

Generarea elementelor combinatoriale.

Backtracking

13 Decembrie 2025

Elemente combinatoriale:

- submulțimi
- combinări
- aranjamente
- permutări

Backtracking

- soluția este reprezentată sub forma unui tablou: $x[1], \dots, x[n]$, unde $x[k] \in A_k$

Pentru fiecare $k \in \{1, \dots, n\}$:

- $x[k]$ primește pe rând valori din mulțimea A_k
- dacă valoarea curentă $x[k]$ este invalidă (nu poate duce la o soluție finală), elementul $x[k]$ primește următoarea valoare posibilă din A_k sau, dacă toate valorile din A_k au fost epuizate, revenim la elementul anterior, $x[k-1]$ și alegem o nouă valoare pentru acesta
- dacă avem o soluție parțială validă, avem următoarele cazuri:
 - dacă vectorul curent x reprezintă o soluție finală, îl afișăm
 - dacă nu am obținut încă o soluție finală, avansăm la următorul element, $x[k+1]$, și reluăm procesul pentru acest element

Problema 22 - Admitere septembrie 2023

22. Se consideră algoritmul $\text{ceFace}(n, k, x, p)$, unde n, k și p sunt numere naturale nenule ($1 \leq n, k, p \leq 10, p \leq n$), iar x este un vector cu $p + 1$ elemente numere naturale ($x[0], x[1], \dots, x[p]$). Presupunem că $x[0]$ este inițializat cu 0.

```
Algorithm ceFace(n, k, x, p):
  If k > p then
    For i ← 1, p execute
      Write x[i]
    EndFor
    Write " " //un singur spațiu
  Else
    For i ← x[k - 1] + 1, n execute
      x[k] ← i
      ceFace(n, k + 1, x, p)
    EndFor
  EndIf
EndAlgorithm
```

Precizați care dintre următoarele variante de răspuns sunt corecte.

- A. După ce algoritmul se apelează sub forma $\text{ceFace}(3, 1, x, 3)$ acesta se va mai autoapela de 6 ori.
- B. Dacă $x[0]$ se inițializează cu o valoare diferită de 0, în urma apelului $\text{ceFace}(5, 1, x, 3)$ numărul de spații afișate este diferit de 10.
- C. Dacă algoritmul se apelează sub forma $\text{ceFace}(5, 1, x, 4)$ se afișează numerele 1245 1234 1345 1235 2345, dar în altă ordine.
- D. Dacă algoritmul se apelează sub forma $\text{ceFace}(5, 1, x, 3)$ rezultatul afișat este 123 124 125 134 135 145 234 235 în această ordine.

Explicații:

- dacă $x[0]=0$, algoritmul generează combinații de n luate câte p folosind elemente din mulțimea $\{1, \dots, n\}$, în ordine lexicografică ($x[k]$ ia valori de la $x[k-1]+1$ până la n)
- varianta A nu este corectă fiindcă funcția se va autoapela de 7 ori:

$\text{ceFace}(3, 1, x, 3)$:

- $x[1] \leftarrow 1, \text{ceFace}(3, 2, x, 3)$
 - $x[2] \leftarrow 2, \text{ceFace}(3, 3, x, 3)$
 - $x[3] \leftarrow 3, \text{ceFace}(3, 4, x, 3)$
 - $x[2] \leftarrow 3, \text{ceFace}(3, 3, x, 3)$
 - $x[1] \leftarrow 2, \text{ceFace}(3, 2, x, 3)$
 - $x[2] \leftarrow 3, \text{ceFace}(3, 3, x, 3)$
 - $x[1] \leftarrow 3, \text{ceFace}(3, 2, x, 3)$
- varianta B este corectă. Numărul de spații afișate este egal cu numărul de soluții afișate. Numărul de soluții al apelului $\text{ceFace}(5, 1, x, 3)$ este 10 dacă și numai dacă se generează toate combinațiile de 5 elemente luate câte 3. Valoare lui $x[0]$ indică mulțimea de valori posibile pentru elementele șirului: $\{x[0]+1, \dots, n\}$. Dacă $x[0] > 0 \Rightarrow$ se vor afișa mai puțin de 10 soluții. Dacă $x[0] < 0$, mai multe
 - varianta C este corectă (ordinea este lexicografică)
 - varianta D nu este corectă fiindcă lipsesc soluțiile 245, 345 (în total algoritmul afișează 10 soluții)

