

**Simulare Examen Admitere UBB 2024 – 29-30 Iunie 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 consideră algoritmul `verifică(n, y)`, unde  $n$  este un număr natural ( $1 \leq n \leq 10^4$ ), iar  $y$  este un vector cu  $n$  elemente numere întregi ( $y[1], y[2], \dots, y[n], -100 \leq y[i] \leq 100$ , pentru  $i = 1, 2, \dots, n$ ):

**Algorithm** `verifică(n, y)`:

$t \leftarrow \text{True}$

$j \leftarrow 1$

**While**  $t = \text{True}$  **AND**  $j < n$  **execute**

**If**  $y[j] > y[j + 1]$  **then**

$t \leftarrow \text{True}$

**else**

$t \leftarrow \text{False}$

**EndIf**

$j \leftarrow j + 1$

**EndWhile**

$\text{return } t$

**EndAlgorithm**

Pentru care dintre următoarele situații algoritmul returnează `True`?

- A. Dacă vectorul  $y$  este format din valorile 10, 9, 8, ..., 1.
  - B. Dacă vectorul  $y$  este strict descrescător.
  - C. Dacă vectorul  $y$  nu are elemente pozitive.
  - D. Dacă vectorul  $y$  are elemente negative situate înaintea celor pozitive.
2. Se consideră un număr natural fără cifre egale cu zero, dat prin șirul  $b$  ( $b[1], b[2], \dots, b[n]$ ) în care se află cele  $n$  cifre ale sale ( $1 \leq n \leq 10$  la momentul apelului inițial). Precizați care dintre următorii algoritmi returnează `True` dacă un număr dat sub această formă este palindrom și `False` în caz contrar. Un număr este palindrom dacă citit de la stânga la dreapta are aceeași valoare ca atunci când se citește de la dreapta la stânga.

A. **Algorithm** `palindrom_1(b, n)`:

$i \leftarrow 1$

$j \leftarrow n$

$k \leftarrow \text{True}$

**While**  $(i \leq j)$  **AND**  $(k = \text{True})$  **execute**

**If**  $b[i] = b[j]$  **then**

```
        i ← i + 1
        j ← j - 1
    else
        k ← False
    EndIf
EndWhile
return k
EndAlgorithm
```

B. **Algorithm** palindrom\_2(b, n):

```
    j ← n
    If (j = 0) OR (j = 1) then
        return True
    EndIf
    If b[1] = b[j] then
        translatare(b, n)
        return palindrom_2(b, n - 2)
    EndIf
    return False
EndAlgorithm
```

**Algorithm** translatare(b, n):

```
    For i = 1, n - 1 execute
        b[i] ← b[i + 1]
    EndFor
EndAlgorithm
```

C. **Algorithm** palindrom\_3(b, n):

```
    i ← n
    j ← 1
    k ← True
    sum1 ← 0
    sum2 ← 0
    While (i > n DIV 2) AND (j ≤ n DIV 2) execute
        sum1 ← sum1 + b[i]
        sum2 ← sum2 + b[j]
        i ← i - 1
        j ← j + 1
    EndWhile
    If sum1 = sum2 then
        k ← True
    else
        k ← False
    EndIf
    return k
EndAlgorithm
```

D. **Algorithm** palindrom\_4(b, n):

$i \leftarrow 1$

$j \leftarrow n$

$k \leftarrow \text{True}$

**While** ( $i \leq j$ ) **AND** ( $k = \text{True}$ ) **execute**

**If** ( $b[i] = b[j]$ ) **AND** ( $i \bmod 2 = 0$ ) **AND** ( $j \bmod 2 = 0$ ) **then**

$i \leftarrow i + 1$

$j \leftarrow j - 1$

**else**

$k \leftarrow \text{False}$

**EndIf**

**EndWhile**

**return** k

**EndAlgorithm**

3. Se consideră algoritmul  $G(m)$ , unde  $m$  este număr natural ( $1 \leq m \leq 10^9$ ).

**Algorithm** G(m):

**If**  $m < 10$  **then**

**return** m

**EndIf**

$v \leftarrow m \bmod 10$

$q \leftarrow G(m \text{ DIV } 10)$

**If**  $v \bmod 3 \leq q \bmod 3$  **then**

**return** v

**EndIf**

**return** q

**EndAlgorithm**

Precizați care dintre următoarele afirmații sunt corecte:

