

**Simulare Examen Admitere UBB 2024 – 18-19 Mai 2024**  
**Proba scrisă la Informatică**

**NOTA IMPORTANTA**

În lipsa altor precizări:

- Presupuneți ca toate operațiile aritmetice se efectuează pe tipuri de date nelimitate (nu există overflow / underflow).
- Numerotarea indicilor tuturor sirurilor / vectorilor începe de la 1.
- Toate restricțiile se referă la valorile parametrilor actuali la momentul apelului inițial.

1. Precizați care dintre urmatoarele expresii verifică corect dacă numarul  $N$  ( $1 \leq N \leq 10^9$ ) este divizibil cu 24:
  - A.  $(N \text{ MOD } 6) = 0 \text{ AND } (N \text{ MOD } 4) = 0 \rightarrow 12$
  - B.  $(N \text{ MOD } 4 = 0) \text{ AND } ((N \text{ DIV } 4) \text{ MOD } 2 = 0) \text{ AND } (N \text{ MOD } 6) = 0$
  - C.  $(N \text{ MOD } 8) = 0 \text{ AND } (N \text{ MOD } 3) = 0$
  - D.  $((N \text{ DIV } 4) \text{ MOD } 2 = 0) \text{ AND } (N \text{ MOD } 6 = 0)$

2. Se consideră subalgoritmul *ce\_face* care primește ca și parametru unic de intrare un număr natural nenul  $n$  ( $1 \leq n \leq 10^9$ ):

**Subalgoritm ce\_face(n):**

```

C ← [0] * (n + 1)
C[1] ← 1
C[0] ← 1
For i ← 2, √n execute
  If C[i] = 0 then
    For j ← i, n DIV i execute
      C[i * j] ← 1
    EndFor
  EndIf
EndFor
return C[n] = 0
EndSubalgoritm
  
```

Precizați care dintre urmatoarele afirmații referitoare la subalgoritm *ce\_face* sunt adevărate:

- A. Subalgoritmul verifică dacă numarul  $n$  este număr par.
- B. Subalgoritmul verifică dacă numarul  $n$  este prim.
- C. Subalgoritmul verifică primalitatea numarului  $n$  printr-un algoritm de complexitate temp minima. → *for în for*
- D. Subalgoritmul returnează 1 pentru orice număr natural impar transmis ca și parametru.

3. Se consideră subalgoritmul *f* definit mai jos care primește ca și parametrii de intrare un sir *a* cu  $n$  elemente numere naturale nenule și un număr natural nenul  $k$  ( $1 \leq n \leq 10^6$ ,  $1 \leq a[i] \leq 10^9$ ,  $1 \leq i \leq n$ ,  $1 \leq k \leq n$ ):

**Subalgoritm f(a, n, k):**

```

maxi ← 1
For i ← 1, n - k + 1 execute
  val ← a[i]
  For j ← i + 1, i + k - 1 execute
    While a[j] ≠ 0 execute
      r ← val MOD a[j]
      val ← a[j]
      a[j] ← r
    EndWhile
  EndFor
  If val > maxi then
    maxi ← val
EndIf
EndFor
return maxi
EndSubalgoritm
  
```

Precizați care dintre urmatoarele afirmații referitoare la subalgoritm *f* definit mai sus sunt adevărate:

- A. Subalgoritmul determină secvența de lungime maximă  $k$  cu cel mai mare CMMDC al elementelor din sir.
- B. Subalgoritmul determină cel mai mare CMMDC al unei secvențe de  $k$  elemente din sir.
- C. Subalgoritmul nu calculează corect CMMDC-ul fiecarei secvențe de lungime  $k$  din sir. → *mănușa*
- D. Pentru  $k = 1$  se returnează valoarea maximă din sir.

