

1. Se consideră subalgoritmul $ceFace(n)$, unde n este un număr natural ($1 \leq n \leq 10000$).

```
Subalgoritm ceFace(n):  
  nr ← 0  
  Pentru d ← 1, n execută  
    Dacă n MOD d = 0 atunci  
      nr ← nr + 1  
    SfDacă  
  SfPentru  
  Dacă nr = 2 atunci  
    returnează adevărat  
  altfel  
    returnează fals  
  SfDacă  
SfSubalgoritm
```

Precizați care dintre următoarele afirmații sunt adevărate:

- A. Subalgoritmul returnează *adevărat* dacă numărul n este impar.
- B. Subalgoritmul returnează *adevărat* dacă numărul n este par.
- C. Subalgoritmul returnează *adevărat* dacă numărul n este prim.
- D. Subalgoritmul returnează *adevărat* dacă numărul n este pătrat perfect.

2. Știind că $x < y$ (x și y sunt numere reale), care din următoarele expresii are valoarea *adevărat* dacă și numai dacă numărul memorat în t (t număr real) NU aparține intervalului (x, y) ?

- A. $(t > x)$ SAU $(t < y)$
- B. $(t \leq x)$ SAU $(t \geq y)$
- C. $(t \leq x)$ ȘI $(t \geq y)$
- D. $(t > x)$ ȘI $(t < y)$

3. Fie subalgoritmul $f(n)$ unde n este un număr natural ($1 \leq n \leq 10000$).

```
Subalgoritm f(n):  
  r ← 0  
  Când n > 0 execută  
    r ← r + (n MOD 10) * (n MOD 2)  
    n ← n DIV 10  
  SfCând  
  returnează r  
SfSubalgoritm
```

Alegeți variantele care completează corect spațiul subliniat din subalgoritmul de mai jos astfel încât cei doi subalgoritmi să returneze mereu aceeași valoare.

```
Subalgoritm fr(n):  
  Dacă n > 0 atunci  
    returnează _____  
  SfDacă  
  returnează 0  
SfSubalgoritm
```

- A. $(n \text{ MOD } 2) * (n \text{ MOD } 10) + fr(n \text{ DIV } 10)$
- B. $(n \text{ MOD } 2) * (n \text{ MOD } 10) * fr(n \text{ DIV } 10)$
- C. $(n \text{ MOD } 10) + fr(n \text{ DIV } 10)$
- D. $(n \text{ MOD } 2) * (n \text{ MOD } 10) + fr(n \text{ MOD } 10)$

4. Fie subalgoritmul $f(n)$ unde n este un număr natural ($1 \leq n \leq 10000$).

```
Subalgoritm f(n):
  Pentru i ← 1, n execută
    Pentru j ← 1, 2 * i - 1 execută
      Scrie '*'
    SfPentru
  SfPentru
SfSubalgoritm
```

Precizați care dintre următoarele afirmații sunt adevărate:

- A. Pentru $n = 3$ subalgoritmul afișează 3 stelute
- B. Pentru $n = 3$ subalgoritmul afișează 9 stelute
- C. Pentru ca subalgoritmul să afișeze 1154 de stelute valoarea lui n trebuie să fie 34
- D. Pentru ca subalgoritmul să afișeze 289 de stelute valoarea lui n trebuie să fie 17

5. Subalgoritmul de mai jos are ca parametri de intrare un vector v cu n numere naturale ($v[1], v[2], \dots, v[n]$) și numărul întreg n ($2 \leq n \leq 10000$). Operatorul $/$ reprezintă împărțirea reală (ex. $3/2=1,5$). Vectorul v conține cel puțin un număr par și cel puțin un număr impar.

```
Subalgoritm fn(v, n):
  a ← 0
  b ← 0
  Pentru i ← 1, n execută
    Dacă v[i] MOD 2 = 0 atunci
      a ← a + v[i]
      b ← b + 1
    SfDacă
  SfPentru
  returnează a / b
SfSubalgoritm
```

Precizați care dintre următoarele afirmații sunt adevărate:

- A. Subalgoritmul returnează numărul de elemente pare din vectorul v
- B. Subalgoritmul returnează media elementelor pare din vectorul v
- C. Subalgoritmul returnează suma elementelor pare din vectorul v
- D. Subalgoritmul returnează media elementelor impare din vectorul v