- A. Dacă  $m = 512367$ , valoarea returnată de algoritm este 7.
  - B. Dacă  $m = 8737631$ , valoarea returnată de algoritm este 1.
  - C. Dacă  $m = 5868$ , valoarea returnată de algoritm este 8.
  - D. Dacă  $m = 71$ , valoarea returnată de algoritm este 1.
4. Cifra de control a unui număr natural  $n$  este definită ca fiind cifra obținută prin înlocuirea numărului  $n$  cu suma cifrelor sale până când  $n$  devine o singură cifră. Se consideră un număr natural strict pozitiv  $n$  ( $1 \leq n \leq 10^9$ ). Alegeți algoritmul care calculează corect cifra de control a numărului  $n$ .

A. **Algorithm** cifraDeControl(n):

**If**  $n \bmod 9 = 0$  **AND**  $n \neq 0$  **then**

**return** 9

**EndIf**

**return**  $n \bmod 9$

**EndAlgorithm**

B. **Algorithm** cifraDeControl(n):

**While**  $n \geq 10$  **execute**

$sum \leftarrow 0$

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

$sum \leftarrow sum + (n \bmod 10)$

$n \leftarrow n \text{ DIV } 10$

**EndWhile**

$n \leftarrow sum$

**EndWhile**

    return n

**EndAlgorithm**

C. **Algorithm** cifraDeControl(n):

$sum \leftarrow n$

**While**  $sum \geq 10$  **execute**

$newSum \leftarrow 0$

**While**  $sum > 0$  **execute**

$newSum \leftarrow newSum + (sum \bmod 10)$

$sum \leftarrow sum \text{ DIV } 10$

**EndWhile**

$sum \leftarrow newSum$

**EndWhile**

    return sum

**EndAlgorithm**

D. **Algorithm** cifraDeControl(n):

$sum \leftarrow 0$

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

$sum \leftarrow sum + (n \bmod 10)$

$n \leftarrow n \text{ DIV } 10$

**EndWhile**

    return sum

**EndAlgorithm**

5. Se consideră funcția  $\text{calc\_func}(x, y)$  care calculează un anumit rezultat pentru două numere naturale  $x$  și  $y$  utilizând algoritmul de factorizare. ( $1 \leq x, y \leq 10^9$ ).

**Algorithm** calc\_func(x, y):

$z \leftarrow 2$

    result  $\leftarrow 1$

**While**  $x > 1$  **AND**  $y > 1$  **execute**

$count\_x \leftarrow 0$

$count\_y \leftarrow 0$

**While**  $x \bmod z = 0$  **execute**

$x \leftarrow x \text{ DIV } z$

$count\_x \leftarrow count\_x + 1$

```

EndWhile
While  $y \bmod z = 0$  execute
     $y \leftarrow y \text{ DIV } z$ 
     $\text{count}_y \leftarrow \text{count}_y + 1$ 
EndWhile
If  $\text{count}_x > 0$  AND  $\text{count}_y > 0$  then
     $\text{result} \leftarrow \text{result} * z^{\min(\text{count}_x, \text{count}_y)}$ 
EndIf
 $z \leftarrow z + 1$ 
EndWhile
return result
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate referitor la funcția `calc_func(x, y)`?

- A. Algoritmul calculează numărul de divizori ale numerelor  $x$  și  $y$ .
  - B. Algoritmul calculează cel mai mare divizor comun al celor 2 numere.
  - C. Algoritmul calculează cel mai mic multiplu comun al celor 2 numere.
  - D. Algoritmul determină numărul de divizori al produsului numerelor  $x$  și  $y$ .
6. Se consideră funcția `calc_func(n)` care calculează un anumit rezultat pentru un număr natural  $n$  utilizând algoritmul de factorizare ( $1 \leq n \leq 10^9$ ).

