

Simulare Examen Admitere UBB 2024 – 08-09 Iunie 2024
Proba scrisa la Informatica

NOTA IMPORTANTA

In lipsa altor precizări:

- Presupuneti ca toate operatiile aritmetice se efectueaza pe tipuri de date nelimitate (nu exista overflow / underflow).
- Numerotarea indicilor tuturor sirurilor / vectorilor incepe de la 1.
- Toate restrictiile se refera la valorile parametrilor actuali la momentul apelului initial.

1. Se considera algoritmul *calculate(n)*, unde **n** este un numar natural ($1 \leq n \leq 10000$):

Algorithm *calculate(n)*:

b \leftarrow 1

c \leftarrow 0

While **n** > 0 **execute**

If **n MOD 2 = 0** **then**

b \leftarrow **b** * (**n MOD 10**)

Else

c \leftarrow **c** + (**n MOD 10**)

EndIf

n \leftarrow **n DIV 10**

EndWhile

Return **b - c**

EndAlgorithm

Care dintre urmatoarele afirmatii este corecta daca **n = 7458**?

- A. Algoritmul returneaza 0.
- B. Algoritmul returneaza 20.
- C. Algoritmul returneaza -20.
- D. Algoritmul returneaza -3.

2. Se considera algoritmul *processArray(n, arr)*, unde **n** este un numar natural ($1 \leq n \leq 1000$) si **arr** este un vector cu **n** numere naturale ($1 \leq \text{arr}[i] \leq 10000$):

Algorithm *processArray(n, arr)*:

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

If **arr[i] > arr[i + 1]** **then**

arr[i] \leftrightarrow arr[i + 1]

EndIf

EndFor

Return **arr[n]**

EndAlgorithm

Care dintre urmatoarele afirmatii este adevarata?

- A. Algoritmul returneaza cea mai mare valoare din sir.

- B. Algoritmul returnează elementul minim din **arr**.
- C. In cazul in care șirul este ordonat descrescător, se returnează elementul minim din sir.
- D. Algoritmul sortează vectorul **arr** în ordine descrescătoare.

3. Se consideră algoritmul *binarySearch*(arr, n, x), unde **n** este un număr natural ($1 \leq n \leq 1000$), **arr** este un vector cu **n** numere naturale sortate crescător, și **x** este numărul căutat:

Algorithm *binarySearch*(arr, n, x):

```
    left ← 1
    right ← n
    While left ≤ right execute
        mid ← (left + right) DIV 2
        If arr[mid] = x then
            Return mid
        Else
            If arr[mid] < x then
                left ← mid + 1
            Else
                right ← mid - 1
            EndIf
        EndIf
    EndWhile
    Return -1
```

EndAlgorithm

Care dintre următoarele afirmații este adevărată?

- A. Algoritmul returnează -1 dacă **x** nu este găsit în **arr**.
 - B. Algoritmul returnează prima apariție a lui **x** în **arr**.
 - C. Algoritmul returnează ultima apariție a lui **x** în **arr**.
 - D. Algoritmul returnează poziția medie a lui **x** în **arr**.
4. Se consideră algoritmul *magic*(n), unde **n** este un număr natural ($1 \leq n \leq 10000$):

Algorithm *magic*(n):

```
    If n = 0 then
        Return 0
    EndIf
    sum ← 0
    While n > 0 execute
        sum ← sum + (n MOD 10)
        n ← n DIV 10
    EndWhile
    Return sum + magic(sum DIV 2)
```

EndAlgorithm

Care este valoarea returnată de algoritm pentru $n = 1234$?

- A. 18
- B. 11
- C. 15
- D. 16

5. Converteți numărul $7BAC_{16}$ în baza 10:

- A. 31564
- B. 31660
- C. 31563
- D. 31566

6. Fie x, y și z trei variabile booleene. Care dintre următoarele expresii este echivalentă cu **NOT** (x **AND** y) **OR** (y **AND** **NOT** z)?