Răspuns: B, C

Problema 15 - Admitere iulie 2024

15. Pentru a determina toate submulțimile mulțimii $A = \{4, 8, 9, 12, 15\}$ cu 5 elemente, un elev a scris algoritmul `generare(i, n, x, A)`. Mulțimea este reprezentată prin vectorul A cu n elemente numere naturale. Submulțimile generate se afișează cu ajutorul algoritmului `afis(m, x, A)`, x fiind un vector auxiliar indexat de la 0 iar m un număr natural reprezentând lungimea vectorului x curent. Înainte de apelul `generare(1, 5, x, A)` elementul $x[0]$ a fost inițializat cu 0.

```
Algorithm genereare(i, n, x, A):  
  For j ← n, x[i - 1] + 1, -1 execute  
    x[i] ← j  
    afis(i, x, A)  
    genereare(i + 1, n, x, A)  
  EndFor  
EndAlgorithm
```

```
Algorithm afis(m, x, A):  
  Write "{", a[x[1]]  
  For i ← 2, m execute  
    Write ", ", a[x[i]]  
  EndFor  
  Write "}", newline  
EndAlgorithm
```

Știind că primele 4 submulțimi afișate sunt, în această ordine: $\{15\}$, $\{12\}$, $\{12, 15\}$, $\{9\}$ care va fi a 8-a submulțime generată (submulțimea vidă nu se ia în considerare)?

A. $\{9, 12\}$

B. $\{8\}$

C. $\{9, 12, 15\}$

D. $\{8, 15\}$

Explicații:

- în vectorul x se generează indici din vectorul (mulțimea) A , elementele de pe fiecare poziție i luând valori de la n până la $x[i-1]+1$
- atunci când, începând cu poziția i , nu mai putem completa o soluție x cu alte valori mai mari decât cele completate până la poziția $i-1$, revenim la poziția $i-1$ și decrementăm $x[i-1]$
- următoarele submulțimi generate vor fi: $\{9, 15\}$, $\{9, 12\}$, $\{9, 12, 15\}$, $\{8\}$

Răspuns: B

Temă: Care vor fi următoarele 4 submulțimi afișate?

Problema 21 - Admitere iulie 2025

21. Se consideră algoritmul $f(A, B, m, n, k)$, unde m, n și k sunt numere naturale ($1 \leq m, n, k \leq 100$), iar A este un vector cu n elemente numere întregi ($A[1], A[2], \dots, A[n]$, unde $-100 \leq A[i] \leq 100$, pentru $i = 1, 2, \dots, n$), iar B este un vector cu n elemente, toate egale cu zero.

```
Algorithm f(A, B, m, n, k):
    aux ← 0
    For j ← 1, k - 1 execute
        aux ← aux + B[j] * A[j]
    EndFor
    If aux = m then
        Write "Solutie: "
        For j ← 1, k - 1 execute
            If B[j] = 1 then
                Write A[j], " "
            EndIf
        EndFor
        Write new line
    Else
        If k ≠ n + 1 then
            For i ← 0, 1 execute
                B[k] ← i
                f(A, B, m, n, k + 1)
            EndFor
        EndIf
    EndIf
EndAlgorithm
```

Dacă $A = [4, 5, 8, 9, 3, 12, 15, 7]$, în urma apelului $f(A, B, 12, 8, 1)$ prima soluție afișată va fi Solutie: 12.

Care va fi a doua și a treia soluție afișată?

- A.
Solutie: 4 8
Solutie: 4 5 3
- B.
Solutie: 4 8
Solutie: 5 7
- C.
Solutie: 9 3
Solutie: 5 7
- D.
Solutie: 9 3
Solutie: 4 8

Explicații:

- în vectorul B se generează pe fiecare poziție 0 sau 1 $\Rightarrow$ vectorul B reprezintă o mască/vector caracteristic ce indică ce elemente din A au fost selectate $\Rightarrow$ se generează submulțimi și se afișează o soluție dacă suma elementelor din acea submulțime este egală cu m
- valorile din B se generează în ordine lexicografică
- condiția : $aux=m$ oprește continuarea generării unei soluții de la poziția k dacă suma obținută folosind configurația curentă este m

Răspuns: C

Temă: 1) Care sunt restul soluțiilor afișate pentru datele de intrare din enunț?

