

Simulare Examen Admitere UBB 2024 – 06-07 Aprilie 2024
Proba scrisa la Informatica

NOTA IMPORTANTA

In lipsa altor precizări:

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

1. Se considera subalgoritmul *ce_face* care primește ca si parametru unic de intrare un număr natural nenul n ($1 \leq n \leq 10^9$).

Subalgorithm *ce_face*(n):

$a \leftarrow 0$

$d \leftarrow 2$

While $d \leq n$ **execute**

$a \leftarrow a + n \text{ DIV } d$

$d \leftarrow d * 2$

EndWhile

$b \leftarrow 0$

$d \leftarrow 3$

While $d \leq n$ **execute**

$b \leftarrow b + n \text{ DIV } d$

$d \leftarrow d * 3$

EndWhile

Return min(a, b)

EndSubalgorithm

Precizați care dintre următoarele informații referitoare la subalgoritmul *ce_face* sunt adevărate:

- A. Pentru $n = 20$ se returnează valoarea 8.
 - B. Pentru $n = 20$ se returnează valoarea 18.
 - C. Mereu se returnează valoarea lui b.
 - D. Subalgoritmul calculează puterea maxima p pentru care 6^p este divizor al numărului $n!$
2. Precizați care dintre următoarele expresii verifica corect daca o variabila n , număr natural, este multiplu de 4 sau nu.
- A. $(n \text{ DIV } 100) \text{ MOD } 4 = 0$
 - B. $(n \text{ MOD } 100) \text{ MOD } 4 = 0$
 - C. $((n \text{ MOD } 10) \text{ MOD } 4 = 0 \text{ AND } (n \text{ DIV } 10) \text{ MOD } 2 = 0) \text{ OR } (n \text{ MOD } 2 = 0 \text{ AND } (n \text{ DIV } 10) \text{ MOD } 2 = 1)$
 - D. $((n \text{ MOD } 10) \text{ MOD } 4 = 0 \text{ AND } (n \text{ DIV } 10) \text{ MOD } 2 = 0) \text{ OR } ((n \text{ MOD } 10) \text{ MOD } 4 \neq 0 \text{ AND } n \text{ MOD } 2 = 0 \text{ AND } (n \text{ DIV } 10) \text{ MOD } 2 = 1)$

3. Se considera algoritmul *ce_face* care primește ca parametru unic de intrare un număr natural nenul n ($1 \leq n \leq 10^9$).

Subalgorithm *ce_face*(n):

```
a ← 0
b ← 0
While  $n > 0$  execute
    If  $n \bmod 2 = 0$  then
         $a \leftarrow a * 10 + n \bmod 10$ 
    EndIf
     $n \leftarrow n \text{ DIV } 10$ 
EndWhile
While  $a > 0$  execute
     $b \leftarrow b * 10 + a \bmod 10$ 
     $a \leftarrow a \text{ DIV } 10$ 
EndWhile
Return b
```

EndSubalgorithm

Precizați care dintre următoarele afirmații referitoare la subprogramul *ce_face* sunt adevărate.

- A. Subprogramul obține numărul n în urma eliminării tuturor cifrelor sale impare.
 - B. În cazul în care numărul n conține doar cifre pare, subalgoritmul va returna valoarea numărului n .
 - C. Subprogramul obține numărul n în urma eliminării tuturor cifrelor sale impare și a cifrelor egale cu 0.
 - D. Nicio afirmație nu este corectă.
4. Se considera subalgoritmul *prelucrare* care primește ca și parametrii de intrare un șir a cu n elemente numere naturale ($1 \leq n \leq 1000$, $1 \leq a[i] \leq 10^9$, $1 \leq i \leq n$).

Subalgorithm *prelucrare*(a , n):

```
For  $i \leftarrow 1, n$  execute
    If  $a[i] \bmod 2 = 0$  then
        For  $j \leftarrow i + 1, n$  execute
             $a[j-1] \leftarrow a[j]$ 
        EndFor
         $n \leftarrow n - 1$ 
    EndIf
```

EndFor

EndSubalgorithm

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

- A. Subalgoritmul produce modificarea șirului a prin eliminarea tuturor numerelor pare care apar în șir.
- B. Subalgoritmul produce modificarea șirului a prin eliminarea tuturor numerelor impare care apar în șir.