- A. (**NOT** x **OR** **NOT** y) **OR** (y **AND** **NOT** z)
- B. (x **OR** y) **OR** (y **AND** z)
- C. (**NOT** x **AND** **NOT** y) **OR** (y **AND** **NOT** z)
- D. (**NOT** x **OR** **NOT** y) **OR** **NOT**(**NOT** y **OR** z)

7. Se consideră algoritmul *functie*(x, y), unde x și y sunt numere întregi ($1 \leq x, y \leq 100$):

Algorithm *functie*(x, y):

```
If  $x = 0$  then
    Return  $y$ 
EndIf
If  $y = 0$  then
    Return  $x$ 
EndIf
If  $x > y$  then
    Return functie( $x - y, y$ )
Else
    Return functie( $x, y - x$ )
EndIf
```

EndAlgorithm

Care dintre următoarele afirmații sunt corecte?

- A. Subprogramul *functie* calculează cel mai mare divizor comun al celor 2 numere.
- B. Subprogramul ar fi calculat cel mai mare divizor comun al celor 2 numere dacă condiția de oprire era " $x = y$ ".
- C. Subprogramul calculează cel mai mic multiplu comun al celor 2 numere.

D. Subprogramul nu calculează nimic concludent.

8. Care este complexitatea subprogramului *functie*(x, y) de la grila anterioara?

- A. $O(x + y)$
- B. $O(x)$
- C. $O(y)$
- D. $O(2 * x + 2 * y)$

9. Se consideră următorul algoritm *cautare*(n, m, x, y), unde n și m sunt numere naturale nenule, iar x și y sunt vectori de dimensiune n , respectiv m :

Algorithm *cautare*(n, m, x, y):

For $i \leftarrow 1, n - m + 1$ **execute**

$ok \leftarrow \text{True}$

For $j \leftarrow 1, m$ **execute**

If $x[i + j - 1] \neq y[j]$ **then**

$ok \leftarrow \text{False}$

EndIf

EndFor

If ok **then**

Return i

EndIf

EndFor

Return -1

EndAlgorithm

Precizați care dintre următoarele afirmații sunt corecte referitoare la algoritmul *cautare* definit mai sus:

- A. Subprogramul verifică dacă sirul y este secvență în sirul x .
- B. Subprogramul returnează 1 dacă sirul y este identic cu sirul x .
- C. Subprogramul returnează ultima apariție a sirului y în sirul x .
- D. Subprogramul returnează 1 doar dacă sirul y apare o singură dată în sirul x .

10. Se consideră algoritmul *procesare*(n, m, matrix), unde n și m sunt numere naturale ($1 \leq n, m \leq 100$), iar **matrix** este o matrice bidimensională de dimensiune $n * m$:

Algorithm *procesare*(n, m, matrix):

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

For $j \leftarrow 1, m$ **execute**

If $(i + j) \bmod 2 = 0$ **then**

$\text{matrix}[i][j] \leftarrow \text{matrix}[i][j] * 2$

Else

$\text{matrix}[i][j] \leftarrow \text{matrix}[i][j] \text{ DIV } 2$

EndIf

```

    EndFor
EndFor
Return matrix
EndAlgorithm

```

Știind ca la apelul inițial **matrix** = [[1, 2, 3], [4, 5, 6], [7, 8, 9]], **n** = 3 și **m** = 4, cum va arăta matricea returnată la apelul *procesare*(n, m, matrix)?

- A. [[2, 1, 6], [2, 8, 3], [14, 4, 18]]
- B. [[2, 1, 6], [2, 10, 3], [3, 16, 4]]
- C. [[2, 1, 6], [2, 10, 3], [14, 4, 18]]
- D. [[2, 1, 6], [2, 10, 4], [14, 4, 18]]

11. Care dintre următoarele afirmații sunt false pentru un graf orientat cu **n** noduri și **m** muchii?

- A. Numărul minim de muchii este $n - 1$.
- B. Numărul maxim de muchii este $n * (n - 1) / 2$.
- C. Un graf complet are exact $n * (n - 1)$ muchii.
- D. Un graf orientat poate avea cicluri.