2) Ce se afișează dacă $A=[4, 1, 2, 3, -2]$ în urma apelului $f(A, B, 4, 5, 1)$?

Problema 21 - Admitere iulie 2023

21. Se consideră algoritmul $p(x, n, k, final)$, unde x este un vector de $n + 1$ numere naturale ($x[0], x[1], x[2], \dots, x[n]$). Inițial $x[i] = 0$, pentru $i = 0, 1, 2, \dots, n$. Variabilele n și k sunt numere naturale nenule ($1 \leq n, k \leq 20$), iar $final$ este de tip boolean. Algoritmul $Afis(x, 1, n)$ afișează elementele $x[1], x[2], \dots, x[n]$.

```
Algorithm p(x, n, k, final):
  While final = False execute
    While x[k] < n execute
      x[k] ← x[k] + 1
      If OK(x, k) = True then
        If k = n then
          Afis(x, 1, n)
        Else
          k ← k + 1
          x[k] ← 0
        EndIf
      EndIf
    EndWhile
    _____ // aici trebuie completat algoritmul
  EndWhile
EndAlgorithm
```

```
Algorithm OK(x, k):
  For i ← 1, k - 1 execute
    If x[k] = x[i] then
      Return False
    EndIf
  EndFor
  Return True
EndAlgorithm
```

Cu ce secvență de cod trebuie completat algoritmul, astfel încât în urma apelului $p(x, n, 1, False)$ să se afișeze toate permutările de ordin n , fiecare o singură dată.

- | | | | |
|---|---|--------------------|---|
| A.
If k > 1 then
k ← k - 1
Else
final ← True
EndIf | B.
If k > 0 then
k ← k - 1
Else
final ← True
EndIf | C.
final ← True | D.
If k > 1 then
k ← k - 1
final ← True
EndIf |
|---|---|--------------------|---|

Explicații

- varianta B afișează toate permutările de ordin n de $n+1$ ori deoarece funcția OK verifică doar ca $x[k]$ să fie diferit de elementele de pe pozițiile $1, \dots, k-1$; astfel, poziția 0 se va completa cu toate valorile de la 0 la n
- variantele C și D afișează doar prima permutare

Răspuns: A

Problema 18 - Admitere iulie 2024

18. Se consideră algoritmul $\text{ceFace}(n)$, unde n este număr natural nenul ($1 \leq n < 10^3$).

```
Algorithm ceFace(n):
    Return ceFaceRecurziv(n, 1, 1)
EndAlgorithm

Algorithm ceFaceRecurziv(n, a, b):
    If n = 0 then
        Return 1
    Else
        If n < 0 OR b > n then
            Return 0
        Else
            Return ceFaceRecurziv(n, a + b, a) + ceFaceRecurziv(n - a, a + b, a)
        EndIf
    EndIf
EndAlgorithm
```

Care dintre următoarele afirmații sunt adevărate?

- A. În intervalul $[11, 16]$ există o singură valoare x , pentru care algoritmul $\text{ceFace}(x)$ returnează 1.
- B. Pentru orice număr n , algoritmul $\text{ceFace}(n)$ va returna valoarea 0 sau 1.
- C. Algoritmul $\text{ceFace}(n)$ returnează numărul de moduri de a scrie numărul n ca sumă de numere consecutive.
- D. Algoritmul $\text{ceFace}(n)$ returnează numărul de mulțimi diferite ale căror elemente sunt numere Fibonacci diferite de 0 și care au suma egală cu n .

