

Concurs Mate-Info – 20 aprilie 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.
- Dacă pe un același rând apar mai multe instrucțiuni de atribuire consecutive, acestea sunt delimitate prin ";".

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

```
Algorithm calcul(v, n):  
  i ← 1; j ← n  
  While i < j execute  
    While i < j AND v[i] MOD 2 = 1 execute  
      i ← i + 1  
    EndWhile  
    While i < j AND v[j] MOD 2 = 1 execute  
      j ← j - 1  
    EndWhile  
    If v[i] ≠ v[j] then  
      Return False  
    EndIf  
    i ← i + 1  
    j ← j - 1  
  EndWhile  
  Return True  
EndAlgorithm
```

Pentru care din următoarele situații algoritmul returnează *True*?

- Dacă vectorul v este format din valorile $[1, 11, 2, 4, 3, 4, 7, 6, 4, 21, 23, 25, 2]$ și $n = 13$
- Dacă vectorul v este format din valorile $[1, 11, 2, 4, 3, 7, 6, 4, 21, 23, 25, 2]$ și $n = 12$
- Dacă și numai dacă valoarea absolută a diferenței dintre două elemente pare ale vectorului v între care există cel puțin un element impar, este egală cu 2
- Dacă vectorul format din elementele pare ale vectorului v parcurs de la stânga la dreapta este egal cu vectorul format din elementele pare ale vectorului v parcurs de la dreapta la stânga

2. Se consideră algoritmul $g(a, b)$ unde a și b sunt numere naturale ($0 \leq a, b \leq 10^4$):

```
Algorithm g(a, b):  
  If a = b then  
    Return a  
  EndIf  
  If a > b then  
    Return g(a - b, b)  
  Else  
    Return g(a, b - a)  
  EndIf  
EndAlgorithm
```

Care dintre următoarele afirmații sunt adevărate?

- Pentru apelul $g(2, 2)$ algoritmul returnează 2.
- Dacă $a = b$, algoritmul nu se autoapelează niciodată.
- Dacă $a = 0$ și $0 \leq b \leq 10^4$, algoritmul se autoapelează o singură dată.
- Dacă $a \neq 0$, $b \neq 0$ și $a \neq b$, algoritmul se autoapelează de $a + b - 1$ ori.

3. Un graf orientat are 8 vârfuri, numerotate de la 1 la 8, și arcele $(1, 7), (1, 8), (3, 5), (3, 7), (4, 3), (4, 7), (6, 3), (6, 5), (6, 7), (6, 8), (8, 5), (8, 7)$. Numărul vârfurilor care au gradul extern nul este:

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

4. Care este valoarea expresiei $\text{NOT}((x \text{ MOD } 2 = 0) \text{ AND } (\text{NOT}((y > x) \text{ AND } (x \text{ MOD } 7 \neq 5))))$, dacă $x = 12$ și $y = 23$?

- True*
- False*
- Aceeași valoare ca a expresiei $\text{NOT}((x \text{ MOD } 2 = 0) \text{ AND } (\text{NOT}((x > y) \text{ AND } (x \text{ MOD } 7 \neq 5))))$
- Aceeași valoare ca a expresiei $\text{NOT}((y \text{ MOD } 2 = 0) \text{ AND } (\text{NOT}((x > y) \text{ AND } (y \text{ MOD } 7 \neq 5))))$

5. Se consideră algoritmul $ghici(n)$, unde n este număr natural ($1 \leq n \leq 10^9$):

```
Algorithm ghici(n):
  f ← 0; y ← -1
  For c ← 0, 9 execute
    x ← n
    k ← 0
    While x > 0 execute
      If x MOD 10 = c then
        k ← k + 1
      EndIf
      x ← x DIV 10
      If k > f then
        f ← k
        y ← c
      EndIf
    EndWhile
  EndFor
  Return y
EndAlgorithm
```

Precizați ce returnează algoritmul:

- A. Numărul de cifre al numărului n
- B. Frecvența maximă a frecvențelor cifrelor din numărul n
- C. Una dintre cifrele cu frecvența maximă din numărul n
- D. Una dintre cifrele cu valoare maximă din numărul n

6. Se consideră algoritmul $divizori(n)$, unde n este număr întreg ($-10^3 \leq n \leq 10^3$).

```
Algorithm divizori(n):
  nr ← 0; d ← 1
  While d * d ≤ n execute
    If n MOD d = 0 then
      nr ← nr + 1
    EndIf
    d ← d + 1
  EndWhile
  Return 2 * nr
EndAlgorithm
```

Care dintre următoarele afirmații sunt adevărate?