12. Se consideră următorul algoritm *superFactorial*(n), unde **n** este un număr natural ($1 \leq n \leq 100$) și care apelează algoritmul *factorial*(n).

```

Algorithm factorial(n):
    If n = 0 then
        Return 1
    EndIf
    Return n * factorial(n-1)
EndAlgorithm

```

```

Algorithm superFactorial(n):
    If n = 0 then
        Return 1
    EndIf
    Return factorial(n) * superFactorial(n-1)
EndAlgorithm

```

Precizați care dintre următoarele afirmații sunt adevărate.

- A. $\text{superFactorial}(n) = \prod_{i=1}^n i^{n-i+1}$
- B. $\text{superFactorial}(n) = \prod_{i=1}^n i^{n-i-1}$
- C. $\text{superFactorial}(n) = \prod_{i=1}^n i^{n-i}$
- D. $\text{superFactorial}(3) = 12$

13. Se consideră următorul algoritm *divizori*(*n*), unde *n* este un număr natural ($1 \leq n \leq 10^9$).

Algorithm *divizori*(*n*):

sum \leftarrow 0

i \leftarrow 2

While *i* * *i* \leq *n* **execute**

While *n* MOD *i* = 0 **execute**

sum \leftarrow *sum* + *i*

n \leftarrow *n* DIV *i*

EndWhile

i \leftarrow *i* + 1

EndWhile

If *n* > 1 **then**

sum \leftarrow *sum* + *n*

EndIf

Return *sum*

EndAlgorithm

Care este valoarea returnata de *divizori*(360)?

- A. 10
- B. 15
- C. 5
- D. 17

14. Care dintre următoarele variante de raspuns descrie corect ordinea de parcurgere a unui arbore binar in preordine?

- A. Vizitează rădăcina, subarborele stâng, subarborele drept
- B. Vizitează subarborele stâng, subarborele drept, rădăcina
- C. Vizitează subarborele drept, subarborele stâng, rădăcina
- D. Vizitează subarborele stâng, rădăcina, subarborele drept

15. Se consideră următorul algoritm *numarare*(*arr*, *n*, *k*), unde *arr* este un vector cu *n* elemente și *k* este un număr natural nenul.

Algorithm *numarare*(*arr*, *n*, *k*):

count \leftarrow 0

For *i* = 1 to *n*-2 **execute**

For *j* = *i*+1 to *n*-1 **execute**

For *l* = *j*+1 to *n* **execute**

If (*arr*[*i*] + *arr*[*j*] + *arr*[*l*]) MOD *k* = 0 **then**

count \leftarrow *count* + 1

EndIf

EndFor

EndFor

EndFor

Return count
EndAlgorithm

Considerand sirul $arr = [23, 45, 32, 41, 34, 28, 39, 29, 13, 23, 43, 21, 45, 63, 34, 12]$, $n = 16$ si $k = 6$, ce se afiseaza?

- A. 99
- B. 100
- C. 101
- D. 102

16. Se consideră următorul algoritm $suma(arr, n)$, unde **arr** este un vector cu **n** elemente.

```
Algorithm  $suma(arr, n)$ :  
  If  $n = 0$  then  
    Return 0  
   $msis \leftarrow [0] * (n + 1)$   
  For  $i = 1$  to  $n$  execute  
     $msis[i] \leftarrow arr[i]$   
  EndFor  
   $mx \leftarrow 0$   
  For  $i \leftarrow 2$  to  $n$  execute  
    For  $j \leftarrow 1$  to  $i - 1$  execute  
      If  $arr[i] > arr[j]$  AND  $msis[i] < msis[j] + arr[i]$  then  
         $msis[i] \leftarrow msis[j] + arr[i]$   
      EndIf  
    EndFor  
    If  $mx < msis[i]$  then  
       $mx \leftarrow msis[i]$   
    EndIf  
  EndFor  
  Return  $mx$   
EndAlgorithm
```

Ce se returneaza la apelul functiei pentru $arr = [1, 101, 2, 3, 100, 4, 5]$, $n = 7$?