6. Subalgoritmul de mai jos are ca parametri de intrare un vector v cu n numere naturale ($v[1], v[2], \dots, v[n]$) și numărul întreg n ($1 \leq n \leq 10000$).

```
Subalgoritm fn(v, n):
  a ← 0
  Pentru i ← 1, n execută
    ok ← 1
    b ← v[i]
    CâtTimp (b ≠ 0) ȘI (ok = 1) execută
      Dacă b MOD 2 = 0 atunci
        ok ← 0
      SfDacă
      b ← b DIV 10
    SfCâtTimp
    Dacă ok = 1 atunci
      a ← a + v[i]
    SfDacă
  SfPentru
  returnează a
SfSubalgoritm
```

Precizați care dintre următoarele afirmații sunt adevărate:

- A. Subalgoritmul returnează suma elementelor impare din vectorul v
- B. Subalgoritmul returnează suma elementelor din vectorul v care sunt puteri ale lui 2
- C. Subalgoritmul returnează suma elementelor din vectorul v care au în componența lor doar cifre pare
- D. Subalgoritmul returnează suma elementelor din vectorul v care au în componența lor doar cifre impare

7. Precizați care dintre următorii subalgoritmi calculează modulul (valoarea absolută) unui număr întreg. Vom presupune că o expresie logică are valoarea 1 dacă este adevărată și 0 dacă este falsă.

- A.
- ```
Subalgoritm modul(n):
 returnează n * (-2 * (n < 0) + 1)
SfSubalgoritm
```
- B.
- ```
Subalgoritm modul(n):
    Dacă n < 0 atunci
        returnează n * (-1)
    SfDacă
    returnează n
SfSubalgoritm
```
- C.
- ```
Subalgoritm modul(n):
 Dacă n < 0 atunci
 returnează n * (-1)
 altfel
 returnează n
 SfDacă
SfSubalgoritm
```
- D.
- ```
Subalgoritm modul(n):
    Dacă n > 0 atunci
        returnează n * (-1)
    altfel
        returnează n
    SfDacă
SfSubalgoritm
```

8. Care este valoarea expresiei de mai jos, dacă $x = 15$ și $y = 17$?

$(\text{NU } (x \text{ MOD } 10 = 0)) \text{ \&I } (y \text{ MOD } 2 = 0) \text{ \&I } (x < y)$

- A. *adevărat* B. *fals* C. Eroare D. Expresia nu poate fi evaluată

9. Se consideră subalgoritmul recursiv $\text{ceFace}(n, i)$, unde n este un număr natural ($2 \leq n \leq 1000$).

```
Subalgoritm ceFace(n, i):
    Dacă i = 1 atunci
        returnează i
    altfel
        Dacă n MOD i = 0 atunci
            returnează i + ceFace(n, i - 1)
        altfel
            returnează ceFace(n, i - 1)
    SfDacă
SfSubalgoritm
```

Precizați care dintre următoarele afirmații sunt adevărate pentru apelul $\text{ceFace}(n, n)$.

- A. Subalgoritmul returnează succesul celui mai mare divizor al lui n
- B. Subalgoritmul returnează suma numerelor naturale neprime, până la n inclusiv
- C. Subalgoritmul returnează suma divizorilor proprii ai numărului n
- D. Subalgoritmul returnează suma divizorilor proprii și improprii ai numărului n

10. Subalgoritmul $\text{magic}(s, n)$ are ca parametri de intrare un șir s cu n caractere ($s[1], s[2], \dots, s[n]$) și numărul întreg n ($1 \leq n \leq 10000$).

```

Subalgoritm magic(s, n):
    i ← 1
    f ← 1
    CâtTimp i ≤ n DIV 2 execută
        Dacă s[i] ≠ s[n - i + 1] atunci
            f ← 0
        SfDacă
        i ← i + 1
    SfCâtTimp
    returnează f
SfSubalgoritm

```

Precizați care dintre următoarele afirmații sunt adevărate:

- A. Subalgoritmul returnează 1 dacă s are un număr par de caractere.
- B. Subalgoritmul returnează 1 dacă s are un număr impar de caractere.
- C. Subalgoritmul returnează 1 dacă s este un palindrom.
- D. Subalgoritmul returnează 1 dacă s conține doar caractere distincte.