- C. Subalgoritmul nu produce întotdeauna ștergerea tuturor elementelor pare din sir.
- D. Dacă primul element din sir este par, niciun element din sir nu-si va păstra poziția inițială.
5. Se considera subprogramul *calcul* definit mai jos care primește ca si parametru de intrare un număr natural nenul x ($1 \leq x \leq 1000$).

Subalgorithm *calcul*(x):

If $x = 0$ **then**

Return 1

EndIf

If $x \bmod 2 = 1$ **then**

Return $2 * \text{calcul}(x - 1)$

Else

$y \leftarrow \text{calcul}(x \text{ DIV } 2)$

Return $y * y$

EndIf

EndSubalgorithm

Precizați care dintre următoarele afirmații sunt false:

- A. Pentru $x = 7$ se returnează valoarea 14.
- B. Pentru $x = 12$ se returnează valoarea 4096.
- C. Subalgoritmul calculează valoarea 2^x .
- D. Subalgoritmul calculează 2^x pentru orice x impar.
6. Se considera subprogramul recursiv *afis* care primește ca si parametru de intrare un număr natural nenul n ($1 \leq n \leq 1000$).

Subalgorithm *afis*(n):

If $n > 0$ **then**

afis($n - 1$)

write n

afis($n - 1$)

EndIf

EndSubalgorithm

Precizați care dintre următoarele afirmații sunt adevărate:

- A. Numărul total de apeluri care se efectuează pentru $n = 7$ la apelul inițial este 2^7 .
- B. Numărul total de apeluri care se efectuează pentru $n = 7$ la apelul inițial este $2^8 - 1$.
- C. Pentru $n = 3$ se afișează "1 2 3 2 1".
- D. Nicio afirmație dintre A, B si C nu este adevărată.

7. Se considera subprogramul recursiv *suma* care primește ca si parametru unic de intrare un număr natural nenul n ($1 \leq n \leq 1000$).

```

Subalgorithm suma(n):
    If  $n = 0$  then
        Return 0
    EndIf
    Return suma(n-1) + n + suma(n-1)
EndSubalgorithm

```

Pentru care dintre următoarele valori ale lui n se afișează ceea ce este precizat?

- A. Pentru $n = 7$ se afișează valoarea 28.
 - B. Pentru $n = 10$ se afișează 2036.
 - C. $\text{Suma}(n)$ este egal cu $\sum_{i=1}^n i * 2^{n-i}$ oricare ar fi n conform intervalului de definiție.
 - D. $\text{Suma}(n)$ este egal cu $\sum_{i=1}^n i * 2^i$ oricare ar fi n conform intervalului de definiție.
8. Se considera subprogramul *f* care primește ca si parametru unic de intrare un număr natural nenul n ($1 \leq n \leq 10^9$).

```

Subalgorithm f(n):
     $F \leftarrow [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ 
    While  $n > 0$  execute
         $F[n \bmod 10] \leftarrow F[n \bmod 10] + 1$ 
         $n \leftarrow n \text{ DIV } 10$ 
    EndWhile
    For  $i \leftarrow 0, 9$  execute
        While  $F[i] \neq 0$  execute
             $n \leftarrow n * 10 + i$ 
             $F[i] \leftarrow F[i] - 1$ 
        EndWhile
    EndFor
    Return n
EndSubalgorithm

```

Precizați care dintre următoarele afirmații sunt adevărate referitoare la secvența de cod de mai sus:

- A. Subprogramul determina cel mai mare număr care se poate obține din numărul n prin rearanjarea cifrelor sale.
- B. Subprogramul determina cel mai mic număr cu număr egal de cifre cu n care se poate obține din numărul n prin rearanjarea cifrelor sale.
- C. Subprogramul determina cel mai mare număr cu număr egal de cifre cu n care se poate obține din numărul n prin rearanjarea cifrelor sale.
- D. Subprogramul determina cel mai mic număr care se poate obține din rearanjarea cifrelor numărului n .