4. Se consideră subalgoritmul *suma* definit mai jos care primește ca și parametrii de intrare un sir *a* cu  $n$  elemente numere naturale nenule, ordonat crescător și o variabilă *val*, de asemenea număr natural nenul ( $1 \leq n \leq 10^6$ ,  $1 \leq a[i] \leq 10^9$ ,  $1 \leq i \leq n$ ,  $1 \leq k \leq 10^6$ ):

**Subalgoritm suma(a, n):**

```

cnt ← 0
st ← 1
dr ← n
While st < dr execute
  If a[st] + a[dr] = val then
    cnt ← cnt + 1
    st ← st + 1
    dr ← dr - 1
  EndIf
  If a[st] + a[dr] > val then
    
```

```

dr ← dr - 1
EndIf
If a[st] + a[dr] < val then
    st ← st + 1
EndIf
EndWhile
Return cnt
EndSubalgorithm

```

Precizati care dintre urmatoarele afirmatii sunt adevarate referitoare la subalgoritm *suma(a, n)*:

- A. Subalgoritmul determina numarul de perechi de elemente, egal departe de capetele sirului, al carei suma este egala cu val.
  - B. Subalgoritm determina cate perechi de elemente exista in sir pentru care suma lor sa fie egala cu val.
  - C. Pentru un  $n$  par, se returneaza mereu o valoare par.
  - D. Niciuna dintre afirmatiile A, B sau C nu este corecta.
5. Câte noduri are un arbore cu rădăcină în care numărul de muchii este egal cu numărul de muchii ale unui graf neorientat complet cu  $n$  noduri?

- A.  $\frac{n^2-n+2}{2}$
- B.  $\frac{n \cdot (n+1)}{2}$
- C.  $\frac{n \cdot (n+1)}{2} + 1$
- D.  $\frac{n \cdot (n+1)}{2} - 1$

6. Se considera subalgoritm *afis* care primeste ca si parametrii de intrare o matrice patratica  $a$  cu  $n$  linii si  $n$  coloane si numarul natural nenul  $n$  ( $1 \leq n \leq 10^3$ ,  $1 \leq a[i][j] \leq 10^9$ ,  $1 \leq i, j \leq n$ ):

**Subalgoritm afis(a, n):**

```

cnt ← 0
i ← 1
j ← 1
While cnt < n * n execute
    While j ≤ n AND cnt < n * n execute
        Write a[i][j], ','
        j ← j + 1
        cnt ← cnt + 1
    EndWhile
    While i ≤ n AND cnt < n * n execute
        Write a[i][j-1], ','
        i ← i + 1
        cnt ← cnt + 1
    EndWhile
    i ← i - 1
    j ← j - 1
    While j ≥ 1 AND cnt < n * n execute

```

1 2 2 2  
→ 2, perechi  
nr 6  
( nr. cresc.)  
( nr. scad.)

```

Write a[i][j], ','
j ← j - 1
cnt ← cnt + 1
EndWhile
While i ≥ 1 AND cnt < n * n execute
    Write a[i][j+1], ','
    i ← i - 1
    cnt ← cnt + 1
EndWhile
i ← i + 1
j ← j + 1
EndWhile
EndSubalgorithm

```

Stiind ca subalgoritmul se apeleaza pentru o matrice generata dupa formula  $a[i][j] = 2 * i + j$  si  $n = 4$ , ce se afiseaza?

- A. 3, 4, 5, 6, 8, 10, 12, 11, 10, 9, 7, 5, 6, 7, 9, 8
- B. 3, 4, 5, 6, 8, 10, 12, 11, 10, 9, 7, 5, 3, 4, 5, 6
- C. 3, 4, 5, 6, 8, 10, 12, 11, 10, 9, 7, 5, 3
- D. 3, 4, 5, 6, 5, 6, 7, 8, 7, 8, 9, 10, 9, 10, 11, 12

7. Se considera subalgoritmul *ce\_face* care primeste ca si parametrii o matrice  $a$  cu  $n$  linii si  $n$  coloane in care elementele de pe fiecare linie luate una dupa cealalta reprezinta elementele unui vector ordonat crescator, parametru  $n$ , numar natural nenul si o valoare  $val$ , de asemenea numar natural ( $1 \leq n \leq 10^3$ ,  $1 \leq a[i][j] \leq 10^9$ ,  $1 \leq i, j \leq n$ ,  $a[1][1] \leq a[1][2] \leq \dots \leq a[1][n] \leq a[2][1] \leq a[2][2] \leq \dots \leq a[2][n] \leq a[3][1] \leq \dots$ )

**Subalgoritm ce\_face(a, n):**

```

st ← 1
dr ← n
While st ≤ dr execute
    mij ← (st + dr) DIV 2
    If a[mij][n - mij + 1] = val then
        Return 1
    EndIf
    If a[mij][n - mij + 1] > val then
        dr ← mij - 1
    Else
        st ← mij + 1
    EndIf
EndWhile
Return 0
EndSubalgorithm

```

Precizati care dintre urmatoarele afirmatii sunt adevarate cu privire la subalgoritm *ce\_face(a, n)*:

- A. Returneaza 1 daca si numai daca valoarea  $val$  apare in matrice.