11. Care dintre următoarele expresii au valoarea *adevărat* dacă și numai dacă x este număr impar și negativ? Notăm cu $|x|$ valoarea absolută a lui x (modulul lui x).

- A. $(|x| \bmod 2 = 1) \text{ \textbf{\textit{ȘI}} } (x < 0)$
- B. $\text{NU } ((|x| \bmod 2 = 0) \text{ \textbf{\textit{ȘI}} } (x \geq 0))$
- C. $\text{NU } ((|x| \bmod 2 = 0) \text{ \textbf{\textit{SAU}} } (x \geq 0))$
- D. $(|x| \bmod 2 \neq 0) \text{ \textbf{\textit{SAU}} } (x < 0)$

12. Subalgoritmul $ceFace(n)$ are ca parametru de intrare un număr natural n ($0 \leq n \leq 10000$).

```

Subalgoritm ceFace(n):
    s ← 0
    CâtTimp n > 0 execută
        c ← n MOD 10
        Dacă c MOD 2 ≠ 0 atunci
            s ← s + c
        SfDacă
        n ← n DIV 10
    SfCâtTimp
    returnează s
SfSubalgoritm

```

Ce va returna apelul $ceFace(1234)$?

- A. 4
- B. 10
- C. 60
- D. 0

13. Considerăm un șir de caractere și o funcție f care primește ca parametru un caracter și returnează 1 dacă acel caracter este cifră și 0 altfel. Care dintre următoarele abordări determină dacă șirul de caractere este format numai din cifre?

- A. Verificăm dacă funcția f , aplicată pe fiecare caracter al șirului de caractere, returnează întotdeauna 1.
- B. Verificăm dacă suma valorilor returnate de f , aplicată pe fiecare caracter al șirului de caractere, este egală cu lungimea șirului de caractere.
- C. Verificăm dacă funcția f , aplicată pe fiecare caracter al șirului de caractere, returnează cel puțin o dată 1.
- D. Aplicăm funcția f pe caractere alese aleatoriu din șir până când sunt returnate un număr de valori egale cu 1 egal cu lungimea șirului.

14. Care dintre algoritmi următorii pot fi implementați în așa fel încât să aibă complexitate de timp liniară ($O(n)$)?

- A. Algoritmul de căutare secvențială a unui element într-un vector de n numere
- B. Algoritmul de sortare prin inserție a unui tablou unidimensional de n numere
- C. Algoritmul de căutare al numărului maxim într-un vector nesortat de n numere
- D. Algoritmul de determinare a sumei elementelor de pe diagonala principală a unei matrice pătratice cu n linii și n coloane.

15. Se consideră subalgoritmul $f(a, b)$, unde a și b sunt numere naturale ($1 \leq a, b \leq 10000$).

```

Subalgoritm f(a, b):
    m ← a
    CâtTimp b MOD m > 0 execută
        m ← m + 1
    SfCâtTimp
    returnează m
SfSubalgoritm
    
```

Pentru care dintre următoarele apeluri corpul buclei **CâtTimp** se va executa cel mult o dată?

- A. $f(10, 11)$
- B. $f(10, 10)$
- C. $f(10, 9)$
- D. $f(10, 15)$

16. Se consideră subalgoritmul $f(a, b)$, unde a și b sunt numere naturale ($1 \leq a, b \leq 10000$).

```

Subalgoritm f(a, b):
    c ← 0
    d ← 0
    p ← 1
    CâtTimp a + b + c > 0 execută
        c ← a MOD 10 + b MOD 10 + c
        d ← d + (c MOD 10) * p
        p ← p * 10
        a ← a DIV 10
        b ← b DIV 10
        c ← c DIV 10
    SfCâtTimp
    returnează d
SfSubalgoritm
    
```

Ce va returna apelul $f(493, 1836)$?