- A. Dacă $n = 5$, algoritmul returnează 2.
- B. Dacă $n > 1$, algoritmul returnează numărul tuturor divizorilor (proprii și improprii) ai numărului n .
- C. Dacă $n = 0$, algoritmul returnează 0.
- D. Dacă $n < 0$, algoritmul returnează numărul tuturor divizorilor (proprii și improprii) corespunzător valorii absolute a lui n .

7. Se consideră algoritmul $ceReturneaza(a, b)$, unde a și b sunt numere naturale ($0 \leq a, b \leq 10^3$):

```
Algorithm ceReturneaza(a, b):
  If a > b then
    c ← a; a ← b; b ← c
  EndIf
  d ← 0
  For i ← a, b execute
    If i MOD 2 = 0 then
      d ← d + 1
    EndIf
  EndFor
  Return d
EndAlgorithm
```

În care din următoarele situații rezultatul returnat este 0?

- A. $a = 11, b = 11$
- B. $a = 4, b = 8$
- C. $a = 12, b = 12$
- D. $a = 0, b = 0$

8. Se consideră algoritmul $ceFace(n)$, unde n este număr natural ($1 \leq n \leq 10^4$):

```
Algorithm ceFace(n):
  k ← 0; s ← 0
  While k ≠ n execute
    k ← k + 1
    s ← s + 2 * k - 1
    Write s, " "
  EndWhile
EndAlgorithm
```

Care dintre următoarele afirmații sunt adevărate?

- A. Pentru $n = 3$, algoritmul va afișa: 0 9
- B. Pentru $n = 10$, penultima valoare atribuită variabilei s în timpul executării este 81
- C. Algoritmul afișează pătratele numerelor naturale $1, 2, \dots, n$
- D. Pentru $n = 4$, algoritmul va afișa: 1 4 8 16

9. Se consideră algoritmi verificare_aux(a, b) și verificare(a, b), unde a și b sunt numere naturale ($1 \leq a, b \leq 10^9$):

```

Algorithm verificare_aux(a, b):
  c ← b
  While a > 0 execute
    While (c > 0) AND (a MOD 10 ≠ c MOD 10) execute
      c ← c DIV 10
    EndWhile
    If c = 0 then
      Return False
    EndIf
    c ← b
    a ← a DIV 10
  EndWhile
  Return True
EndAlgorithm

```

```

Algorithm verificare(a, b):
  Return verificare_aux(a, b) AND verificare_aux(b, a)
EndAlgorithm

```

Pentru care dintre condițiile următoare algoritmul verificare(a, b) returnează *True*?

- A. Dacă a și b au același număr de cifre.
- B. Dacă $a = 1001$ și $b = 10$.
- C. Dacă vectorul de frecvență a cifrelor lui a este identic cu vectorul de frecvență a cifrelor lui b .
- D. Dacă $a = 123$ și $b = 321$.

10. Se consideră algoritmul verifica(n), unde n este număr natural ($1 \leq n \leq 10^4$).

```

Algorithm verifica(n):
  a ← n MOD 10
  n ← n DIV 10
  While n > 0 execute
    b ← n MOD 10
    If a ≤ b then
      Return False
    EndIf
    a ← b
    n ← n DIV 10
  EndWhile
  Return True
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- A. În urma apelului verifica(2024) algoritmul returnează *False*.
- B. Algoritmul returnează *True* dacă și numai dacă n este un număr în care cifrele sunt în ordine strict crescătoare.
- C. Algoritmul returnează *True* dacă și numai dacă n este un număr în care cifrele sunt în ordine strict descrescătoare.
- D. Algoritmul returnează *True* dacă și numai dacă cifra cea mai semnificativă a numărului n este mai mică decât cifra sa cea mai nesemnificativă.

11. Se consideră algoritmul $F(x, n, i, S, k)$, unde x este un vector de n ($1 \leq n \leq 10^4$) numere întregi ($x[1], x[2], \dots, x[n]$, $-10^3 \leq x[i] \leq 10^3$ pentru $i = 1, 2, \dots, n$), S este număr real, iar i și k sunt numere naturale. Operatorul „/” reprezintă împărțirea reală, de exemplu: $3 / 2 = 1.5$.