9. Se considera subprogramul *gasit* care are ca parametru unic de intrare un număr natural nenul n ($0 \leq n \leq 10^9$).

Subalgorithm *gasit*(n):

```

     $x \leftarrow n$ 
     $y \leftarrow 0$ 
    While  $x \geq y$  execute
         $z \leftarrow ((x + y) \text{ DIV } 2) * ((x + y) \text{ DIV } 2)$ 
        If  $z * z = n$  then
            Return 1
        Else
            If  $z * z < n$  then
                 $y \leftarrow (x + y) \text{ DIV } 2 + 1$ 
            Else
                 $x \leftarrow (x + y) \text{ DIV } 2 - 1$ 
            EndIf
        EndIf
    EndWhile
    Return 0
EndSubalgorithm

```

Pentru cate valori n din intervalul $(0, 10.000)$ subalgorithmul *gasit*(n) returnează valoarea 1?

- A. 9
 - B. 10
 - C. 11
 - D. 99
10. Se considera subalgorithmul *exp* care primește ca si parametrii de intrare 2 numere naturale nenul a si b ($1 \leq a, b \leq 1000$).

Subalgorithm *exp*(a, b):

```

     $p \leftarrow 1$ 
    While  $b \neq 0$  execute
        If  $b \text{ MOD } 2 = 1$  then
             $p \leftarrow (p * a) \text{ MOD } 10$ 
        EndIf
         $a \leftarrow (a * a) \text{ MOD } 10$ 
         $b \leftarrow b \text{ DIV } 2$ 
    EndWhile
    Return  $p$ 
EndSubalgorithm

```

Care dintre următoarele afirmații sunt adevărate?

- A. $\text{exp}(122, 134) = 6$
- B. $\text{exp}(342, 124) = 6$
- C. $\text{exp}(1843, 37) = 3$
- D. $\text{exp}(3721, 382) = 1$

11. Se considera subprogramul *prim* care verifica primalitatea unui număr natural n si subprogramul *modificare* care primește ca si parametrii de intrare un sir cu n elemente numere naturale nenule ($1 \leq n \leq 1000$, $1 \leq a[i] \leq 10^9$, $1 \leq i \leq 1000$).

Subalgorithm *modificare*(a, n):

```
p ← 1
For i ← 1, n execute
    If prim(a[i]) then
        a[i] ↔ a[p]
        p ← p + 1
    EndIf
```

EndIf

EndFor

n ← p - 1

EndSubalgorithm

Precizați care este efectul subalgoritmului *modificare*(a, n):

- A. Subalgoritmul elimina toate elementele prime din sir.
 - B. Subalgoritmul modifica șirul a in așa fel încât acesta sa conțină toate elementele prime pe primele poziții si in continuarea acestora sa fie toate elementele care nu erau prime.
 - C. Subalgoritmul elimina toate elementele care nu sunt prime din sir.
 - D. Subalgoritmul amesteca fără logica elementele șirului.
12. Se considera următoarea implementare a algoritmului lui Euclid de determinare a celui mai mare divizor comun a 2 numere naturale.

Subalgorithm *cmmdc*(a, b):

```
If b = 0 then
```

```
    Return a
```

```
EndIf
```

```
Return cmmdc(b, a MOD b)
```

EndSubalgorithm

Precizați care este complexitatea acestui algoritm.

- A. $O(\log_2(a + b))$
 - B. $O(\sqrt{a + b})$
 - C. $O(a + b)$
 - D. $O(\log_2(a))$
13. Se realizează un experiment chimic având la dispoziție 10 eprubete cu substanțe de cantități diferite. Scopul este sa se obțină o singura eprubeta cu lichid. Eprubetele se pot combina doar doua cate doua si, de fiecare data când se combina doua, trebuie turnat tot lichidul din eprubeta cu mai puțin lichid in cea cu mai mult lichid. Eprubetele sunt numerotate de la 1 la 10 si au următoarele cantități de lichid exprimate in mililitri: 5, 27, 6, 18, 10, 2, 16, 8, 15, 9. De exemplu, pentru a se combina eprubetele 1 si 2 este necesar un timp de $5 + 27 = 32$ de secunde (timpul de combinare este suma cantităților de lichid), iar combinarea presupune turnarea conținutului eprubetei 1 in eprubeta 2. Se dorește combinarea eprubetelor doua cate doua astfel încât in

final lichidul sa fie într-o singura eprubeta si timpul total necesar efectuării tuturor combinațiilor sa fie minim.