- A. 2329
- B. 2229
- C. 2430
- D. 3292

17. Se consideră subalgoritmul $afisare(M, n)$ care primește ca și parametru un șir M cu n ($n \leq 10$) numere întregi ($M[1], M[2], \dots, M[n]$) reprezentând o mulțime.

```

Subalgoritm afisare(M, n):
    nr ← 2^n
    k ← 0
    CâtTimp k < nr execută
        curent ← k
        Scrie '{'
        Pentru j = 1, n execută
            r ← curent MOD 2
            curent ← curent DIV 2
            Dacă r = 1 atunci
                Scrie M[j]
            SfDacă
        SfPentru
        Scrie '}'
        Scrie linie nouă
        k ← k + 1
    SfCâtTimp
SfSubalgoritm
    
```

Precizați care dintre următoarele afirmații sunt adevărate:

- A. Subalgoritmul afișează toate permutările mulțimii M .
- B. Subalgoritmul afișează toate combinațiile elementelor mulțimii M luate câte $i, i = 0, 1, \dots, n$ (nu neapărat în această ordine).
- C. Subalgoritmul afișează toate aranjamentele elementelor mulțimii M luate câte $i, i = 0, 1, \dots, n$ (nu neapărat în această ordine).
- D. Subalgoritmul afișează toate submulțimile mulțimii M .

18. Se dă subalgoritmul $s(a, b, c)$, unde a, b, c sunt numere naturale pozitive ($1 \leq a, b, c \leq 10000$).

```
Subalgoritm s(a, b, c):
  Dacă (a = 1) SAU (b = 1) SAU (c = 1) atunci
    returnează 1
  altfel
    Dacă a > b atunci
      returnează a * s(a - 1, b, c)
    altfel
      Dacă a < b atunci
        returnează b * s(a, b - 1, c)
      altfel
        returnează c * s(a - 1, b - 1, c - 1)
  SfDacă
SfDacă
SfSubalgoritm
```

Precizați care dintre următoarele afirmații sunt adevărate în cazul în care $a = b$ și $a < c$:

- A. Subalgoritmul calculează și returnează $c!$
- B. Subalgoritmul calculează și returnează $c! / (c - a + 1)!$
- C. Subalgoritmul calculează și returnează $c! / (c - a - 1)!$
- D. Subalgoritmul calculează și returnează aranjamente de c luate câte $(a - 1)$

19. Subalgoritmul de mai jos are ca parametri de intrare un șir A cu n numere naturale ($A[1], A[2], \dots, A[n]$) și numărul natural n ($1 \leq n \leq 10000$). Pentru numerele naturale x și y , x^y semnifică x la puterea y (x^y).

```
Subalgoritm h(A, n):
  Dacă n = 0 atunci
    returnează 0
  altfel
    returnează A[n] * (-1)^(1 - A[n] MOD 2) + h(A, n - 1)
  SfDacă
SfSubalgoritm
```

Precizați care dintre următoarele afirmații sunt adevărate:

- A. Subalgoritmul returnează diferența dintre suma elementelor de pe poziții pare și suma elementelor de pe pozițiile impare din șirul A
- B. Subalgoritmul returnează diferența dintre suma elementelor pare și suma elementelor impare din șirul A
- C. Subalgoritmul returnează diferența dintre suma elementelor impare și suma elementelor pare din șirul A
- D. Niciunul din celelalte răspunsuri nu este corect

20. Un fișier Excel conține n înregistrări cu număr de ordine de la 1 la n . Aceste înregistrări trebuie copiate într-un fișier Word în care înregistrările se vor aranja în maxim r rânduri și exact c coloane pe fiecare pagină. Se garantează că valoarea lui n întotdeauna permite aranjarea pe exact c coloane.

Să notăm cu x_1, \dots, x_c numărul de înregistrări, care sunt copiate pe fiecare coloană pe o anumită pagină.

Pe prima pagină a documentului Word, datorită prezenței unui antet, numărul de rânduri este r_1 , $r_1 < r$ (numărul de rânduri prezent pe prima pagină este mai mic), adică $x_p = r_1, \forall 1 \leq p \leq c$.

Înregistrările vor fi aranjate în fișierul Word pe fiecare pagină de sus în jos pe fiecare coloană, coloanele fiind completate de la stânga la dreapta: dacă prima înregistrare de pe o pagină are numărul de ordine i , înregistrarea cu numărul de ordine $(i + 1)$ va fi prezentă sub ea, iar înregistrarea cu numărul de ordine $(i + x_1)$ va fi prima înregistrare de pe coloana 2 de pe pagina respectivă ș.a.m.d.