```

Algorithm calc_func(n):
    result  $\leftarrow 1$ 
    p  $\leftarrow 2$ 
    While  $p * p \leq n$  execute
        power  $\leftarrow 0$ 
        While  $n \bmod p = 0$  execute
             $n \leftarrow n \text{ DIV } p$ 
            power  $\leftarrow \text{power} + 1$ 
        EndWhile
        If power  $> 0$  then
             $\text{result} \leftarrow \text{result} * ((p^{\text{power} + 1} - 1) \text{ DIV } (p - 1))$ 
        EndIf
        p  $\leftarrow p + 1$ 
    EndWhile
    If  $n > 1$  then
        result  $\leftarrow \text{result} * (n + 1)$ 
    EndIf
    return result
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate referitor la funcția `calc_func(n)`?

- A. Subalgoritmul calculează numărul de divizori pari ai numărului  $n$ .

- B. Subalgoritmul calculeaza cel mai mic numar mai mare ca  $n$  care are aceiasi factori primi ca numarul  $n$ .
  - C. Subalgoritmul calculeaza rezultatul utilizand un algoritm de descompunere in factori primi in complexitate timp radical.
  - D. Subalgoritmul calculeaza suma divizorilor numarului  $n$ .
7. Se consideră funcția `compute_value(x)` care calculează un anumit rezultat pentru un număr natural  $x$  care contine doar cifre distincte ( $1 \leq x \leq n$ ).

**Algorithm** `compute_value(x)`:

```

count ← 0
fact ← 1
result ← 0
digit_sum ← 0
While  $x > 0$  execute
    count ← count + 1
    digit_sum ← digit_sum + ( $x \bmod 10$ )
     $x \leftarrow x \div 10$ 
EndWhile
For  $i = 2$ , count execute
    fact ← fact *  $i$ 
EndFor
factor ← fact / count
result ← digit_sum * factor * ( $10^{\text{count}} - 1$ ) / 9
return result

```

**EndAlgorithm**

Precizati care dintre urmatoarele afirmatii sunt adevarate:

- A. Algoritmul calculeaza suma tuturor permutarilor circulare ale numarului  $n$ .
  - B. Algoritmul calculeaza suma cifrelor numarului  $n$  si o inmulteste cu 10 la puterea numarului de cifre - 1.
  - C. Algoritmul determina suma tuturor numerelor obtinute prin permutarea cifrelor numarului  $n$ .
  - D. Algoritmul determina suma tuturor permutarilor fara puncta fixe ale cifrelor numarului  $n$ .
8. Se consideră funcția `check_number(y)` care determină dacă un număr natural  $y$  îndeplinește o anumită proprietate ( $1 \leq y \leq 10^9$ ).

**Algorithm** `is_prime(n)`:

```

If  $n < 2$  then
    return False
EndIf
For  $i = 2$ , sqrt( $n$ ) execute
    If  $n \bmod i = 0$  then
        return False
    EndIf

```

```

EndFor
    return True
EndAlgorithm

```

```

Algorithm check_number(y):
    length  $\leftarrow$  0
    temp  $\leftarrow$  y
    While temp > 0 execute
        length  $\leftarrow$  length + 1
        temp  $\leftarrow$  temp DIV 10
    EndWhile
    pow  $\leftarrow$  10(length - 1)
    For i = 0, length - 1 execute
        If is_prime(y) then
            return True
        EndIf
        y  $\leftarrow$  (y DIV 10) + (y MOD 10) * pow
    EndFor
    return False
EndAlgorithm

```

Alegeți variantele corecte din cele de mai jos:

- A. Algoritmul utilizează o funcție auxiliară pentru a verifica dacă un număr este prim.
- B. Algoritmul verifica daca numarul y este prim sau daca are vreo permutare circulara care sa fie numar prim.
- C. Algoritmul returnează True dacă și numai dacă numărul inițial y este prim.
- D. Complexitatea subalgoritmului `check_number(y)` are complexitate  $O(\lg(y))$ .