- A. 106
- B. 110
- C. 113
- D. 116

17. Se consideră următorul algoritm $rec(n, k)$, unde **n** și **k** sunt numere naturale ($1 \leq n, k \leq 100$). Algoritmul calculează o valoare recursivă bazată pe **n** și **k**.

```
Algorithm  $rec(n, k)$ :  
  If  $n < 1$  then  
    Return  $k$   
  EndIf
```

Return $rec(n-1, k) + rec(n-2, k)$
EndAlgorithm

Care este valoarea returnată de $rec(5, 2)$?

- A. 26
- B. 12
- C. 20
- D. 32

18. Se consideră următorul algoritm $sum(n)$, unde n este un număr natural. Algoritmul calculează o sumă recursivă pe baza lui n .

Algorithm $sum(n)$:
 If $n = 0$ **then**
 Return 0
 Elseif $n = 1$ **then**
 Return 1
 EndIf
 Return $n + sum(n \text{ DIV } 2) + sum(n \text{ DIV } 3)$
EndAlgorithm

Care este valoarea returnată de $sum(10)$?

- A. 15
- B. 20
- C. 24
- D. 25

19. Se consideră următorul algoritm $ccc(n, adjMatrix)$, unde n este numărul de noduri și **adjMatrix** este matricea de adiacență a unui graf neorientat ($1 \leq n \leq 100$).

Algorithm $ccc(n, adjMatrix)$:
 $visited \leftarrow [0] * (n + 1)$
 $count \leftarrow 0$
 Function $prc(node)$:
 $visited[node] \leftarrow 1$
 For $i \leftarrow 1$ to n **execute**
 If $adjMatrix[node][i] = 1$ **AND** $visited[i] = 0$ **then**
 $prc(i)$
 EndIf
 EndFor
 EndFunction
 For $i = 1$ to n **execute**
 If $visited[i] = 0$ **then**
 $prc(i)$
 EndIf
 $count \leftarrow count + 1$


```
EndFor
Return count
EndAlgorithm
```

Care dintre urmatoarele afirmatii sunt adevarate?

- A. Subalgoritmul *prc* este un algoritm de parcurgere a grafului in adancime.
- B. In cazul in care graful este conex se returneaza 1.
- C. Algoritmul numara cate noduri sunt in graf.
- D. Algoritmul numara din cate componente conexe este format graful.

20. Se consideră următorul algoritm $secv(arr, n)$, unde **arr** este un vector cu **n** elemente.

```
Algorithm secv(arr, n):
  If n = 0 then
    Return 0
  EndIf
  lis ← [1] * (n + 1)
  lds ← [1] * (n + 1)
  For i ← 2 to n execute
    For j ← 1 to i - 1 execute
      If arr[i] > arr[j] then
        lis[i] ← max(lis[i], lis[j] + 1)
      EndIf
    EndFor
  EndFor
  For i ← n-1 to 1 execute
    For j ← n to i + 1 execute
      If arr[i] > arr[j] then
        lds[i] ← max(lds[i], lds[j] + 1)
      EndIf
    EndFor
  EndFor
  maxi ← 1
  For i ← 1 to n execute
    maxi ← max(maxi, lis[i] + lds[i] - 1)
  EndFor
  Return maxi
EndAlgorithm
```

Care este valoarea returnata de algoritm pentru **arr** = [1, 11, 2, 10, 4, 5, 2, 1] si **n** = 8?

- A. 5
- B. 6
- C. 7
- D. 8

21. Se considera functia $g(n)$ care primeste ca si parametru unic de intrare un numar natural nenul n ($1 \leq n \leq 10^3$).

Subalgorithm $g(n)$:

If $n \leq 1$ **then**

return 1

Else

return $g(n - 1) + g(n - 2) + 1$

EndIf

EndSubalgorithm

Care dintre urmatoarele afirmatii sunt adevarate?

- A. $g(10) = 177$
- B. $g(n) = 2 * \text{Fibonacci}(n) - 1$
- C. $g(10) = 89$
- D. $g(10) = 90$

22. Fie subalgoritmul $\text{sum}(a, \text{low}, \text{high})$ care primeste ca si parametrii de intrare un sir a cu n elemente numere naturale nenule ($1 \leq n \leq 10^3$) si doi parametrii low si high , numere naturale.