Pe ultima pagină a documentului Word se dorește ca pe toate coloanele numărul înregistrărilor să fie echilibrat, adică diferența dintre numărul înregistrărilor de pe oricare două coloane să fie cel mult 1 ($|x_j - x_k| \leq 1, \forall 1 \leq j, k \leq c, j \neq k$).

În cazul celorlalte pagini (în afară de prima și ultima) $x_p = r, \forall 1 \leq p \leq c$.

Pentru $n = 5883$, $r = 46$, $r_1 = 12$ și $c = 2$ pe ce rând al paginii se poate regăsi ultima înregistrare din document (cea cu număr de ordine $i = 5883$)?

- A. 29 B. 30 C. 31 D. 32

21. Se consideră subalgoritmul prelucrează(a, b, c, d, e), care primește ca parametri cinci numere întregi a, b, c, d și e ($1 \leq a, b \leq 10000, 2 \leq c \leq 16, 1 \leq d < c$).

```

Subalgoritm prelucrează( $a, b, c, d, e$ ):
    Dacă  $a = 0$  ȘI  $b = 0$  atunci
        Dacă  $e = 0$  atunci
            returnează 1
        altfel
            returnează 0
    SfDacă
    SfDacă
    Dacă  $(a \text{ MOD } c = d)$  ȘI  $(b \text{ MOD } c = d)$  atunci
        returnează prelucrează( $a \text{ DIV } c, b \text{ DIV } c, c, d, e$ )
    SfDacă
    Dacă  $a \text{ MOD } c = d$  atunci
        returnează prelucrează( $a \text{ DIV } c, b \text{ DIV } c, c, d, e + 1$ )
    SfDacă
    Dacă  $b \text{ MOD } c = d$  atunci
        returnează prelucrează( $a \text{ DIV } c, b \text{ DIV } c, c, d, e - 1$ )
    altfel
        returnează prelucrează( $a \text{ DIV } c, b \text{ DIV } c, c, d, e$ )
    SfDacă
SfSubalgoritm
    
```

Precizați care dintre următoarele afirmații sunt adevărate pentru apelul prelucrează($a, b, c, d, 0$) :

- A. Returnează 1 dacă reprezentările în baza c a numerelor a și b conțin cifra d de număr egal de ori, 0 în caz contrar
- B. Returnează 1 dacă cifra d apare în reprezentarea în baza c a numărului a și în reprezentarea în baza c a numărului b , 0 în caz contrar
- C. Returnează 1 dacă cifra d apare în reprezentarea în baza c a numărului a sau în reprezentarea în baza c a numărului b , 0 în caz contrar
- D. Returnează 1 dacă cifra d nu apare deloc în reprezentările în baza c a numerelor a și b , 0 în caz contrar

22. Se consideră subalgoritmii $\text{val}(p, s, i, n, x)$ și $\text{val_exp}(p, n, x)$ a căror parametri au următoarea specificație: un șir p cu n numere întregi ($p[1], p[2], \dots, p[n]$), numerele naturale s, i și n ($n \leq 1000, n = 2^k, k < 10$), și numărul real x . Valorile șirului p reprezintă coeficienții expresiei în ordine crescătoare a exponenților, exponentul maxim fiind egal cu $n - 1$, într-o expresie de forma $p[1] + p[2] \cdot x + p[3] \cdot x^2 + \dots + p[n] \cdot x^{n-1}$

Exemplu: $p = [1, 2, 3, 4]$ corespunde expresiei $E(x) = 1 + 2x + 3x^2 + 4x^3$.

```

Subalgoritm val(p, s, i, n, x):
    Dacă s + i ≤ n atunci
        _____
        altfel
            returnează p[s]
    SfDacă
SfSubalgoritm

Subalgoritm val_exp (p, n, x):
    returnează val(p, 1, 1, n, x)
SfSubalgoritm

```

Care dintre următoarele variante completează corect spațiul subliniat astfel încât subalgoritmul $\text{val_exp}(p, n, x)$ să returneze valoarea expresiei $E(x)$?