```

Algorithm F(x, n, i, S, k):
  If n < i then
    If k = n then
      Return 0
    Else
      Return S / (n - k)
    EndIf
  Else
    If x[i] MOD 2 = 0 then
      Return F(x, n, i + 1, S + x[i], k)
    Else
      Return F(x, n, i + 1, S, k + 1)
    EndIf
  EndIf
EndAlgorithm

```

Știind că algoritmul se apelează în forma $F(x, n, 1, 0.0, 0)$, precizați care dintre următoarele afirmații sunt adevărate?

- A. Algoritmul returnează suma numerelor pare din vectorul x , împărțită la numărul numerelor impare din vector.
- B. Dacă $n = 10$ și $x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$, algoritmul returnează valoarea 6.0.
- C. Algoritmul returnează media aritmetică a numerelor pare din vectorul x .
- D. Dacă $n = 10$ și $x = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]$, algoritmul returnează valoarea 0.

12. Se consideră o matrice pătratică x de dimensiunea n cu elemente numere naturale distincte ($2 \leq n \leq 50$, $x[1][1], \dots, x[1][n], x[2][1], \dots, x[2][n], \dots, x[n][1], \dots, x[n][n]$, $1 \leq x[i][j] \leq 10^4$, pentru $i = 1, 2, \dots, n, j = 1, 2, \dots, n$). Elementele fiecărei linii și elementele fiecărei coloane sunt ordonate crescător. Algoritmul caută(n, x, v) caută o valoare v în matricea x și returnează perechea formată din indicele de linie și indicele de coloană a poziției pe care se află valoarea v în matrice sau (-1, -1) dacă valoarea v nu se află printre elementele matricei. Presupunem că algoritmul cautareBinara(t, n, v) implementează algoritmul căutării binare pentru a determina dacă un număr v este prezent în vectorul t cu n elemente ordonate crescător. Dacă v nu se află în a i -a linie a matricei, în urma apelului cautareBinara($x[i], n, v$) se returnează -1.

Care dintre următorii algoritmi sunt cei mai eficienți din punct de vedere al complexității timp și realizează cerințele descrise?

A.

```

Algorithm cauta(n, x, v):
  a ← -1
  b ← -1
  For i ← 1, n execute
    For j ← 1, n execute
      If x[i][j] = v then
        a ← i
        b ← j
      EndIf
    EndFor
  EndFor
  Return a, b
EndAlgorithm

```

C.

```

Algorithm cauta(n, x, v):
  a ← -1
  b ← -1
  i ← 1; j ← n
  While i ≤ n AND j > 0 execute
    If x[i][j] = v then
      a ← i
      b ← j
    EndIf
    If x[i][j] > v then
      j ← j - 1
    Else
      i ← i + 1
    EndIf
  EndWhile
  Return a, b
EndAlgorithm

```

B.

```

Algorithm cauta(n, x, v):
  a ← -1
  b ← -1
  For i ← 1, n execute
    j ← cautareBinara(x[i], n, v)
    If j ≠ -1 then
      a ← i
      b ← j
    EndIf
  EndFor
  Return a, b
EndAlgorithm

```

D.

```

Algorithm cauta(n, x, v):
  a ← -1
  b ← -1
  i ← 1; j ← 1
  While i ≤ n AND x[i][j] < v execute
    i ← i + 1
  EndWhile
  While j ≤ n AND x[i][j] < v execute
    j ← j + 1
  EndWhile
  If x[i][j] = v then
    a ← i
    b ← j
  EndIf
  Return a, b
EndAlgorithm

```

13. Fiind dată o matrice pătratică M de 3×3 elemente, care dintre următoarele secvențe de cod implementează corect o rotire cu 90 de grade în sens trigonometric a matricei în jurul elementului de pe poziția (2, 2)?

A.