- B. Algoritmul verifica daca valoarea **val** apare pe diagonala secundara din matrice.
- C. Algoritmul nu verifica corect daca valoarea **val** apare pe diagonala secundara intrucat elementele acesteia nu sunt ordonate crescator.
- D. Algoritmul returneaza mereu 0.
8. Se considera subalgoritmul **sortare** care primeste ca si parametrii de intrare un sir a cu n elemente numere naturale nenule ( $1 \leq n \leq 10^6$ ,  $1 \leq a[i] \leq 10^9$ ,  $1 \leq i \leq n$ ):

```
Subalgorithm sortare(a, n):
    For i ← 1, n - 1 execute
        For j ← i + 1, n execute
            If prim(a[i]) AND prim(a[j]) AND a[i] < a[j] then
                a[i] ← a[i] + a[j]
                a[j] ← a[i] - a[j]
                a[i] ← a[i] - a[j]
            EndIf
        EndFor
    EndFor
EndSubalgorithm
```

Care dintre urmatoarele afirmatii sunt adevarate considerand ca functia prim verifica corect daca un numar este prim?

- A. Subalgoritmul sorteaza elementele sirului in asa fel incat cele prime sa ajunga pe primele pozitii.
- B. Subalgoritmul sorteaza in ordine descrescatoare elementele prime din sir, pastrand elementele care nu sunt prime pe pozitiile lor initiale.
- C. Subalgoritmul sorteaza corect in ordine descrescatoare elementele sirului.
- D. Subalgoritmul sorteaza elementele prime in ordine crescatoare, pastrand elementele care nu sunt prime pe pozitiile lor initiale.
9. Care afirmatii sunt adevarate?
- 2122021021(3) = 78237(9)
  - 24321(5) = 5232(7)
  - 2436(9) = 111000101111(2)
  - 1010111110101001010(2) = AF54A(16)
10. Precizati care dintre urmatoarele perechi reprezinta cel mai mic si cel mai mare numar care poate fi stocat pe un tip de date intreg pentru care au fost alocati 20 de biti:
- (-1.048.576, 1.048.575)
  - (-1.048.575, 1.048.575)
  - (-524.288, 524.287)
  - (-524.287, 524.287)

11. Se considera subalgoritmul **rec** care primeste ca si parametrii de intrare un sir a cu n elemente numere naturale nenule si valoarea **n** si returneaza o valoare numar natural ( $1 \leq n \leq 10^3$ ,  $1 \leq a[i] \leq 10^9$ ,  $1 \leq i \leq n$ ).

**Subalgorithm** rec(a, n):

```
If n = 1 then
    Return a[n]
EndIf
val ← rec(a, n-1)
prod ← val * a[n]
While val != 0 then
    r ← a[n] MOD val
    a[n] ← val
    val ← r
EndWhile
Return prod DIV a[n]
EndSubalgorithm
```

$$\text{Complexitate} = \log(a + b) \\ (a, b)$$

Precizati care dintre urmatoarele afirmatii sunt adevarate referitoare la subalgoritmul rec(a, n):

- Subalgoritmul calculeaza cel mai mare divizor comun al elementelor din sir.
- Subalgoritmul are o complexitate liniara ( $O(n)$ ).
- Subalgoritmul determine cel mai mic multiplu comun al elementelor din sir.
- Subalgoritmul nu are o complexitate liniara ( $O(n)$ ).

12. Precizati pentru ce valori ale lui A, B si C, expresia urmatoare este adevarata:  
**A AND B OR !(B OR C) AND A OR B AND C**

- A ← TRUE, B ← TRUE, C ← TRUE
- A ← FALSE, B ← FALSE, C ← TRUE
- Orice valori pentru A, B, C
- Niciuna dintre variantele de raspuns de mai sus nu este corecta.