Subalgorithm $\text{sum}(a, \text{low}, \text{high})$:

If $\text{low} = \text{high}$ **then**

return $a[\text{low}]$

Else

$\text{mid} \leftarrow (\text{low} + \text{high}) \text{ DIV } 2$

$s1 \leftarrow \text{sum}(a, \text{low}, \text{mid})$

$s2 \leftarrow \text{sum}(a, \text{mid} + 1, \text{high})$

$\text{cross_sum} \leftarrow \text{maxi}(a, \text{low}, \text{mid}, \text{high})$

return $\text{max}(s1, s2, \text{cross_sum})$

EndIf

EndSubalgorithm

Subalgorithm $\text{maxi}(a, \text{low}, \text{mid}, \text{high})$:

$\text{sum1} \leftarrow -\infty$

$\text{sum} \leftarrow 0$

For $i \leftarrow \text{mid}, \text{low}, -1$ **execute**

$\text{sum} \leftarrow \text{sum} + a[i]$

If $\text{sum} > \text{sum1}$ **then**

$\text{sum1} \leftarrow \text{sum}$

EndIf

EndFor

$\text{sum2} \leftarrow -\infty$

$\text{sum} \leftarrow 0$

For $i \leftarrow \text{mid} + 1, \text{high}$ **execute**

$\text{sum} \leftarrow \text{sum} + a[i]$

If $\text{sum} > \text{sum2}$ **then**

```

        sum2 ← sum
    EndIf
EndFor
return sum1 + sum2
EndSubalgorithm

```

Considerand apelul initial de forma $\text{sum}(a, 1, n)$, ce face aceasta secventa de cod?

- A. Calculeaza suma maxima a unei secvente de elemente din sir.
- B. Calculeaza suma tuturor elementelor din sir.
- C. Calculeaza suma unei secvente de elemente din sir de suma minima.
- D. Calculeaza suma maxima a unei secvente crescatoare de elemente din sir.

23. Fie subalgoritmul $\text{fct}(a, \text{low}, \text{high}, k)$ care primeste ca si parametrii de intrare un sir **a** cu **n** elemente numere naturale nenule ($1 \leq n \leq 10^3$) si 3 variabile numere naturale nenule **low**, **high** si **k**.

```

Subalgorithm fct(a, low, high, k):
    If low = high then
        return a[low]
    Else
        id ← partitionare(a, low, high)
        If id - low + 1 = k then
            return a[id]
        Elseif id - low + 1 > k then
            return fct(a, low, id - 1, k)
        Else
            return fct(a, id + 1, high, k - (id - low + 1))
    EndIf
EndSubalgorithm

```

```

Subalgorithm partitionare(a, low, high):
    pivot ← a[high]
    i ← low - 1
    For j ← low to high - 1 do
        If a[j] ≤ pivot then
            i ← i + 1
            a[i] ↔ a[j]
        EndIf
    EndFor
    a[i + 1] ↔ a[high]
    return i + 1
EndSubalgorithm

```

Ce face subalgoritmul considerand apelul initial $\text{fct}(a, 1, n, k)$?

- A. Determina si returneaza cel mai mic element din sir.
- B. Determina si returneaza cel mai mare element din sir.
- C. Determina al k-lea cel mai mic element din sir.
- D. Determina si returneaza cel mai mare element din sir atunci cand k este egal cu n.

24. Fie următoarea funcție *mcst* care primește ca si parametrii de intrare un sir cost cu *n* elemente numere naturale nenule ($1 \leq n \leq 10^3$).

Subalgorithm *mcst*(cost, n):

If *n* = 1 **then**

return cost[1]

Elseif *n* = 2 **then**

return cost[2]

Else

return min(cost[*n*] + *mcst*(cost, *n*-1), cost[*n*] + *mcst*(cost, *n*-2))

EndIf

EndSubalgorithm

Ce valoare va returna functia la apelul *mcst*([1, 100, 1, 1, 100, 1], 6)?

- A. 4
- B. 2
- C. 101
- D. 102

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