```

For i ← 0, 1 execute
  X ← M[1][1]
  M[1][1] ← M[1][2]
  M[1][2] ← M[1][3]
  M[1][3] ← M[2][3]
  M[2][3] ← M[3][3]
  M[3][3] ← M[3][2]
  M[3][2] ← M[3][1]
  M[3][1] ← M[2][1]
  M[2][1] ← X
EndFor

```

C.

```

For i ← 1, 2 execute
  X ← M[1][1]
  M[1][1] ← M[1][2]
  M[1][2] ← M[1][3]
  M[1][3] ← M[2][3]
  M[2][3] ← M[3][3]
  M[3][3] ← M[3][2]
  M[3][2] ← M[3][1]
  M[3][1] ← M[2][1]
  M[2][1] ← X
EndFor

```

B.

```

For i ← 0, 2 execute
  X ← M[1][1]
  M[1][1] ← M[1][2]
  M[1][2] ← M[1][3]
  M[1][3] ← M[2][3]
  M[2][3] ← M[3][3]
  M[3][3] ← M[3][2]
  M[3][2] ← M[3][1]
  M[3][1] ← M[2][1]
  M[2][1] ← X
EndFor

```

D.

```

For i ← 1, 3 execute
  X ← M[1][1]
  M[1][1] ← M[1][i]
  M[1][i] ← M[1][3]
  M[1][3] ← M[i][3]
  M[i][3] ← M[3][3]
  M[3][3] ← M[3][i]
  M[3][i] ← M[3][1]
  M[3][1] ← M[i][1]
  M[i][1] ← X
EndFor

```

14. Se consideră algoritmul `rearanjeaza(x, n)`, unde n este număr natural ($1 \leq n \leq 200$), iar x este un vector de n numere întregi distincte ($x[1], x[2], \dots, x[n]$, $-100 \leq x[i] \leq 100$, pentru $i = 1, 2, \dots, n$). Algoritmul `interschimba(x, i, j)` interschimbă elementele $x[i]$ și $x[j]$.

```

Algorithm rearanjeaza(x, n):
  v ← x[n]
  i ← 0; j ← 1
  While j ≤ n - 1 execute
    If x[j] ≤ v then
      i ← i + 1
      interschimba(x, i, j)
    EndIf
    j ← j + 1
  EndWhile
  i ← i + 1
  interschimba(x, i, n)
  Return i
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- A. Algoritmul sortează în ordine crescătoare elementele vectorului x .
- B. Dacă vectorul x este sortat crescător, nu se va modifica ordinea elementelor din vector.
- C. Vectorul x va fi rearanjat astfel încât ultimul element din vectorul inițial va avea în stânga sa doar elemente cu valoare mai mică și în dreapta doar elemente cu valoare mai mare.
- D. Algoritmul returnează indicele inițial al elementului cu valoarea minimă din vectorul x .

15. Se consideră algoritmul `calcul(v, n)`, unde n este număr natural ($1 \leq n \leq 10^4$), iar v este un vector cu n elemente numere naturale ($v[1], v[2], \dots, v[n]$, $1 \leq v[i] \leq 200$, pentru $i = 1, 2, \dots, n$):

```

Algorithm calcul(v, n):
  If n = 1 then
    Return v[1]
  EndIf
  If v[1] MOD v[n] = 0 then
    v[1] ← v[n]
    n ← n - 1
    Return calcul(v, n)
  Else
    aux ← v[n]
    v[n] ← v[1] MOD v[n]
    v[1] ← aux
    Return calcul(v, n)
  EndIf
EndAlgorithm

```

Pentru care din următoarele valori algoritmul returnează valoarea 12?