13. Se considera subalgoritmul **verificare** care primeste ca si parametrii de intrare 3 numere naturale nenule **a**, **b** si **n** ( $1 \leq a, b, n \leq 10^6$ ):

**Subalgorithm** verificare(a, b, n):

```
If a = b then
    Return a * a = n
EndIf
mij ← (a + b) DIV 2
Return verificare(a, mij, n) OR verificare(mij+1, b, n)
EndSubalgorithm
```

Stiind ca subalgoritmul este apelat sub forma **verificare(1, n, n)**, care dintre urmatoarele afirmatii sunt adevarate?

✓ verificare(1, n, n) are două jumătăți  
! nu const. lin

- A. Algoritmul verifica daca  $n$  este patrat perfect intr-o complexitate logaritmica.
- B. Algoritmul verifica daca  $n$  este patrat perfect intr-o complexitate liniara.
- C. Algoritmul cauta binar radacina numarului  $n$ .
- D. Algoritmul determina si returneaza radacina numarului  $n$ .
14. Se considera subalgoritmul *cifre* care primeste ca si parametru unic de intrare un numar natural nenul  $n$  care contine doar cifre nenule ( $1 \leq n \leq 10^6$ ).

**Subalgoritm** *cifre(n)*:

```

cnt1 ← 0
cnt2 ← 0
While n != 0 execute
    If n MOD 2 = 0 then
        cnt2 ← cnt2 + 1
    Else
        cnt1 ← cnt1 + 1
    EndIf
    n ← n DIV 10
EndWhile
If cnt1 > cnt2 then
    Return cnt1 - cnt2 < 2
Else
    Return cnt2 - cnt1 < 2
EndIf
EndSubalgoritm
```

Care dintre urmatoarele afirmatii sunt adevarate?

- A. Subalgoritmul verifica daca cifrele numarului  $n$  pot fi rearanjate in asa fel incat sa nu avem 2 cifre de aceiasi paritate alaturate.
- B. Subalgoritmul verifica daca diferența dintre numarul de cifre de pe pozitii pare si impare din numar este mai mica decat 2.
- C. Subalgoritmul verifica daca valoarea absoluta a diferenței dintre numarul de cifre pare si numarul de cifre impare ale numarului  $n$  este mai mica decat 2.
- D. Subalgoritmul returneaza diferența dintre numarul de cifre pare si numarul de cifre impare ale numarului  $n$ .
15. Se considera subalgoritmul *suma* care primeste ca si parametrii de intrare un sir *a* cu  $n$  elemente numere naturale nenule si numarul natural  $n$  ( $1 \leq n \leq 10^2$ ,  $1 \leq a[i] \leq 10^9$ ,  $1 \leq i \leq n$ ):

**Subalgoritm** *suma(a, n)*:

```

If n ≤ 0 then
    Return 0
EndIf
sum1 ← suma(a, n-1)
sum2 ← suma(a, n-2)
sum3 ← suma(a, n-3) + a[n]
```

```

If sum1 ≥ sum2 AND sum1 ≥ sum3 then
    Return sum1
Else
    If sum2 ≥ sum1 AND sum2 ≥ sum3 then
        Return sum2
    Else
        Return sum3
    EndIf
EndIf
EndSubalgoritm
```

Ce se afiseaza pentru apelul *suma([1, 3, 2, 5, 1, 7, 3, 4, 1, 2, 11, 5, 3], 13)*:

- A. 21  
B. 33  
C. 48  
D. 24

16. Scriem un cod backtracking care incercă să determine o parola despre care stim ca este formată din 4 caractere mici ale alfabetului englez și cifre. Stim că parola conține cel puțin o cifră și cel puțin o literă. Precizați care este numărul maxim de încercări în care am putea să ghicim această parola:

- A. 1.212.650  
B. 1.212.640  
C. 1.212.641  
D. 1.212.651

17. Se considera subalgoritmul *back* care primeste ca si parametrii de intrare un sir *sol*, un numar natural *poz* si un numar natural *n* ( $1 \leq \text{poz} \leq n$ ,  $1 \leq n \leq 10^2$ ):