9. Se consideră funcția `fib_matrix(n)` care primește ca si parametru unic de intrare un numar natural nenul n ( $1 \leq n \leq 1000$ ). Subprogramul `matrix_mult` primește ca si parametrii 2 matrici de numere naturale si returneaza o matrice care contine elementele produsului celor 2 matrici transmise ca parametru.

Definitia sirului lui Fibonacci:  $F(0) = 0$ ,  $F(1) = 1$ ,  $F(2) = 1$ ,  $F(n) = F(n-1) + F(n-2)$

```

Algorithm matrix_pow(M, p):
    result  $\leftarrow$  [[1, 0], [0, 1]]
    base  $\leftarrow$  M
    While p > 0 execute
        If p MOD 2 = 1 then
            result  $\leftarrow$  matrix_mult(result, base)
        EndIf
        base  $\leftarrow$  matrix_mult(base, base)
        p  $\leftarrow$  p DIV 2
    EndWhile
    return result
EndAlgorithm

```

```

Algorithm fib_matrix(n):
    If n = 0 then
        return 0
    EndIf
    M  $\leftarrow$  [[1, 1], [1, 0]]
    result  $\leftarrow$  matrix_pow(M, n - 1)
    return result[0][0]
EndAlgorithm

```

Precizati care dintre urmatoarele afirmatii sunt adevarate:

- A. Subprogramul fib\_matrix(n) returneaza Fibonacci(n).
- B. Subprogramul fib\_matrix(n) returneaza Fibonacci(n + 1).
- C. Subalgoritmul returneaza intotdeauna un numar Fibonacci.
- D. Subprogramul utilizeaza un algoritm de exponentiere rapida.

10. Se consideră funcția `process_array(a, n)` care primește un șir `a` cu `n` elemente întregi, elimina anumite elemente și le dublica pe altele.

```

Algorithm is_prime(x):
    If x < 2 then
        return False
    EndIf
    For i = 2, sqrt(x) execute
        If x MOD i = 0 then
            return False
        EndIf
    EndFor
    return True
EndAlgorithm

Algorithm process_array(a, n):
    For i  $\leftarrow$  1, n execute
        If is_prime(a[i]) then
            For j  $\leftarrow$  i + 1, n execute
                a[j - 1]  $\leftarrow$  a[j]
            EndFor
            n  $\leftarrow$  n - 1
            i  $\leftarrow$  i - 1
        Else
            For j  $\leftarrow$  n, i, -1 execute
                a[j + 1]  $\leftarrow$  a[j]
            EndFor
            n  $\leftarrow$  n + 1
            i  $\leftarrow$  i + 1
        EndIf
    EndFor

```



return a  
**EndAlgorithm**

Precizati care este afectul subalgoritmului process\_array(a, n):

- A. Subalgoritmul dubleaza toate elementele pare din sir.
- B. Subalgoritmul elimina din sir toate elementele prime.
- C. Subalgoritmul dubleaza toate elementele pare din sir si dintre elementele impare, le elimina pe cele care sunt prime.
- D. Subalgoritmul elimina din sir toate elementele pare si le dublica pe cele prime.

11. Se consideră funcția  $f(b, m)$  care primește ca si parametrii de intrare un sir b cu m elemente numere naturale nenule ( $1 \leq m \leq 10^6$ ).

**Algorithm** f(b, m):  
 result  $\leftarrow$  1  
**For** j = 1, m **execute**  
     result  $\leftarrow$  (result \* (b[j] MOD 10)) MOD 10  
**EndFor**  
 return result  
**EndAlgorithm**

Care dintre următoarele afirmații sunt adevărate referitor la funcția  $f(b, m)$ ?

- A.  $f([3243, 23423, 134, 23432, 432131, 432132, 432, 513, 1647], 8) = 8$
- B.  $f([2, 4, 5, 6, 8, 9, 19, 10], 8) = 0$
- C.  $f([212, 3, 423, 41, 43, 765, 7854, 6, 234, 135, 44224], 11) = 0$
- D.  $f([1, 2, 3, 4], 4) = 2$

12. Se consideră funcția process\_digits(y) care primește ca si parametru unic de intrare un numar natural nenul y ( $100 \leq y \leq 10^9$ ).