Care este timpul minim de combinare?

- A. 116
- B. 358
- C. 576
- D. 643

14. Se considera subalgoritmul *imp* care primește ca si parametri de intrare 3 numere naturale nenule **a**, **b**, **n** ($1 \leq a \leq n$, $1 \leq b \leq n$, $1 \leq n \leq 10^6$)

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

If a = b **then**

Return a * a = n

Else

 mj \leftarrow (a + b) / 2

Return *imp*(a, mj, n) **OR** *imp*(mj+1, b, n)

EndIf

EndSubalgorithm

Precizați care dintre următoarele afirmații sunt adevărate considerând apelul inițial *imp*(1, n, n):

- A. Secvența de cod verifica daca numărul n este pătrat perfect.
- B. Complexitatea algoritmului este $O(\sqrt{n})$
- C. Complexitatea algoritmului este $O(\log_2 n)$
- D. Algoritmul utilizează principiul de Divide et Impera pentru a determina daca numărul n este pp.

15. Se considera subalgoritmul *secv* care primește ca si parametri de intrare un sir **a** cu **n** elemente numere naturale, numărul **n** si un număr **p** natural nenul ($1 \leq n \leq 1000$, $1 \leq a[i] \leq 10^9$, $1 \leq i \leq n$, $1 \leq p \leq 10^{20}$).

Subalgorithm *secv*(a, n, p):

 st \leftarrow 1

 prod \leftarrow 1

 L \leftarrow 0

For i \leftarrow 1, n **execute**

 prod \leftarrow prod * a[i]

While prod > p **execute**

 prod \leftarrow prod **DIV** a[st]

 st \leftarrow st + 1

EndWhile

If i - st + 1 > L **then**

 L \leftarrow i - st + 1

EndIf

EndFor

Return L

EndSubalgorithm

Precizați care dintre următoarele afirmații sunt adevărate referitoare la subalgoritmul *secv*.

- A. Subalgoritmul are o complexitate pătratică.
- B. Subalgoritmul returnează lungimea maxima a unei secvențe de elemente cu produsul mai mic strict ca **p**.
- C. Subalgoritmul returnează lungimea maxima a unei secvențe de elemente cu suma mai mica strict ca **p**.
- D. Subalgoritmul returnează lungimea maxima a unei secvențe cu produsul mai mic sau egal decât **p**.

16. Se considera subalgoritmul *modif* care primește ca si parametrii de intrare o matrice pătratică **a** de latura egală cu **n** si numărul natural nenul **n** ($1 \leq n \leq 1000$, $1 \leq a[i][j] \leq 1000$, $1 \leq i, j \leq n$).

Subalgorithm *modif*(a, n):

For i ← 1, n **execute**

For j ← i + 1, n **execute**

 aux ← a[i][j]

 a[i][j] ← a[j][i]

 a[j][i] ← aux

EndFor

EndFor

For i ← 1, n **execute**

For j ← 1, n – i **execute**

 aux ← a[i][j]

 a[i][j] ← a[n-j+1][n-i+1]

 a[n-j+1][n-i+1] ← aux

EndFor

EndFor

EndSubalgorithm

Precizați care dintre următoarele afirmații sunt adevărate referitoare la subalgoritmul *modif*(a, n):

- A. Subprogramul interschimba elementele situate de deasupra de diagonala principala cu cele situate sub diagonala principala, după care interschimba elementele situate deasupra de diagonala secundara cu cele situate sub diagonala secundara.
- B. Elementele de pe cele 2 diagonale rămân nemodificate.
- C. In urma rulării subprogramului, elementele situate deasupra de ambele diagonale vor fi interschimbate cu elementele situate sub ambele diagonale si elementele din celelalte 2 zone vor fi interschimbate intre ele.
- D. Subprogramul interschimba doar elementele de deasupra diagonalei secundare cu cele de sub diagonala secundara.