**Subalgoritm** *back(sol, poz, n)*:  
**For** i ← 1, n **execute** → remenție form puncte fixe
 

```

For i ← 1, n execute
    If poz != i then
        sol[poz] ← i
        ok ← 1
    For j ← 1, poz - 1 execute
        If sol[j] = sol[poz] then
            ok ← 0
        EndIf
    EndFor
    If ok = 1 then
        If poz = n then
            write 1, ''
        Else
            back(sol, poz + 1, n)
        EndIf
    EndIf
EndFor
EndSubalgoritm
```

$$n! = \frac{n!}{1!} + \frac{n!}{2!} + \dots + \frac{n!}{3!} + \dots + \frac{n!}{n!}$$

Considerand de fiecare data apelul initial sub forma  $back(sol, 1, n)$ , care dintre urmatoarele afirmatii sunt adevărate?

- A. Pentru  $n = 5$ , se afiseaza pe ecran 120 de valori.
- B. Pentru  $n = 4$ , se afiseaza pe ecran 9 valori.
- C. Pentru  $n = 6$ , se afiseaza pe ecran 265 de valori.
- D. Pentru  $n = 6$ , se afiseaza pe ecran 720 de valori.

18. Se considera subalgoritmul  $f$  care primeste ca si parametrii de intrare 3 numere naturale nenule ( $1 \leq a, b, c \leq 10^6$ ).

**Subalgoritm  $f(a, b, c)$ :**

```

For i ← 1, a execute
    cnt ← 1
    For j ← 1, b DIV i execute
        cnt ← cnt + 1
    EndFor
    While cnt > 1 execute
        cnt ← cnt - 1
    EndWhile
    For k ← 1, c execute
        cnt ← cnt + 1
    EndFor
EndFor
EndSubalgoritm

```

Precizati care este complexitatea timp a acestui subalgoritm.

- A.  $O(\log(a) * b + a * c)$
- B.  $O(a * \log(b) + a * c)$
- C.  $O(\log(a) * (b + c * a / \log(a)))$
- D.  $O(a * b + a * c)$

19. Se considera subalgoritmul  $genMat$  care primeste ca si parametrii o matrice  $a$  cu  $n$  linii si  $n$  coloane si valoare  $n$  numar natural nenul ( $1 \leq n \leq 10^3$ ,  $1 \leq a[i][j] \leq 10^9$ ,  $1 \leq i, j \leq n$ ).

**Subalgoritm  $genMat(a, n)$ :**

```

For i ← 1, n execute
    For j ← 1, n execute
        If |i-j| < |n+1-i-j| then
            a[i][j] ← |i-j|
        Else
            a[i][j] ← |n+1-i-j|
        EndIf
    EndFor
EndFor
EndSubalgoritm

```

Precizati care dintre urmatoarele afirmatii sunt adevărate considerand apelul initial  $genMat(a, 5)$  considerand ca  $|x|$  este modulul lui  $x$ :

$$n = 5 \Rightarrow 5! = 5! + 12 - 4 \cdot 10 = 9$$

- A. Suma elementelor din matrice este 20.
- B. Suma elementelor din matrice este 30.
- C. Algoritmul genereaza o matrice patratica in care fiecare element este o valoare pozitiva.
- D. Algoritmul genereaza o matrice patratica in care fiecare element este egal cu distanta fata de cea mai apropiata diagonală.

20. Se considera subalgoritmul  $construire$  care primeste ca parametru unic de intrare un numar natural nenul  $n$  ( $1 \leq n \leq 10^9$ )

**Subalgoritm  $construire(n)$ :**

```

nr ← 1
d ← 2
While n > 1 execute
    While n MOD d = 0 execute
        If nr MOD d != 0 then
            nr ← nr * d
        EndIf
        n ← n DIV d
    EndWhile
    d ← d + 1
EndWhile
Return nr
EndSubalgoritm

```

Precizati care dintre urmatoarele afirmatii sunt adevărate referitoare la subalgoritmul  $construire(n)$ :

- A. Subalgoritmul returneaza valoarea lui  $n$ .
- B. Subalgoritmul are o complexitate liniara ( $O(n)$ ).  $\rightarrow$  doar  $n$  este prim
- C. Subalgoritmul determina cel mai mic numar care are aceiasi factori primi ca  $n$ .
- D. Se returneaza valoarea lui  $n$  daca si numai daca acesta este numar prim.