**Algorithm** process\_digits(y):  
 freq  $\leftarrow$  [0] \* 11  
**While** y > 0 **execute**  
     d  $\leftarrow$  y MOD 10  
     freq[d]  $\leftarrow$  freq[d] + 1  
     y  $\leftarrow$  y DIV 10  
**EndWhile**  
 min\_num  $\leftarrow$  0  
**For** i = 1 to 9 **execute**  
     **If** freq[i] > 0 **then**  
         min\_num  $\leftarrow$  i  
         freq[i]  $\leftarrow$  freq[i] - 1  
         **Break**  
     **EndIf**  
**EndFor**  
**For** i = 0 to 9 **execute**  
     **While** freq[i] > 0 **execute**

```

        min_num ← min_num * 10 + i
        freq[i] ← freq[i] - 1
    EndWhile
EndFor
return min_num
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate referitor la funcția `process_digits(y)`?

- A. Algoritmul determina cel mai mic număr care se poate obține cu cifrele numărului  $n$ .
- B. Algoritmul sortează în ordine crescătoare cifrele din numărul  $n$ .
- C. Algoritmul utilizează algoritmul de sortare prin numărare.
- D. Complexitatea programului este  $O(n)$ .

13. Mai jos se dau diverși algoritmi. În cazul carora este complexitatea precizată corectă?

A. **Algorithm** func1( $n$ ):

```

    If  $n = 1$  then
        return 1
    EndIf
    sum ← 0
    For  $i = 1$  to  $n$  execute
        For  $j = 1$  to  $i$  execute
            sum ← sum + 1
        EndFor
    EndFor
    return sum
EndAlgorithm

```

COMPLEXITATE:  $O(n * n)$

B. **Algorithm** func2( $a, n$ ):

```

    If  $n = 1$  then
        return  $a[1]$ 
    EndIf
     $x \leftarrow \text{func2}(a, n \text{ DIV } 2)$ 
     $y \leftarrow \text{func2}(a, n \text{ DIV } 2)$ 
    return  $x + y$ 
EndAlgorithm

```

COMPLEXITATE:  $O(n)$

C. **Algorithm** func3( $a, n$ ):

```

    If  $n = 1$  then
        return
    EndIf
    For  $i = 1$  to  $n$  execute

```

```

        a[i] ← a[i] * 2
    EndFor
    mid ← n DIV 2
    left_sum ← 0
    right_sum ← 0
    For i = 1 to mid execute
        left_sum ← left_sum + a[i]
    EndFor
    For i = mid + 1 to n execute
        right_sum ← right_sum + a[i]
    EndFor
    func3(a, mid)
    func3(a, n - mid)
EndAlgorithm
COMPLEXITATE  $O(2^n)$ 

```

D. **Algorithm** func4(a, n):

```

    For i = 1 to n execute
        For j = 1 to n execute
            k ← i
            While k ≤ j execute
                a[i][j] ← a[i][j] + 1
                k ← k + 1
            EndWhile
        EndFor
    EndFor
    return a
EndAlgorithm
COMPLEXITATE  $O(n^3)$ 

```

14. Se consideră funcția  $\text{max\_sequence}(a, n, \text{SUM})$  care primește ca si parametrii de intrare un sir a cu n elemente numere naturale nenule si o variabila SUM ( $1 \leq n \leq 1000, 1 \leq \text{SUM} \leq 10^9$ )

**Algorithm** max\_sequence(a, n, SUM):

```

    max_len ← 0
    current_sum ← 0
    start ← 1
    end ← 1
    While end ≤ n execute
        current_sum ← current_sum + a[end]
        While current_sum > SUM AND start ≤ end execute
            current_sum ← current_sum - a[start]
            start ← start + 1
        EndWhile
        If end - start + 1 > max_len then
            max_len ← end - start + 1
        EndIf
        end ← end + 1
    EndWhile

```

```

        EndIf
        end ← end + 1
    EndWhile
    return max_len
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate referitor la funcția `max_sequence(a, n, SUM)`?