Explicații:

- se calculează numărul de moduri de a-1 scrie pe n ca sumă de numere Fibonacci distincte (și diferite de 0) $\Rightarrow$ varianta D este corectă
- varianta A este corectă (sunt colorate cu albastru scrierile sub forma de suma de numere Fibonacci) - doar pentru 12 există o singură descompunere:

$$11 = 3 + 8 = 1 + 2 + 8 \Rightarrow 3 \text{ moduri}$$

$$12 = 1 + 3 + 8$$

$$13 = 5 + 8 = 2 + 3 + 8 \Rightarrow 3 \text{ moduri}$$

$$14 = 1 + 5 + 8 = 1 + 2 + 3 + 8 = 1 + 13 \Rightarrow 3 \text{ moduri}$$

$$15 = 2 + 5 + 8 = 2 + 13 \Rightarrow 2 \text{ moduri}$$

$$16 = 1 + 2 + 5 + 8 = 1 + 2 + 13 = 3 + 5 + 8 = 3 + 13 \Rightarrow 4 \text{ moduri}$$

- varianta B este incorectă (de exemplu, pentru 11 se va returna 3), C este incorectă (nu se generează numere consecutive).

Răspuns: A, D

Problema 24 - Admitere iulie 2024

24. Se consideră algoritmul $\text{calculeaza}(v, b, n, i)$, unde b, n, i sunt numere naturale nenule ($1 \leq b, n, i \leq 10^3$), iar v este un vector cu n elemente numere naturale ($v[1], v[2], \dots, v[n]$, $0 \leq v[i] \leq 10^3$, pentru $i = 1, 2, \dots, n$):

Algorithm $\text{calculeaza}(v, b, n, i)$:

If $b = 0$ **then**

Return *True*

EndIf

If $i = n$ **then**

Return *False*

EndIf

Return $\text{calculeaza}(v, b - v[i], n, i + 1)$ **OR** $\text{calculeaza}(v, b, n, i + 1)$

EndAlgorithm

Pentru care din următoarele date de intrare algoritmul returnează *True*?

A. $v = [3, 1, 7, 4, 2]$, $b = 10$, $n = 5$, $i = 1$

B. $v = [2, 6, 4, 8, 12]$, $b = 12$, $n = 5$, $i = 1$

C. $v = [3, 1, 7, 4, 2]$, $b = 10$, $n = 5$, $i = 2$

D. $v = [2, 6, 4, 8, 12]$, $b = 12$, $n = 5$, $i = 3$

Explicații:

- se verifică dacă b poate fi scris ca sumă de elemente de pe poziții distincte din intervalul $[i_start, n-1]$ din vectorul v , unde i_start este valoarea inițială a variabilei i (se returnează *True* dacă putem scădea succesiv elemente din b astfel încât să obținem rezultatul 0)
- varianta A este corectă: $10=3+7$
- varianta B este corectă: $12=4+8$
- varianta C este incorectă fiindcă putem folosi doar elementele $\{1, 7, 4\}$ și nu putem forma suma 10 cu ele
- varianta D este corectă: $12=4+8$

Răspuns: A, B, D

Problema 21 - Admitere septembrie 2024

21. Se consideră algoritmul `ceFace(arr, n)` unde **arr** este un vector cu n ($1 \leq n \leq 100$) elemente numere întregi (**arr**[1], **arr**[2], ..., **arr**[**n**], unde $-10^5 \leq \text{arr}[i] \leq 10^5$ pentru $i = 1, 2, \dots, n$).

```
Algorithm ceFace(arr, n):  
    sum ← 0  
    For i ← 1, n execute  
        sum ← sum + arr[i]  
    EndFor  
    If sum MOD 2 ≠ 0 then  
        Return False  
    EndIf  
    Return auxiliar(arr, n, sum DIV 2)  
EndAlgorithm
```

```
Algorithm auxiliar(arr, n, sum):  
    If sum = 0 then  
        Return True  
    EndIf  
    If n = 1 AND sum ≠ 0 then  
        Return False  
    EndIf  
    If arr[n - 1] > sum then  
        Return auxiliar(arr, n - 1, sum)  
    EndIf  
    Return auxiliar(arr, n - 1, sum) OR  
        auxiliar(arr, n - 1, sum - arr[n - 1])  
EndAlgorithm
```

Care din următoarele afirmații sunt adevărate?

