

ACADEMIA ZECELAINFO

Simulare Examen Admitere UBB 2023 - Iulie 2023

Proba scrisa la Informatica

NOTA IMPORTANTA:

In lipsa altor precizari:

- 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. Fie subalgoritmul `ce_face(x)` unde x este un numar natural nenul ($1 \leq x \leq 10^6$).

```
Subalgorithm ce_face(x):  
    s ← 0  
    cnt ← 0  
    While x > 0 execute  
        cnt ← cnt + 1  
        If cnt MOD 2 = 0 then  
            s ← s + x MOD 10  
        EndIf  
        x ← [x / 10]  
    EndWhile  
    return s  
EndSubalgorithm
```

Care dintre urmatoarele afirmatii descriu corespunzator efectul acestui subalgoritm?

- A. Algoritmul numara cate cifre pe pozitii pare contine numarul.
- B. Algoritmul calculeaza suma cifrelor aflate pe pozitii pare in numar considerand numerotarea cifrelor de la 1 incepand cu cifra unitatilor catre cifrele mai semnificative.
- C. Algoritmul calculeaza suma cifrelor aflate pe pozitii pare in numar considerand numerotarea cifrelor de la 1 incepand cu cifra cea mai semnificativa din numar si parcurgand catre cifra unitatilor.
- D. Algoritmul calculeaza suma cifrelor numarului x .

2. Se considera urmatorul subalgoritm $f(a, b)$ unde a si b sunt 2 numere naturale nenule ($1 \leq a, b \leq 10^6$).

```
Subalgorithm  $f(a, b)$ :  
     $cmmdc \leftarrow a$   
    While  $a \neq b$  execute  
        If  $a > b$  then  
             $a \leftarrow a - b$   
        Else  
             $b \leftarrow b - a$   
        EndIf  
         $cmmdc \leftarrow a$   
    EndWhile  
    return  $a * b / cmmdc$   
EndSubalgorithm
```

Care dintre urmatoarele afirmatii sunt adevarate referitor la subalgoritmul $f(a, b)$?

- A. Subalgoritmul calculeaza cel mai mic multiplu comun al celor 2 numere.
- B. Subalgoritmul nu calculeaza nimic concret.
- C. Subalgoritmul calculeaza cel mai mare divizor comun al celor 2 numere.
- D. Subalgoritmul returneaza numarul de divizori comuni al celor 2 numere.

3. Subalgoritmul $vector(a, n)$ primeste ca si parametrii un sir a cu n elemente numere intregi ($1 \leq n \leq 1000$, $a[1], a[2], \dots, a[n]$ si $-1000 \leq a[i] \leq 1000$).

```
Subalgorithm  $vector(a, n)$ :  
    If  $n = 1$  then  
        return  $(a[n] \text{ MOD } 2) * a[n]$   
    EndIf  
    return  $vector(a, n-1) + (a[n] \text{ MOD } 2) * a[n]$   
EndSubalgorithm
```

Precizati care dintre urmatoarele afirmatii sunt adevarate.

- A. Pentru $vector([10, 11, 12, 13, 15, 16, 17], 6)$ se returneaza 56.
- B. Pentru $vector([1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 10)$ se returneaza 25.
- C. Algoritmul calculeaza numarul de numere impare din sir.
- D. Subalgoritmul returneaza suma numerelor impare din sir.

4. Se considera subalgoritmul `ce_face(a, n)` care primeste ca si parametrii un sir **a** cu **n** elemente numere naturale nenule ($a[1], a[2], \dots, a[n]$ cu $1 \leq a[i] \leq n$ si $1 \leq n \leq 1000$).

```
Subalgorithm ce_face(a, n):  
    i ← 1  
    j ← n  
    s ← 0  
    While i ≤ j execute  
        If a[i] + a[j] = 2 * a[i] then  
            s ← s + a[i] + a[j]  
        EndIf  
        i ← i + 1  
        j ← j - 1  
    EndWhile  
    return s  
EndSubalgorithm
```

Precizati care este efectul acestui subalgoritm stiind ca apelul initial este `ce_face(a, n)`?

- A. Calculeaza cate numere egal departate de capetele sirului sunt egale.
- B. Pentru fiecare pereche de numere egal departate de capetele sirului care au aceiasi valoare, aduna la o suma valoarea respectiva.
- C. Returneaza mereu un numar par.
- D. Returneaza suma tuturor numerelor din sir care au proprietatea ca sunt egale cu perechea egal departata de capatul sirului (de la celalalt capat).