- A. Subalgoritmul calculează lungimea primei secvențe de suma mai mare sau egală cu SUM.
- B. Subalgoritmul calculează lungimea ultimei secvențe de suma mai mare sau egală cu SUM.
- C. Subalgoritmul calculează lungimea celei mai lungi secvențe de numere ordonate și suma mai mică sau egală cu SUM.
- D. Subalgoritmul calculează lungimea celei mai lungi secvențe de elemente cu suma mai mică sau egală cu SUM.

15. Se consideră funcția `generate_and_sum_matrix(n)` care generează o matrice de dimensiune  $n \times n$  conform specificațiilor și calculează suma tuturor elementelor din matrice.

```

Algorithm generate_and_sum_matrix(n):
    sum_elements ← 0
    For i = 1 to n execute
        For j = 1 to n execute
            If i MOD 4 = 1 then
                matrix[i][j] ← j
            Elseif i MOD 4 = 2 then
                matrix[i][j] ← 2
            Elseif i MOD 4 = 3 then
                matrix[i][j] ← j
            Else
                matrix[i][j] ← 4
            EndIf
            sum_elements ← sum_elements + matrix[i][j]
        EndFor
    EndFor
    return sum_elements
EndAlgorithm

```

Precizați care este valoarea returnată de subalgoritm pentru  $n = 100$ :

- A. 266600
- B. 267550
- C. 267300
- D. 267500

16. Se consideră funcția `check_value(y)` care primește ca si parametru unic de intrare un numar natural nenul  $y$ .

```

Algorithm check_value(y):
    If  $y < 0$  then
         $y \leftarrow -y$ 
    EndIf
    return binary_check(0, y, y)
EndAlgorithm

```

```

Algorithm binary_check(low, high, target):
    If  $low > high$  then
        return False
    EndIf

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

    If  $patrat * mid = target$  then
        return True
    ElseIf  $patrat * mid < target$  then
        return binary_check( $mid + 1$ , high, target)
    Else
        return binary_check(low,  $mid - 1$ , target)
    EndIf
EndAlgorithm

```

Care dintre următoarele afirmații sunt adevărate referitor la funcția `check_value(y)`?

- A. Algoritmul verifica daca numarul  $y$  este patrat perfect.
  - B. Algoritmul verifica daca numarul  $y$  este cub perfect.
  - C. Algoritmul are o complexitate liniara folosind Divide et Impera.
  - D. Algoritmul are o complexitate logaritmica.
17. Se consideră funcția `rec(x, m, p, q, r)` care primește ca si parametrii de intrare un sir  $x$  cu  $m$  elemente numere intregi nenule ( $1 \leq m \leq 1000$ ).

```

Algorithm rec(x, m, p, q, r):
    If  $p > m$  then
        return r
    EndIf
    If  $x[p] \text{ MOD } 2 = 0$  then
         $is \leftarrow \text{rec}(x, m, p + 1, q + x[p], \max(r, q + x[p]))$ 
    Else
         $is \leftarrow \text{rec}(x, m, p + 1, q, r)$ 
    EndIf
     $es \leftarrow \text{rec}(x, m, p + 1, q, r)$ 

```

**return** max(is, es)

**EndAlgorithm**

Precizati care dintre urmatoarele afirmatii sunt adevarate?

- A.  $\text{rec}([-2, 4, 3, 5, 1, 4, -4, 6, 3, 10, 2, 6], 12, 1, 0, -\text{inf}) = 28$
- B.  $\text{rec}([-2, 4, 3, 5, 1, 4, -4, 6, 3, 10, 2, 6], 12, 1, 0, -\text{inf}) = 30$
- C.  $\text{rec}([-2, 4, 3, 5, 1, 4, -4, 6, 3, 10, 2, 6], 12, 1, 0, -\text{inf}) = 32$
- D.  $\text{rec}([-2, 4, 3, 5, 1, 4, -4, 6, 3, 10, 2, 6], 12, 1, 0, -\text{inf}) = 26$

18. Care este numarul de anagrame ale cuvintului autoutilitara ?

- A. 6.227.020.800
- B. 518.918.400
- C. 86.486.400
- D. 43.243.200

19. Se consideră funcția  $f(n)$  care primește ca si parametru unic de intrare un numar natural nenul  $n$  ( $1 \leq n \leq 100$ ).