- A. Apelul `ceFace([11, 5, 6, 22, 0, 7, 6, 13], 8)` returnează *True*.
- B. Apelul `ceFace([-5, -6, -22, -7, -6, -13], 6)` **NU** returnează *True*.
- C. Dacă vectorul **arr** conține doar valori negative, algoritmul `auxiliar(arr, n, sum)` va intra în ciclu infinit.
- D. Dacă și numai dacă elementele din **arr** pot fi partiționate în două mulțimi astfel încât media elementelor din cele două mulțimi să fie egală, algoritmul `ceFace(arr, n)` returnează *True*.

Explicații:

- algoritmul calculează într-o primă etapă suma elementelor din vectorul **arr**
- se efectuează apelul funcției `auxiliar` cu această sumă / 2 dacă suma este pară
- dacă **arr** conține doar numere pozitive (cum este cazul în exemplul A din grilă), se returnează *True* dacă există elemente din **arr** având suma egală cu **sum**, adică dacă elementele din **arr** pot fi partiționate în 2 submulțimi de sumă egală
- $7 + 6 + 5 + 6 + 11 = 13 + 22 + 0 \Rightarrow$ varianta A este corectă
- apelul `ceFace([-5, -6, -22, -7, -6, -13], 6)` returnează *False* (suma este impară) $\Rightarrow$ varianta B este corectă
- nu se va intra în ciclu infinit datorită condițiilor de oprire $\Rightarrow$ C este incorectă
- algoritmul poate returna *True* dacă suma este egală, fără ca media să fie egală

Răspuns: A, B

Temă: Ce se returnează pentru apelul `ceFace([1, -5, 8, 2], 4)`? De ce?

Problema 23 - concurs Mate-Info 2024

23. Se consideră algoritmul $\text{maiMare}(n)$ unde n este număr natural nenul ($1 \leq n < 10^6$) alcătuit din cifre distincte.

Algoritmul ar trebui să returneze numărul numerelor strict mai mari ca n , formate din cifrele lui n .

De exemplu, $\text{maiMare}(213) = 3$.

Presupunem că n nu are cifre 0 la început și că avem următorii algoritmi implementați conform specificațiilor:

- $\text{factorial}(n)$ – returnează factorialul numărului natural n ($1 \leq n \leq 10$)
- $\text{nrCifre}(n)$ – returnează numărul cifrelor numărului natural n ($1 \leq n < 10^6$)
- $\text{imparte}(n)$ – returnează un vector având ca elemente cifrele numărului natural n ($1 \leq n < 10^6$), în ordine inversă. De exemplu: $\text{imparte}(1352)$ returnează vectorul $[2, 5, 3, 1]$.

```
Algorithm maiMare(n):  
    cifre ← imparte(n)  
    nrCif ← nrCifre(n)  
    Return calculeaza(cifre, nrCif)  
EndAlgorithm
```

```
1. Algorithm calculeaza(v, n):  
2.   If n < 2 then  
3.       Return 0  
4.   EndIf  
5.   mm ← 0  
6.   For i ← 1, n - 1 execute  
7.       If v[i] > v[n] then  
8.           mm ← mm + 1  
9.       EndIf  
10.  EndFor  
11.  ...  
12. EndAlgorithm
```

Care dintre următoarele instrucțiuni trebuie scrisă la linia 11 a algoritmului $\text{calculeaza}(v, n)$?

- A. **Return** $\text{factorial}(n) - ((n - mm - 1) * \text{factorial}(n - 1) + \text{calculeaza}(v, n - 1))$
- B. **Return** $\text{calculeaza}(v, n - 1) * mm + \text{factorial}(n - 1)$
- C. **Return** $(mm * \text{factorial}(n) + \text{calculeaza}(v, n - 1)) \text{ DIV } n$
- D. **Return** $\text{calculeaza}(v, n - 1) + mm * \text{factorial}(n - 1)$

Notăm cu $N = a_1 \dots a_n$ numărul dat și cu n numărul de cifre ale lui N .