5. Subalgoritmul `divide(a, st, dr)` primeste ca si parametrii un sir **a** cu **n** elemente numere naturale si doua valori **st** si **dr** reprezentand 2 pozitii din acest sir ($1 \leq n \leq 1000$, $1 \leq st \leq dr \leq n$, $1 \leq a[i] \leq 1000$, $a[1], a[2], \dots, a[n]$).

```
Subalgorithm divide(a, st, dr):  
    If st = dr then  
        x ← a[st]  
        d ← 2  
        cnt ← 1  
    While x > 1 execute
```

```

        p ← 0
        While x MOD d = 0 execute
            x ← x DIV d
            p ← p + 1
        EndWhile
        If d > 2 then
            cnt ← cnt * (p + 1)
        EndIf
        d ← d + 1
        If d * d > x then
            d ← x
        EndIf
    EndWhile
    return cnt
EndIf
mij ← (st + dr) / 2
return max(divide(a, st, mij), divide(a, mij+1, dr))
EndSubalgorithm

```

Stiind ca subalgoritmul $\max(a, b)$ returneaza valoarea mai mare dintre parametrii a si b , precizati care dintre urmatoarele afirmatii sunt adevarate!

- A. Subalgoritmul $\text{divide}(a, st, dr)$ calculeaza si returneaza, utilizand principiul divide et impera, numarul maxim de divizori impari ai unui element al sirului a din intervalul de indici $[st, dr]$.
- B. Subalgoritmul nu calculeaza nimic concret intrucat in cazul in care st este egal cu dr , nu verifica nimic clar.
- C. Subalgoritmul calculeaza numarul maxim de divizori ai unui numar din intervalul de indici $[st, dr]$.
- D. Algoritmul are o complexitate liniara.

6. Precizati care dintre urmatorii algoritmi au o complexitate $O(\log_2 n)$?

- A. Cautarea unui element intr-un sir ordonat.
- B. Verificarea unui numar daca este patrat perfect.
- C. Verificarea primalitatii unui numar.
- D. Determinarea elementului minim dintr-un sir ordonat.

7. Utilizând metoda backtracking se generează toate posibilitățile de a forma șiruri de câte 4 animale din mulțimea {**cal**, **câine**, **papagal**, **porumbel**, **ponoi**}, astfel încât pe oricare două poziții alăturate să nu se afle animale din submulțimea {**cal**, **papagal**, **ponoi**}. Primele opt soluții generate sunt, în această ordine: (**cal**, **câine**, **cal**, **câine**), (**cal**, **câine**, **cal**, **porumbel**), (**cal**, **câine**, **câine**, **cal**), (**cal**, **câine**, **câine**, **câine**), (**cal**, **câine**, **câine**, **papagal**), (**cal**, **câine**, **câine**, **porumbel**), (**cal**, **câine**, **câine**, **ponoi**), (**cal**, **câine**, **papagal**, **câine**). Indicați numărul de soluții generate care au pe prima poziție un **câine** și pe ultima poziție un **porumbel**.

- A. 20
- B. 18
- C. 16
- D. 14

8. Se considera subalgoritmul calcul(n) unde **n** este un numar intreg ($-10000 \leq n \leq 10000$).

Subalgorithm calcul(n):

ok \leftarrow true

While n > 0 **then**

If n MOD 2 = 0 **then**

 ok \leftarrow false

EndIf

 n \leftarrow n DIV 2

EndWhile

return ok

EndSubalgorithm

Precizati care dintre urmatoarele afirmatii sunt adevarate:

- A. Subalgoritmul returneaza true pentru orice numar care este o putere a lui 2.
- B. Subalgoritmul verifica daca **n** poate fi scris ca suma de puteri distincte ale lui 2.
- C. Subalgoritmul returneaza true daca si numai daca numarul **n** este de forma $2^k - 1$ oricare ar fi **k** numarul natural sau **n** este un numar negativ.
- D. Pentru orice numar negativ se returneaza true.

9. In vederea determinării celui de-al k -lea număr prim, se poate implementa un algoritm cu antetul *getKthPrimeNumber(k)* care utilizează funcția predefinită *isPrime(n)* care returnează **true** în cazul în care numărul n e prim sau **false** în caz contrar ($1 \leq k \leq 1000$). Precizați care dintre următoarele implementări reprezintă implementări corecte ale subalgoritmului *getKthPrimeNumber(k)*.

A. **Subalgorithm** *getKthPrimeNumber(k)*:

```
d ← 2
While k > 0 execute
    If isPrime(d) then
        k ← k - 1
    EndIf
    d ← d + 1
EndWhile
return d
EndSubalgorithm
```

B. **Subalgorithm** *getKthPrimeNumber(k)*:

```
d ← 2
While k > 0 execute
    If isPrime(d) then
        k ← k - 1
    EndIf
    d ← d + 1
EndWhile
return d - 1
EndSubalgorithm
```

C. **Subalgorithm** *getKthPrimeNumber(k)*:

```
d ← 2
While k > 0 execute
    If isPrime(d) then
        k ← k - 1
    EndIf
    If d > 2 then
        d ← d + 1
    EndIf
    d ← d + 1
```

```

    EndWhile
    return d
EndSubalgorithm

```

D. Subalgorithm *getKthPrimeNumber*(k):

```

    d ← 2
    While k > 0 execute
        If isPrime(d) then
            k ← k - 1
        EndIf
        If d > 2 then
            d ← d + 1
        EndIf
        d ← d + 1
    EndWhile
    If d > 3 then
        return d - 2
    Else
        return d - 1
    EndIf
EndSubalgorithm