17. Se considera subalgoritmul *generare* care primește ca si parametrii de intrare o matrice **a** cu toate elementele nule inițial si un parametru **n** număr natural ($1 \leq n \leq 1000$).

Subalgorithm *generare*(a, n):

For i \leftarrow 1, n **execute**

For j \leftarrow 1, n **execute**

If i + j > n + 1 **then**

 a[i][j] \leftarrow i + j - n - 1

Else

 a[i][j] \leftarrow n + 1 - i - j

EndIf

EndFor

EndFor

EndSubalgorithm

Care este suma elementelor din matricea **a** in urma executării subprogramului cu **n** = 20.

- A. 2660
 - B. 1330
 - C. 210
 - D. 420
18. Care este numărul maxim de frunze pe care le putem avea intr-un arbore binar de înălțime minima posibila si 264 de noduri?
- A. 131
 - B. 132
 - C. 128
 - D. 129
19. Se considera subalgoritmul *secv_sum* care primește ca si parametrii de intrare un sir **a** cu **n** elemente numere naturale nenule si numărul natural nenul **n** ($1 \leq n \leq 1000$, $1 \leq a[i] \leq 1000$, $1 \leq i \leq n$)

Subalgorithm *secv_sum*(a, n):

 smax \leftarrow 0

For i \leftarrow 1, n **execute**

For j \leftarrow i, n **execute**

 s \leftarrow 0

For k \leftarrow i, j **execute**

 s \leftarrow s + a[i][j]

EndFor

If smax < s **then**

 smax \leftarrow s

EndIf

EndFor

EndFor

Return smax

EndSubalgorithm

Pentru $n = 100$, care este numărul de pași care se efectuează pentru a determina suma maxima a unei secvențe?

- A. 171700
- B. 338350
- C. 5050
- D. 10100

20. Se considera o matrice sume parțiale **sp** in care pe fiecare poziție este stocata suma tuturor elementelor din submatricea cu coltul stânga sus la poziția (1,1) si coltul dreapta jos la poziția (i, j). Deci, cu alte cuvinte, $\mathbf{sp}[i][j] = (a[1][1] + a[1][2] + \dots + a[1][j] + a[2][1] + a[2][2] + \dots + a[2][j] + \dots + a[i][1] + a[i][2] + \dots + a[i][j])$.

Precizați care dintre următoarele formule poate fi utilizata pentru a determina suma elementelor din submatricea definita de coltul stânga sus la poziția (i1, j1) si coltul dreapta jos la poziția (i2, j2):

- A. $\text{sum} = \mathbf{sp}[i2][j2] - \mathbf{sp}[i1-1][j1-1]$
- B. $\text{sum} = \mathbf{sp}[i2][j2] - \mathbf{sp}[i2][j1-1] - \mathbf{sp}[i1-1][j2]$
- C. $\text{sum} = \mathbf{sp}[i1][j1] + \mathbf{sp}[i2][j2]$
- D. $\text{sum} = \mathbf{sp}[i2][j2] + \mathbf{sp}[i1-1][j1-1] - \mathbf{sp}[i2][j1-1] - \mathbf{sp}[i1-1][j2]$

21. Se considera subalgoritmul *calcul* care primește ca si parametri de intrare un sir **a** cu **n** elemente numere naturale, numărul natural nenul **n** si o valoarea **val** ($1 \leq n \leq 100$, $1 \leq a[i] \leq 100$, $1 \leq i \leq n$, $1 \leq \text{val} \leq 1000$).