- A. returnează $p[s] + x * \text{val}(p, s + i, i * 2, n, x * x)$
- B. returnează $\text{val}(p, s, i * 2, n - i, x * x) + x * \text{val}(p, s + i, i * 2, n, x * x)$
- C. returnează $\text{val}(p, s + i, i * 2, n, x * x) + x * \text{val}(p, s, i * 2, n - i, x * x)$
- D. returnează $p[s] + x * \text{val}(p, s + i, i, n, x)$

23. Se consideră subalgoritmul $f(a)$, care primește ca și parametru un număr natural a ($2 \leq a < 1000000$) și returnează *adevărat* dacă există un număr natural d , $1 < d < a$ cu proprietatea că d divide a , și *fals* în caz contrar. Notația $[x]$ reprezintă partea întreagă a numărului x .

Care dintre variantele următoare ale subalgoritmului $f(a)$ sunt corecte?

A.

```

Subalgoritm f(a):
    Dacă a = 2 atunci
        returnează fals
    SfDacă
    Dacă a MOD 2 = 0 atunci
        returnează adevărat
    SfDacă
    Pentru d ← 3, [√a] - 1, 2 execută
        Dacă a MOD d = 0 atunci
            returnează adevărat
    SfDacă
    SfPentru
    returnează fals
SfSubalgoritm

```

B.

```

Subalgoritm f(a):
    Pentru d ← 2, [√a] execută
        Dacă a MOD d = 0 atunci
            returnează adevărat
    SfDacă
    SfPentru
    returnează fals
SfSubalgoritm

```

C.

```

Subalgoritm f(a):
    Dacă a ≤ 2 atunci
        returnează fals
    SfDacă
    Dacă a MOD 2 = 0 atunci
        returnează adevărat
    SfDacă
    Pentru d ← 3, [√a], 2 execută
        Dacă a MOD d = 0 atunci
            returnează adevărat
    SfDacă
    SfPentru
    returnează fals
SfSubalgoritm

```

D.

```

Subalgoritm f(a):
    d ← a - 1
    CâtTimp adevărat execută
        Dacă a MOD d = 0 atunci
            returnează adevărat
    SfDacă
    d ← d - 1
    SfCâtTimp
    returnează fals
SfSubalgoritm

```

24. Fie expresia de mai jos, unde $1 < A < 2021$ și $1 < n < 10202110$.

$$E(A, n) = (A + A^2 + A^3 + \dots + A^n) \text{ MOD } 2021$$

Care dintre următorii subalgoritmi calculează corect valoarea $E(A, n)$ și are complexitatea timp specificată?

Presupuneți că toate calculele se realizează pe tipuri de date pe 32 de biți. Presupuneți că x^k se calculează în $O(\log k)$.

A.

```
Subalgoritm E(A, n):
    returnează (A * (A^n - 1) DIV (A - 1)) MOD 2021
SfSubalgoritm
```

Complexitatea timp: $O(\log n)$

B.

```
Subalgoritm E(A, n):
    returnează ((A * (A^n - 1)) MOD 2021) DIV ((A - 1) MOD 2021)
SfSubalgoritm
```

Complexitatea timp: $O(\log n)$

C.

```
Subalgoritm E1(A, n):
    Dacă n = 1 atunci
        returnează (A, A) //returnează o pereche de valori
    SfDacă
    Dacă n MOD 2 = 1 atunci
        (t1, t2) ← E1(A, n - 1)
        p ← (t1 * A) MOD 2021
        returnează (p, (p + t2) MOD 2021)
    altfel
        (t1, t2) ← E1(A, n DIV 2)
        p ← (t1 * t1) MOD 2021
        returnează (p, ((1 + t1) * t2) MOD 2021)
    SfDacă
SfSubalgoritm

Subalgoritm E(A, n):
    (aux1, aux2) ← E1(A, n)
    returnează aux2
SfSubalgoritm
```

Complexitatea timp: $O(\log n)$

D.

```
Subalgoritm E(A, n):
    raspuns ← A
    Pentru i = 2, n execută
        raspuns ← raspuns + A^i
    SfPentru

    returnează raspuns MOD 2021
SfSubalgoritm
```

Complexitatea: $O(n \cdot \log n)$

25. Pe un cerc se scriu, în ordine crescătoare, toate numerele de la 1 la 1000, în sensul acelor de ceasornic. Începând de la 1, colorăm, în sensul acelor de ceasornic, fiecare al k -lea număr ($1, k + 1, 2 \cdot k + 1, \dots$). Procedul se continuă până când ajunge la un număr deja colorat, fiind colorate la final x numere. Care dintre următoarele afirmații sunt adevărate?