**Algorithm**  $f(n)$ :

$dp \leftarrow [[0] * 11] * 11$  // matrice nula de  $11 * 11$  elemente

**For**  $d = 1$  to 9 **execute**

$dp[0][d] \leftarrow 1$

**EndFor**

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

**For**  $d1 = 0$  to 9 **execute**

**For**  $d2 = 0$  to 9 **execute**

**If**  $d1 \% 2 = 0$  **OR**  $d2 \% 2 = 0$  **then**

$dp[i][d1] \leftarrow dp[i][d1] + dp[i-1][d2]$

**EndIf**

**EndFor**

**EndFor**

**EndFor**

$result \leftarrow 0$

**For**  $d = 0$  to 9 **execute**

$result \leftarrow result + dp[n - 1][d]$

**EndFor**

**return** result

**EndAlgorithm**

Precizati care dintre urmatoarele afirmatii sunt adevarate:

- A. Pentru  $n = 4$  se returneaza 4375
- B. Pentru  $n = 5$  se returneaza 35525
- C. Pentru  $n = 6$  se returneaza 287500
- D. Pentru  $n = 5$  se returneaza 35625

20. Se considera algoritmul  $\text{pattern}(i1, j1, i2, j2, a)$  care primește ca si parametrii de intrare 4 indici  $i1, i2, j1, j2$  si o matrice  $a$ .

```

Algorithm pattern(i1, j1, i2, j2, a):
    If i1 = i2 then
        return
    EndIf
    miji  $\leftarrow$  (i1 + i2) DIV 2
    mijj  $\leftarrow$  (j1 + j2) DIV 2
    For i = i1 to miji execute
        For j = j1 to mijj execute
            a[i][j]  $\leftarrow$  1
        EndFor
    EndFor
    pattern(i1, mijj+1, miji, j2)
    pattern(miji+1, j1, i2, mijj)
    pattern(miji+1, mijj+1, i2, j2)
EndAlgorithm

```

Precizati care dintre urmatoarele afirmatii sunt adevarate:

- A. In urma apelului pattern(1, 1, 8, 8) in matrice vor fi 37 valori de 0.
- B. In urma apelului pattern(1, 1, 8, 8) in matrice vor fi 34 valori de 1.
- C. In urma apelului pattern(1, 1, 8, 8) in matrice vor fi 27 valori de 0.
- D. In urma apelului pattern(1, 1, 8, 8) in matrice vor fi 37 valori de 1.

21. Se consideră algoritmul ceFace(n, a), unde a este un tablou unidimensional cu cel mult  $10^6$  elemente și n reprezintă numărul elementelor din a. Subalgoritmul A(n, a) produce o schimbare asupra tabloului a, sortând elementele crescător folosind metoda Selection Sort. (R. Cotoi)

```

Algorithm ceFace(n, a):
    A(n, a)
    For i  $\leftarrow$  1 to n - 2 execute
        For j  $\leftarrow$  i + 1 to n - 1 execute
            st  $\leftarrow$  j + 1
            dr  $\leftarrow$  n
            k  $\leftarrow$  n
            While st  $\leq$  dr execute
                k  $\leftarrow$  (st + dr) / 2
                If a[i] + a[j] > a[k] then
                    st  $\leftarrow$  k + 1
                Else
                    dr  $\leftarrow$  k - 1
                EndIf
            EndWhile
            cnt  $\leftarrow$  cnt + (dr - j)
        EndFor
    EndFor
    Write cnt

```

**EndAlgorithm**

Care dintre următoarele afirmații sunt adevărate?

- A. Algoritmul *ceFace* are complexitatea  $O(n^3)$ ;
- B. Algoritmul *ceFace* are complexitatea  $O(n^2 * \log(n))$  și calculează numărul tripletelor  $(i, j, k)$  din tabloul **a** cu proprietatea că  $i - j + k > 0$ .
- C. Algoritmul *ceFace* are complexitatea  $O(n^3)$  și calculează numărul tripletelor  $(i, j, k)$  din tabloul **a** cu proprietatea că  $i + j - k > 0$ .
- D. Algoritmul *ceFace* calculează numărul triunghiurilor distincte care au lungimile laturilor în tabloul **a**.

