
Else
    If (n & 1) = k then
        Return 1 + countBits_C(n >> 1, k)
    Else
        Return countBits_C(n >> 1, k)
    EndIf
EndIf
EndAlgorithm

```

B. Algorithm `countBits_B(n, k)`:

```

count ← 0
While n > 0 execute
    If (n & 1) = 1 then
        count ← count + 1
    EndIf
    n ← n >> 1
EndWhile
If k = 0 then
    count ← 32 - count
EndIf
Return count
EndAlgorithm

```

D. Algorithm `countBits(n, k, poz)`:

```

If poz < 0 then
    Return 0
Else
    If ((n & (1 << poz)) >> poz) = k then
        Return 1 + countBits(n, k, poz - 1)
    Else
        Return countBits(n, k, poz - 1)
    EndIf
EndIf
EndAlgorithm

```

Algorithm `countBits_D(n, k)`:

```

Return countBits(n, k, 31)
EndAlgorithm

```


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

Concurs Mate-Info – 12 aprilie 2025
Proba scrisă la INFORMATICĂ

BAREM ȘI REZOLVARE

OFICIU: 10 puncte

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