Subalgorithm *calcul*(a, n, val):

```

If val = 0 then
    Return 1
EndIf
If n = 0 then
    Return 0
EndIf
n1  $\leftarrow$  calcul(a, n-1, val)
n2  $\leftarrow$  0
If val  $\geq$  a[n] then
    n2  $\leftarrow$  calcul(a, n-1, val - a[n])
EndIf
Return n1 + n2

```

EndSubalgorithm

Precizați ce se afișează știind ca la apelul inițial șirul **a** = [2, 3, 6, 1, 4, 5, 9, 8, 7], **n** = 9 si **val** = 12:

- A. 11
- B. 8
- C. 12
- D. 9

22. Se considera subalgoritmul *interclass* care primește ca si parametrii de intrare un sir **a** cu elementele ordonate crescător reprezentând elementele unei mulțimi matematice, un parametru **n** care este dimensiunea șirului **a**, un sir **b** cu elementele ordonate crescător care de asemenea reprezintă o mulțime matematică si un parametru **m** reprezentând dimensiunea șirului **b** ($1 \leq n \leq 100000$, $1 \leq a[i] \leq 10^6$, $1 \leq i \leq n$, $1 \leq m \leq 100000$, $1 \leq b[j] \leq 10^6$, $1 \leq j \leq m$)

Subalgorithm *interclass*(a, n, b, m):

$c \leftarrow [0] * 200000$

$indc \leftarrow 0$

$inda \leftarrow 1$

$indb \leftarrow 1$

While $inda \leq n$ **AND** $indb \leq m$ **execute**

If $a[inda] < b[indb]$ **then**

$indc \leftarrow indc + 1$

$c[indc] \leftarrow a[inda]$

$inda \leftarrow inda + 1$

Else

If $a[inda] > b[indb]$ **then**

$indc \leftarrow indc + 1$

$c[indc] \leftarrow b[indb]$

$indb \leftarrow indb + 1$

Else

$inda \leftarrow inda + 1$

$indb \leftarrow indb + 1$

EndIf

EndIf

EndWhile

While $inda \leq n$ **execute**

$indc \leftarrow indc + 1$

$c[indc] \leftarrow a[inda]$

$inda \leftarrow inda + 1$

EndWhile

While $indb \leq m$ **execute**

$indc \leftarrow indc + 1$

$c[indc] \leftarrow b[indb]$

$indb \leftarrow indb + 1$

EndWhile

Return (c, indc)

EndSubalgorithm

Precizați care dintre următoarele afirmații referitoare la subalgoritmul de mai sus sunt adevărate:

- A. Subalgoritmul construiește un tablou **c** care conține toate elementele celor 2 vectori in ordine crescătoare.
- B. Complexitatea de calcul al șirului **c** este liniară.
- C. Subalgoritmul construiește un tablou **c** care conține elementele comune celor 2 vectori in ordine crescătoare.

- D. Subalgoritmul construiește și returnează un tablou **c** care conține elementele celor 2 siruri care nu apar în ambele siruri, în ordine crescătoare.

23. Să se determine în câte moduri poate fi scrisă valoarea 13 ca sumă de numere din mulțimea {1, 2, 3, 5} (inspirată de la Mihai Nan, UPB 2024):

- A. 24
- B. 29
- C. 34
- D. 40

24. Se considera subalgoritmul *ordonare* care are ca și parametri de intrare un sir **a** cu **n** elemente numere naturale, un număr natural **n** și 2 parametri **i** și **j** inițializați înainte de apelul inițial cu valoarea 2 ($1 \leq n \leq 1000$, $1 \leq a[i] \leq 10^9$, $1 \leq i \leq n$).

Subalgorithm *ordonare*(a, n, i, j):

If $i \leq n$ **then**

If $j > 1$ **then**

If $a[j] < a[j-1]$ **then**

$a[j] \leftrightarrow a[j-1]$

EndIf

ordonare(a, n, i, j-1)

Else

ordonare(a, n, i+1, i+1)

EndIf

EndIf

EndSubalgorithm

Care dintre următoarele afirmații sunt adevărate referitoare la subalgoritmul *ordonare*?

- A. Subprogramul ordonează crescător șirul **a** cu **n** elemente.
- B. Subprogramul ordonează descrescător șirul **a** cu **n** elemente.
- C. Subprogramul utilizează o variantă de Insertion Sort scrisă recursiv.
- D. Subprogramul utilizează o variantă de Bubble Sort scrisă recursiv.

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