mm reprezintă numărul de cifre mai mari decât a_1 (prima cifră a numărului N)

$$a_1 \dots a_n < b_1 \dots b_n \Leftrightarrow a_1 < b_1 \text{ (1) sau } a_1 = b_1 \text{ și } a_2 \dots a_n < b_2 \dots b_n \text{ (2)}$$

Trebuie să calculăm numărul numerelor care verifică proprietatea (1) și numărul numerelor care verifică proprietatea (2), ținând cont de faptul că $(b_1, \dots, b_n)$ reprezintă o permutare a cifrelor $(a_1, \dots, a_n)$

$$(1) a_1 < b_1$$

b_1 poate lua mm valori, iar pentru fiecare valoare a lui b_1 , restul cifrelor $b_2 \dots b_n$ pot fi obținute prin permutarea celor $n-1$ cifre rămase după alegerea lui $b_1 \Rightarrow$

$mm * \text{factorial}(n-1)$ numere

$$(2) a_1 = b_1 \text{ și } a_2 \dots a_n < b_2 \dots b_n$$

acest număr se poate calcula recursiv, prin apelul: $\text{calculeaza}(v, n-1)$ deoarece cifrele $b_2, \dots, b_n$ vor fi alese din cele $n-1$ cifre rămase (vor reprezenta o permutare a cifrelor

$a_2, \dots, a_n$ care se află pe pozițiile $n-1, \dots, 1$ în vectorul v)

Răspuns: D

Problema 731 - Culegerea de probleme și teste pregătitoare

731. Se dă următorul algoritm $C(n, k)$, unde n, k sunt numere naturale.

```
Algorithm C(n, k)
  If n = k or k = 0 then
    Return 1
  EndIf
  Return C(n - 1, k - 1) + C(n - 1, k)
EndAlgorithm
```

Pentru $n = 10, k = 7$, de câte ori se autoapelează funcția C ?

A. 120

B. 142

C. 238

D. 207

Observăm că nu vom putea număra efectiv numărul de autoapeluri pentru aceste date de intrare, dar putem scrie o recurență care să descrie acest număr:

$$nrAp(n, n) = nrAp(n, 0) = 0$$

$$nrAp(n, k) = 2 + nrAp(n - 1, k - 1) + nrAp(n - 1, k), \text{ altfel}$$

Observăm că această recurență seamănă cu formula de recurență a combinărilor, motiv pentru care putem încerca să deducem formula pentru $nrAp$ ca o funcție liniară a numărului de combinări:

$$nrAp(n, k) = \alpha C_n^k + \beta$$

Înlocuim în relația de recurență pentru a obține α și β

$$\alpha C_n^k + \beta = 2 + \alpha C_{n-1}^{k-1} + \beta + \alpha C_{n-1}^k + \beta \Rightarrow \beta = -2$$

$$\alpha C_n^0 - 2 = 0 \Rightarrow \alpha = 2 \Rightarrow nrAp(n, k) = 2C_n^k - 2$$

$$nrAp(10, 7) = 2 * \frac{10!}{7!3!} - 2 = 238$$

Răspuns: C

Temă: Problema 21- concurs Mate-Info 2024

21. Se consideră algoritmul $f(x, n, m)$, unde n și m sunt numere naturale ($1 \leq n, m \leq 10^4$), iar x este un vector de n numere naturale ($x[1], x[2], \dots, x[n]$, $1 \leq x[i] \leq 10^4$, pentru $i = 1, 2, \dots, n$):

Ce valoare va returna algoritmul, dacă apelul are forma $f(x, 9, 41)$, unde $x = [41, 15, 5, 8, 10, 1, 16, 18, 19]$?