- A. Dacă $k = 15$ atunci $x = 300$
- B. Dacă $k = 45$ atunci $x = 200$
- C. Dacă $k = 25$ atunci $x = 40$
- D. Dacă $k = 30$ atunci $x = 150$

26. Se consideră subalgoritmul $ceFace(n, k)$ unde n și k sunt numere naturale ($1 \leq n, k \leq 1000000$).

```

Subalgoritm ceFace(n, k):
    nr ← 0
    p ← 1
    CâtTimp (n ≠ 0) ȘI (k ≠ 0) execută
        Dacă n MOD 2 ≠ 0 atunci
            nr ← nr + ((n DIV 10) MOD 10) * p
            p ← p * 10
        altfel
            k ← k - 1
    SfDacă
    n ← n DIV 10
    SfCâtTimp
    returnează nr
SfSubalgoritm

```

Care dintre următoarele perechi de apeluri returnează valori identice?

- A. $ceFace(32345, 3)$ și $ceFace(321458, 7)$
- B. $ceFace(321458, 4)$ și $ceFace(2314587, 4)$
- C. $ceFace(2314, 3)$ și $ceFace(23145, 4)$
- D. $ceFace(23145, 3)$ și $ceFace(231458, 4)$

27. Se consideră subalgoritmii:

- $putere(b, p)$ – determină b^p (b la puterea p), b, p - numere naturale ($1 \leq b \leq 20, 1 \leq p \leq 20$);
- $nrCifre(nr)$ – returnează numărul cifrelor unui număr natural nenul nr ($0 < nr \leq 1000000$), sau valoarea 0 atunci când $nr = 0$;
- $produs(st, dr)$ – subalgoritmul de mai jos, unde st, dr – numere naturale ($100 < st < 1000000, 0 \leq dr < 1000000, st$ – număr care, reprezentat în baza 10, are cel puțin două cifre nenule).

```

Subalgoritm produs(st, dr):
    Dacă st > 0 atunci
        drCrt ← _____
        stCrt ← st DIV 10
        Dacă st * dr < stCrt * drCrt atunci
            returnează produs(stCrt, drCrt)
        altfel
            returnează st * dr
    SfDacă
    altfel
        returnează st * dr
    SfDacă
SfSubalgoritm

```

Care dintre următoarele variante completează corect spațiul subliniat astfel încât subalgoritmul $produs(st, dr)$ prin executarea secvenței de instrucțiuni

```

scrie produs(1092, 0)
scrie produs(75981, 0)

```

să se afișeze 920 și 73575?

- A. $(st \text{ MOD } 10) * putere(10, nrCifre(dr)) + dr$
- B. $(st \text{ MOD } 10) * putere(10, dr) + dr$
- C. $(st \text{ MOD } 10) * putere(10, nrCifre(dr))$
- D. $(st \text{ MOD } 10) * nrCifre(dr)$

28. Se consideră subalgoritmul $ceFace(a, n, i, f)$, care primește ca parametru un șir a cu n numere întregi ($a[1], a[2], \dots, a[n]$) și numerele întregi i, f și n ($2 \leq n \leq 10000$).

```

Subalgoritm ceFace(a, n, i, f):
    Dacă (i = n) ȘI (f = 2) atunci
        returnează ADEVĂRAT
    altfel
        Dacă (i = n) atunci
            returnează FALS
        altfel
            Dacă (f ≤ 1) ȘI (a[i] < a[i + 1]) atunci
                returnează ceFace(a, n, i + 1, 1)
            SfDacă
            Dacă (1 ≤ f) ȘI (a[i] > a[i + 1]) atunci
                returnează ceFace(a, n, i + 1, 2)
            SfDacă
            returnează FALS
        SfDacă
    SfDacă
SfSubalgoritm
    
```

Precizați care dintre următoarele afirmații sunt adevărate, considerând apelul inițial $ceFace(a, n, 1, 0)$.