21. Se considera subalgoritmul  $suma$  care primeste ca si parametru unic de intrare un numar natural nenul  $n$  ( $1 \leq n \leq 10^9$ ).

**Subalgoritm  $suma(n)$ :**

```

If n = 0 then
    Return 1
ElseIf
    Return n * suma(n - 1) + n
EndSubalgoritm

```

Precizati care afirmatii sunt adevărate:

- A.  $suma(10) = 3.628.855$
- B.  $suma(n) = n! + n$
- C.  $suma(n) = n! + n * (n + 1) / 2$
- D.  $suma(4) = 88$

$$n! = 2^n + 12$$

22. Se consideră subalgoritmul *calcul* care primește ca și parametrii de intrare 2 numere naturale nenule  $n$  și  $m$  ( $1 \leq n, m \leq 10^3$ ):

```
Subalgoritm calcul(n, m):
  If n = 1 OR m = 1 then
    Return 2
  EndIf
  Return calcul(n-1, m) * 2 + calcul(n, m-1)
EndSubalgoritm
```

Precizați care dintre următoarele afirmații sunt adevărate:

- A.  $\text{calcul}(3, 3) = 34$
- B.  $\text{calcul}(2, 5) = 18$
- C.  $\text{calcul}(4, 5) = 416$
- D.  $\text{calcul}(1, 1) = 1$

23. Subalgoritmul *calcul* primește 2 numere naturale  $a$  și  $b$  ca parametri de intrare ( $1 \leq a, b \leq 10^3$ ):

```
Subalgoritm calcul(a, b):
  If b = -1 then
    Return 1
  EndIf
  Return (a - b) * calcul(a, b-1)
EndSubalgoritm
```

Ce se calculează?

- A.  $A_a^b$
- B.  $A_a^{b-1}$
- C.  $A_{a-1}^b$
- D.  $\text{calcul}(5, 3) = 120$

$$\begin{array}{c} (5-2) \\ 2 \cdot 3 \cdot 4 \cdot 5 = 5! = 120 \\ (5-3) \quad (5-4) \end{array}$$

24. Se consideră subalgoritmul *graf* care primește ca și parametrii de intrare o matrice  $a$  cu  $n$  linii și  $n$  coloane reprezentând matricea de adiacență a unui graf neorientat cu  $n$  noduri, numărul natural nenul  $n$  și un **nod** din acest graf ( $1 \leq n \leq 10^3$ ,  $0 \leq a[i][j] \leq 1$ ,  $1 \leq i, j \leq n$ ,  $1 \leq \text{nod} \leq n$ ):

```
Subalgoritm graf(a, n, nod):
  P  $\leftarrow [0]^*(n+1)$ 
  Q  $\leftarrow [0]^*(n+1) \rightarrow$  woda
  st  $\leftarrow 1$ 
  dr  $\leftarrow 1$ 
  Q[nod]  $\leftarrow$  nod  $\rightarrow$  adânci i woda
  P[nod]  $\leftarrow 1 \rightarrow$  re vizitare
  While st  $\leq$  dr execute
    nd  $\leftarrow$  Q[st]
    st  $\leftarrow$  st + 1
    For i  $\leftarrow 1, n$  execute
      If P[i] = 0 AND a[nd][i] = 1 then
        L, revizitare
```

$$P \in [0]^*(n+1) = \{0, 1\}^{n+1}$$

```
P[i]  $\leftarrow$  P[nd] + 1
dr  $\leftarrow$  dr + 1
Q[dr]  $\leftarrow$  i
EndIf
EndFor
EndWhile
Return P[n] - 1
EndSubalgoritm
```

Precizați care dintre următoarele afirmații sunt adevărate referitoare la subalgoritmul *graf*:

- A. Subalgoritmul determine căte componente conexe are graful nostru.
- B. Subalgoritmul verifică dacă graful este conex.
- C. Subalgoritmul returnează valoarea -1 dacă nu există niciun lant de la nodul **nod** către nodul **n**.
- D. Subalgoritmul returnează lungimea celui mai scurt drum de la nodul **nod** către nodul **n**.