- A. $v = [60, 96, 120, 84]$, $n = 4$
- B. $v = [75, 24, 12, 84]$, $n = 4$
- C. $v = [75, 24, 49, 80]$, $n = 4$
- D. $v = [60, 24, 12, 84]$, $n = 4$

16. Se consideră algoritmul `ceFace(n)`, unde n este număr întreg ($-10^4 \leq n \leq 10^4$):

```

Algorithm ceFace(n):
  If n = 0 then
    Return "0"
  EndIf
  If n < 0 then
    Return "-" + ceFace(-n)
  EndIf
  If n MOD 3 = 0 then
    Return ceFace(n DIV 3) + "0"
  EndIf
  If n MOD 3 = 1 then
    Return ceFace(n DIV 3) + "1"
  EndIf
  Return ceFace(n DIV 3) + "2"
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- A. Dacă numărul n este o putere a lui 3, șirul de caractere returnat conține un singur caracter "1".
- B. Pentru $n = 3$ și $n = -3$ algoritmul `ceFace(n)` returnează valori identice.
- C. Dacă $n = 82$, algoritmul returnează "010001".
- D. Dacă n este număr negativ, algoritmul intră în ciclul infinit.

17. Se consideră algoritmul $\text{decide}(n, x)$, unde n este număr natural ($1 \leq n \leq 10^4$), iar x este un vector cu n elemente numere naturale ($x[1], x[2], \dots, x[n]$, $0 \leq x[i] \leq 100$, pentru $i = 1, 2, \dots, n$).

Care dintre următoarele afirmații sunt adevărate?

- A. Dacă $n = 5$ și $x = [1, 2, 1, 3, 1]$ algoritmul returnează 1.
- B. Dacă $n = 5$ și $x = [1, 2, 2, 3, 1]$ algoritmul returnează -1.
- C. Pentru orice vector de intrare algoritmul returnează -1.
- D. Algoritmul returnează primul element al vectorului x .

```

Algorithm decide(n, x):
  a ← x[1]
  i ← 2; j ← 1
  While i ≤ n execute
    If x[i] = a then
      j ← j + 1
    Else
      If j > 0 then
        j ← j - 1
      Else
        a ← x[i]
        j ← 1
      EndIf
    EndIf
    i ← i + 1
  EndWhile
  i ← 1; j ← 0
  While i ≤ n execute
    If x[i] = a then
      j ← j + 1
    EndIf
    i ← i + 1
  EndWhile
  If j > (n DIV 2) then
    Return a
  Else
    Return -1
  EndIf
EndAlgorithm

```

18. Se consideră algoritmul $\text{ceFace}(n)$, în cadrul căruia se vor citi n numere, unde n este număr natural ($1 \leq n \leq 10^9$):

```

Algorithm ceFace(n):
  nr ← 0
  Read a
  For i ← 2, n execute
    Read b
    If a ≠ b then
      nr ← nr + 1
    EndIf
    a ← b
  EndFor
  Return nr
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- A. Algoritmul returnează numărul numerelor care se repetă printre numerele citite (de exemplu, dacă numerele sunt 3, 34, 34, 7, 3, 34, atunci valoarea returnată este 2).
- B. Algoritmul returnează lungimea celei mai lungi subsecvențe de numere citite ce au valori egale (de exemplu, dacă numerele sunt 2, 34, 34, 34, 5, atunci valoarea returnată este 3).
- C. Algoritmul returnează numărul perechilor de elemente consecutive cu valori diferite printre numerele citite (de exemplu, dacă numerele sunt 2, 34, 34, 7, atunci (2, 34), (34, 7) sunt perechi de elemente consecutive cu valori diferite și se returnează 2).
- D. Algoritmul returnează numărul perechilor de elemente consecutive cu valori egale printre numerele citite (de exemplu, dacă numerele sunt 2, 2, 3, 3, atunci (2, 2), (3, 3) sunt perechi de elemente consecutive cu valori egale și se returnează 2).

19. Se consideră algoritmul $f(a)$, unde a este un număr natural ($0 \leq a \leq 10^4$):

```

Algorithm f(a):
  n ← 0
  While a > 1 execute
    b ← 1
    While b ≤ a execute
      b ← 3 * b
      n ← n + 1
    EndWhile
    a ← a DIV 3
  EndWhile
  Return n
EndAlgorithm

```

Care este valoarea returnată de algoritm dacă se apelează pentru $a = 81$?