- A. 1 B. 3 C. 5 D. 7

```
Algorithm f(x, n, m):
  If m = 0 then
    Return 1
  EndIf
  If n = 0 then
    Return 0
  EndIf
  If x[n] > m then
    Return f(x, n - 1, m)
  Else
    Return f(x, n - 1, m) + f(x, n - 1, m - x[n])
  EndIf
EndAlgorithm
```

Temă: Problema 23 - Admitere septembrie 2024

23. Se consideră un număr natural nenul par n ($2 \leq n \leq 12$). Dorim să generăm în șirul de caractere x toate șirurile formate din n paranteze rotunde care se deschid și se închid corect. Algoritmul $\text{paranteze}(i, \text{desc}, \text{inc}, x, n)$ se apelează sub forma $\text{paranteze}(2, 1, 0, x, n)$, știind că au avut loc inițializările $x[1] \leftarrow '('$ și $x[n] \leftarrow ')'$. Algoritmul $\text{afisare}(n, x)$ afișează șirul de caractere x de lungime n .

```
1. Algorithm paranteze(i, desc, inc, x, n):
2.   If i = n then
3.     afisare(n, x)
4.   Else
5.     If ____ then
6.       x[i] ← '('
7.       paranteze(i + 1, desc + 1, inc, x, n)
8.     EndIf
9.     If ____ then
10.      x[i] ← ')'
11.      paranteze(i + 1, desc, inc + 1, x, n)
12.    EndIf
13.  EndIf
14. EndAlgorithm
```

Cu ce ar trebui completate liniile precizate mai jos, astfel încât algoritmul să afișeze doar șirurile de caractere corecte conform cerinței?

- A. Linia 5 trebuie completată cu $\text{desc} < n$, iar linia 9 trebuie completată cu $\text{inc} < \text{desc}$
B. Linia 5 trebuie completată cu $\text{desc} < n \text{ DIV } 2$, iar linia 9 trebuie completată cu $\text{inc} < \text{desc}$
C. Linia 5 trebuie completată cu $\text{desc} < n$, iar linia 9 trebuie completată cu $\text{inc} < n \text{ DIV } 2$
D. Indiferent de comparațiile cu care se completează liniile 5 și 9, algoritmul nu va afișa toate șirurile de caractere conform cerinței.

Temă: Problema 23 - Admitere septembrie 2025

23. Pentru o partidă de fotbal între echipele **A** și **B** în care se înscriu 3 goluri și care se termină cu scorul de 2 - 1 pentru echipa **A**, golurile pot fi înscrise în 3 succesiuni:

- 0 - 0, 0 - 1, 1 - 1, 2 - 1
- 0 - 0, 1 - 0, 1 - 1, 2 - 1
- 0 - 0, 1 - 0, 2 - 0, 2 - 1

De exemplu, în succesiunea 1., partida începe de la scorul de 0 - 0, înscrie echipa **B** (0 - 1), egalează echipa **A** (1 - 1), și înscrie echipa **A** (2 - 1).

Câte succesiuni posibile există pentru o partidă cu 4 goluri?

- A. 4!
B. 16
C. 64
D. $1! + 2! + 3! + 2! + 1!$

Temă: Problema 23 - Admitere iulie 2025

23. Se consideră algoritmul `interesant(a, b, c, n)`, unde c și n sunt numere naturale ($1 \leq n \leq 100$, $1 \leq c \leq 10^4$), iar a și b sunt doi vectori de lungime n , cu elemente numere întregi ($a[1], a[2], \dots, a[n]$ și $b[1], b[2], \dots, b[n]$, $1 \leq a[i], b[i] \leq 100$, pentru $i = 1, 2, \dots, n$). Algoritmul `max(x, y)` returnează valoarea maximă dintre numerele x și y .

Algorithm `interesant(a, b, c, n):`

If $n = 0$ **OR** $c = 0$ **then**

Return 0

EndIf

If $a[n] > c$ **then**

Return `interesant(a, b, c, n - 1)`

EndIf

$x \leftarrow b[n] + \text{interesant}(a, b, c - a[n], n - 1)$

$y \leftarrow \text{interesant}(a, b, c, n - 1)$

Return `max(x, y)`

EndAlgorithm

Ce valoare se returnează în urma apelului `interesant([2, 3, 1], [6, 10, 3], 5, 3)`?

A. 13

B. 10

C. 16

D. 19

Bibliografie

1. <https://www.cs.ubbcluj.ro/probleme-si-teste-pregatitoare-de-informatica-pentru-admiterea-2026/>
2. <https://www.cs.ubbcluj.ro/admitere/nivel-licenta/subiecte-din-anii-precedenti/>
3. <https://www.pbinfo.ro/articole/16597/metoda-backtracking>