- Subalgoritmul returnează *adevărat* dacă și numai dacă maximul șirului a se află pe o poziție i , $1 < i < n$.
- Subalgoritmul returnează *adevărat* dacă și numai dacă $\exists k$, ($1 < k < n$), astfel încât $a[1] < a[2] < \dots < a[k] > a[k + 1] > \dots > a[n]$.
- Subalgoritmul returnează *fals* dacă șirul a este strict crescător.
- Subalgoritmul returnează *adevărat* dacă și numai dacă $\exists k$, ($1 < k < n$), astfel încât $a[k] > a[k + 1] > \dots > a[n]$.

29. Fie următorul subalgoritm, având ca parametru numărul natural nenul n și care returnează un număr natural.

```

Subalgoritm f(n):
    j ← n
    CâtTimp j > 1 execută
        i ← 1
        CâtTimp i ≤ n4 execută
            i ← 4 * i
        SfCâtTimp
        j ← j DIV 2
    SfCâtTimp
    returnează j
SfSubalgoritm
    
```

În care dintre următoarele clase de complexitate se încadrează complexitatea timp a algoritmului?

- $O(\log_2 n^2)$
- $O(\log_2^2 n^2)$
- $O(\log_4^2 n)$
- $O(\log_2 \log_4 n)$

30. Se dă un șir s de n caractere din alfabetul englez, ($s[1], s[2], \dots, s[n]$). Dorim să aflăm cel mai lung sufix al său care este palindrom. Un sufix al unui șir de caractere este o subsecvență a șirului care conține ultimul caracter. De exemplu, pentru șirul $abab$, cel mai lung sufix palindrom al său este bab .

Presupunem că avem definit următorul subalgoritm:

- $ascii(c)$ - returnează codul ASCII al caracterului c .

Presupunem că operațiile aritmetice nu produc depășire pe mulțimea numerelor întregi.

Care dintre următoarele implementări returnează lungimea acestui sufix la apelul $sufix(s, n)$?

A.
Subalgoritm sufix(s, n):
 hf ← 0
 hb ← 0
 raspuns ← 1
Pentru i ← n, 1, -1 **execută**
 hf ← ascii(s[i]) + 2021 * hf
 hb ← hb + ascii(s[i]) * 2021^{n - i}

Dacă hf = hb **atunci**
 raspuns ← n - i + 1
SfDacă
SfPentru
returnează raspuns
SfSubalgoritm

B.
Subalgoritm sufix(s, n):
 hf ← 0
 hb ← 0
 raspuns ← 1
Pentru i ← n, 1, -1 **execută**
 hf ← ascii(s[i]) + 3 * hf
 hb ← hb + ascii(s[i]) * 3^{n - i}

Dacă hf = hb **atunci**
 raspuns ← n - i + 1
SfDacă
SfPentru
returnează raspuns
SfSubalgoritm

C.
Subalgoritm sufix(s, n):
 hf ← 0
 hb ← 0
 raspuns ← 1
Pentru i ← n, 1, -1 **execută**
 hf ← ascii(s[i]) + 2021 * hb
 hb ← hf + ascii(s[i]) * 2021^{n - i}

Dacă hf = hb **atunci**
 raspuns ← n - i + 1
SfDacă
SfPentru
returnează raspuns
SfSubalgoritm

D. Niciuna dintre celelalte variante nu este corectă.

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

Concurs de Admitere 9 septembrie 2021
Proba scrisă la INFORMATICĂ
BAREM ȘI REZOLVARE

OFICIU: 10 puncte

1	C	3 puncte
2	B	3 puncte
3	A	3 puncte
4	B, D	3 puncte
5	B	3 puncte
6	D	3 puncte
7	A, B, C	3 puncte
8	B	3 puncte
9	D	3 puncte
10	C	3 puncte
11	A, C	3 puncte
12	A	3 puncte
13	A, B	3 puncte
14	A, C, D	3 puncte
15	A, B	3 puncte
16	A	3 puncte
17	B, D	3 puncte
18	B, D	3 puncte
19	C	3 puncte
20	C, D	3 puncte
21	A	3 puncte
22	B, D	3 puncte
23	B, C	3 puncte
24	C	3 puncte
25	B, C	3 puncte
26	A, D	3 puncte
27	A	3 puncte
28	B, C	3 puncte
29	B, C	3 puncte
30	A	3 puncte