- A. 0
- B. 14
- C. 16
- D. 9

20. Se consideră algoritmul $h(n, a)$, unde n este număr natural ($1 \leq n \leq 10^3$), iar a este un vector cu n elemente numere întregi ($a[1], a[2], \dots, a[n]$, $-10^4 \leq a[i] \leq 10^4$, pentru $i = 1, 2, \dots, n$) ordonate crescător.

Care dintre următoarele apeluri va returna valoarea 4?

- A. $h(5, [1, 2, 3, 4, 5])$
- B. $h(6, [2, 4, 6, 10, 18, 20])$
- C. $h(7, [2, 2, 3, 4, 6, 9, 13])$
- D. $h(5, [2, 2, 2, 4, 6])$

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

Ce valoare va returna algoritmul, dacă apelul are forma $f(x, 9, 41)$, unde $x = [41, 15, 5, 8, 10, 1, 16, 18, 19]$?

- A. 1
- B. 3
- C. 5
- D. 7

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

Care dintre următoarele afirmații sunt adevărate?

- A. În cazul apelului $select([0, 1, 2, 4, 5, 8, 9, 10, 7, 6], 10, 10)$, algoritmul returnează 10.
- B. Algoritmul returnează poziția pe care apare elementul x în vectorul v dacă și numai dacă vectorul v este sortat crescător.
- C. Complexitatea algoritmului este $O(\log_2 n)$.
- D. În cazul apelului $select([0, 1, 2, 4, 5, 8, 9, 10, 7, 6], 7, 10)$, algoritmul returnează -1.

```

Algorithm h(n, a):
  t ← 0; i ← n
  While i > 2 execute
    k ← 1
    j ← i - 1
    b ← a[i]
    While k < j execute
      If a[k] + a[j] = b then
        t ← t + 1
        k ← k + 1
        j ← j - 1
      Else
        If a[k] + a[j] < b then
          k ← k + 1
        Else
          j ← j - 1
        EndIf
      EndIf
    EndWhile
    i ← i - 1
  EndWhile
  Return t
EndAlgorithm

```

```

Algorithm f(x, n, m):
  If m = 0 then
    Return 1
  EndIf
  If n = 0 then
    Return 0
  EndIf
  If x[n] > m then
    Return f(x, n - 1, m)
  Else
    Return f(x, n - 1, m) + f(x, n - 1, m - x[n])
  EndIf
EndAlgorithm

```

```

Algorithm select(v, x, n):
  i ← 1; j ← n
  While i ≤ j execute
    k ← (i + j) DIV 2
    If v[k] = x then
      Return k
    EndIf
    If v[i] ≤ v[k] then
      If v[i] ≤ x AND x < v[k] then
        j ← k - 1
      Else
        i ← k + 1
      EndIf
    Else
      If v[k] < x AND x ≤ v[j] then
        i ← k + 1
      Else
        j ← k - 1
      EndIf
    EndIf
  EndWhile
  Return -1
EndAlgorithm

```

23. Se consideră algoritmul $\text{maiMare}(n)$ unde n este număr natural nenul ($1 \leq n < 10^6$) alcătuit din cifre distincte.

Algoritmul ar trebui să returneze numărul numerelor strict mai mari ca n , formate din cifrele lui n .

De exemplu, $\text{maiMare}(213) = 3$.

Presupunem că n nu are cifre 0 la început și că avem următorii algoritmi implementați conform specificațiilor:

- $\text{factorial}(n)$ – returnează factorialul numărului natural n ($1 \leq n \leq 10$)
- $\text{nrCifre}(n)$ – returnează numărul cifrelor numărului natural n ($1 \leq n < 10^6$)
- $\text{imparte}(n)$ – returnează un vector având ca elemente cifrele numărului natural n ($1 \leq n < 10^6$), în ordine inversă. De exemplu: $\text{imparte}(1352)$ returnează vectorul $[2, 5, 3, 1]$.

```
Algorithm maiMare(n):
    cifre ← imparte(n)
    nrCif ← nrCifre(n)
    Return calculeaza(cifre, nrCif)
EndAlgorithm
```

```
1. Algorithm calculeaza(v, n):
2.   If n < 2 then
3.     Return 0
4.   EndIf
5.   mm ← 0
6.   For i ← 1, n - 1 execute
7.     If v[i] > v[n] then
8.       mm ← mm + 1
9.     EndIf
10.  EndFor
11.  ...
12. EndAlgorithm
```

Care dintre următoarele instrucțiuni trebuie scrisă la linia 11 a algoritmului $\text{calculeaza}(v, n)$?