22. Se consideră algoritmul *ceFace*( $n, m, a, c, d, e, f$ ), unde **a** este un tablou bidimensional cu cel mult  $10^4 \times 10^4$  numere naturale, **n** și **m** reprezintă dimensiunea lui **a** (**n** linii și **m** coloane), iar **b** este un tablou bidimensional cu cel mult  $10^4 \times 10^4$  elemente, inițial toate **0**. Parametrii **c, d, e, f** sunt numere întregi, cel mult  $10^4$ . **Pentru tablourile bidimensionale, indexarea se face de la 1.** (R. Cotoi)

**Algorithm** *ceFace*( $n, m, a, c, d, e, f$ ):

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

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

$b[i][j] \leftarrow b[i-1][j] + b[i][j-1] - b[i-1][j-1] + a[i][j]$

**EndFor**

**EndFor**

$g \leftarrow b[d][f] - b[c-1][f] - b[d][e-1] + b[c-1][e-1]$

**EndAlgorithm**

Care dintre următoarele afirmații sunt adevărate?

- A. În urma executării algoritmului,  $b[n][m]$  conține suma elementelor din matricea **a**.
- B. Variabila **g** va reprezenta, după executarea algoritmului, suma elementelor din submatricea extrasă din **a** cu colțul stânga sus la poziția (**c, e**) și colțul dreapta jos la poziția (**d, f**).
- C. Variabila **g** va reprezenta, după executarea algoritmului, suma elementelor din submatricea extrasă din **a** cu colțul stânga sus la poziția (**c, d**) și colțul dreapta jos la poziția (**e, f**).
- D. După execuția programului, în matricea **b** nu va exista niciun număr negativ.

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

**Algorithm** *dc*( $n, \text{adjMatrix}$ ):

$\text{visited} \leftarrow [0] * 101$

**Function** DFS(**node**, **parent**):

$\text{visited}[\text{node}] \leftarrow \text{True}$

**For**  $i = 1$  to **n** **execute**

**If**  $\text{adjMatrix}[\text{node}][i] = 1$  **AND NOT**  $\text{visited}[i]$  **then**



```
        If DFS(i, node) then
            Return True
        EndIf
    Elseif adjMatrix[node][i] = 1 AND i ≠ parent then
        Return True
    EndIf
EndFor
Return False
EndFunction

For i = 1 to n execute
    If NOT visited[i] then
        If DFS(i, -1) then
            Return True
        EndIf
    EndIf
EndFor
Return False
EndAlgorithm
```

Precizati care dintre urmatoarele afirmatii sunt adevarate:

- A. Algoritmul  $dc(n, adjMatrix)$  returneaza false daca si numai daca graful este arbore.
- B. Algoritmul  $dc(n, adjMatrix)$  returneaza false daca si numai daca graful este aciclic.
- C. Algoritmul  $dc(n, adjMatrix)$  returneaza true in cazul in care graful este Hamiltonian.
- D. Exista grafuri Euleriene pentru care algoritmul  $dc(n, adjMatrix)$  returneaza false.

24. Se considera toate cuvintele de lungime 4 care se pot forma cu litere mici si mari ale alfabetului englez. Sa se determine cate dintre aceste cuvinte au proprietatea ca nu contin 2 vocale alaturate.

- A. 5510736
- B. 1093680
- C. 6604416
- D. 131208

Nr. Grila	Raspuns	Nr. Grila	Raspuns
1.	A, B	13.	A, B, D
2.	A, B	14.	D
3.	D	15.	D
4.	A, B, C	16.	B, D
5.	B	17.	C
6.	C, D	18.	D
7.	C	19.	A, C, D
8.	A, B, D	20.	C, D
9.	A, C, D	21.	D
10.	B	22.	A, B, D
11.	B, C	23.	B, C
12.	C	24.	C