- A. Return $\text{factorial}(n) - ((n - \text{mm} - 1) * \text{factorial}(n - 1) + \text{calculeaza}(v, n - 1))$
 B. Return $\text{calculeaza}(v, n - 1) * \text{mm} + \text{factorial}(n - 1)$
 C. Return $(\text{mm} * \text{factorial}(n) + \text{calculeaza}(v, n - 1)) \text{ DIV } n$
 D. Return $\text{calculeaza}(v, n - 1) + \text{mm} * \text{factorial}(n - 1)$

24. Un eveniment trebuia să aibă loc în sala I, dar trebuie mutat în sala II, unde numerotarea scaunelor diferă. În ambele săli există L rânduri de scaune ($2 \leq L \leq 50$), fiecare rând fiind împărțit la mijloc de un culoar și având K scaune ($2 \leq K \leq 50$) în fiecare parte a culoarului (deci, sala conține în total $2 * K * L$ scaune). În sala II fiecare loc este identificat printr-un singur număr. Locurile din stânga culoarului au numere pare, iar numerotarea scaunelor începe pe rândul din fața scenei. Deci scaunele din primul rând au numerele (pornind dinspre culoar spre marginea sălii) 2, 4, 6 etc. După ce toate scaunele de pe un rând au fost numerotate, pe rândul următor se continuă numerotarea, reîncepând cu scaunul de lângă culoar cu următorul număr par. Locurile din partea dreaptă a culoarului sunt numerotate la fel, dar folosind numere impare. Deci scaunele din primul rând au numerele 1, 3, 5, etc, pornind dinspre culoar spre marginea sălii.

În sala I fiecare loc este identificat prin trei valori. Numărul rândului (o valoare între 1 și L inclusiv, rândul 1 fiind cel din fața scenei), direcția locului față de culoar (valoarea "stanga" sau "dreapta") și numărul scaunului în cadrul rândului (o valoare între 1 și K inclusiv, scaunul 1 fiind cel de lângă culoar). Din cauza mutării spectacolului, locurile de pe bilete din sala I (reprezentate prin *rand*, *loc*, *directie*) trebuie transformate în locuri valabile în sala II (un singur număr).

Care dintre algoritmi de mai jos, având ca date de intrare K , *rand*, *loc*, *directie* conform enunțului, execută în mod corect transformarea (o transformare este corectă dacă fiecare spectator va avea un loc unic în sala II)?

A.

```
Algorithm transforma(K, rand, loc, directie):
  If directie = "stanga" then
    rez ← 2 * (loc + K * (rand - 1))
  Else
    rez ← 2 * (loc + K * (rand - 1) + 1)
  EndIf
  Return rez
EndAlgorithm
```

C.

```
Algorithm transforma(K, rand, loc, directie):
  rez ← (rand - 1) * K * 2
  rez ← rez + (loc * 2)
  If directie = "dreapta" then
    rez ← rez - 1
  EndIf
  Return rez
EndAlgorithm
```

B.

```
Algorithm transforma(K, rand, loc, directie):
  rez ← rand * (K - 1) * 2
  rez ← rez + (loc * 2)
  If directie = "dreapta" then
    rez ← rez - 1
  EndIf
  Return rez
EndAlgorithm
```

D.

```
Algorithm transforma(K, rand, loc, directie):
  rez ← (rand - 1) * K * 2
  rez ← rez + (loc * 2)
  If directie = "dreapta" then
    rez ← rez + 1
  EndIf
  Return rez
EndAlgorithm
```


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

Concurs Mate-Info – 20 aprilie 2024
Proba scrisă la INFORMATICĂ

BAREM ȘI REZOLVARE

OFICIU: 10 puncte

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