```

10. In vederea determinarii celui mai mic multiplu comun al unui sir cu n elemente numere naturale nenule, o abordare incorecta ar fi:

- A. Sa determinam cel mai mare divizor comun al tuturor numerelor din sir si produsul numerelor din sir sa-l impartim la aceasta valoare.
- B. Sa calculam cel mai mic multiplu comun al primelor 2 numere si mai departe sa luam pe rand fiecare numar ramas din sir si sa calculam cel mai mic multiplu comun dintre numarul curent si cel mai mic multiplu comun calculat anterior.
- C. Rezultatul este mereu egal cu produsul numerelor din fisier deoarece la un sir cu multe elemente cel mai mare divizor comun e mereu 1.
- D. Sa calculam suma tuturor numerelor din sir si sa o impartim la numarul de elemente pe care le contine sirul nostru.

11. Se considera subalgoritmul *modificare*(a, n) care primeste ca si parametrii un sir **a** cu **n** elemente numere naturale nenule ($1 \leq n \leq 10^6$, $a[1], a[2], \dots, a[n]$, $10 \leq a[i] \leq 1000$).

Subalgorithm modificare(a, n):

$i \leftarrow 1$

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

$x \leftarrow a[j]$

$s \leftarrow 0$

For $d \leftarrow 1, \sqrt{x}$ **execute**

If $x \bmod d = 0$ **then**

$s \leftarrow s + d + x / d$

EndIf

EndFor

If $s = x + 1$ **then**

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

$a[j] \leftarrow a[i] - a[j]$

$a[i] \leftarrow a[i] - a[j]$

$i \leftarrow i + 1$

EndIf

EndFor

EndSubalgorithm

Precizati care dintre urmatoarele afirmatii sunt adevarate referitoare la secventa de cod de mai sus:

- A. Algoritmul rearanjeaza elementele sirului in asa fel incat elementele prime sa ocupe primele pozitii din sir.
- B. Algoritmul rearanjeaza elementele sirului in asa fel incat elementele patrate perfecte sa ocupe primele pozitii din sir.
- C. Algoritmul pozitioneaza elementele pare din sir pe primele pozitii, iar pe cele impare in continuarea acestora.
- D. Daca in sir apare valoarea 2, aceasta va ocupa prima pozitie din sir in urma modificarilor produse de subprogram.

12. Subalgoritmul $f(n)$ primeste ca si parametru de intrare un numar natural nenul n ($1 \leq n \leq 10^4$).

Subalgorithm $f(n)$:

If $n = 0$ **then**

return 0

EndIf


```

If n MOD 2 = 0 then
    return f(n / 10) * 100 + n MOD 10
EndIf
return f(n / 10) * 10 + n MOD 10
EndSubalgorithm

```

Precizati care dintre urmatoarele apeluri afiseaza ca rezultat numarul prezentat:

- A. $f(12345) \rightarrow 1023045$
- B. $f(246) \rightarrow 20406$
- C. $f(1234) \rightarrow 102304$
- D. $f(1357) \rightarrow 1030507$

13. Stiind ca y este numar natural, care dintre urmatoarele expresii au valoarea True daca si numai daca y este numar impar care **NU** apartine intervalului inchis $[5, 15]$?

- A. **NOT**(($y \geq 5$) **AND** ($y \leq 15$)) **AND** ($y \bmod 2 = 1$)
- B. ($y \bmod 2 = 1$) **AND** (($y < 5$) **OR** ($y > 15$))
- C. **NOT**($y \bmod 2 = 0$) **AND** (($y > 5$) **AND** ($y < 15$))
- D. **NOT**(($y \bmod 3 = 1$) **OR** ($y \bmod 5 = 0$) **OR** (($y \geq 5$) **AND** ($y \leq 15$)))

14. Se considera subalgoritmul verifica(a, n) care primeste ca si parametrii un sir a cu n elemente numere naturale nenule ($1 \leq n \leq 1000$, $a[1], a[2], \dots, a[n]$, $1 \leq a[n] \leq 10^5$).

```

Subalgorithm verifica( $a, n$ ):
    For  $i \leftarrow 1, n - 1$  execute
        If  $a[i] \geq a[i+1]$  then
            return false
        EndIf
    EndFor
    return true
EndSubalgorithm

```

Precizati care dintre urmatoarele informatii descriu corect efectul subalgoritmului considerand datele de intrare in conformitate cu restrictiile din textul problemei.

- A. Subalgoritmul verifica daca numerele din sir sunt in ordine crescatoare.
- B. Subalgoritmul verifica daca numerele din sir sunt distincte si in ordine strict crescatoare.
- C. Subalgoritmul verifica daca elementele sirului sunt in ordine descrescatoare.
- D. Subalgoritmul verifica daca numerele din sir sunt in ordine strict crescatoare.

15. Se considera subalgoritmul suma(n) care primeste ca parametru un numar natural nenul n ($1 \leq n \leq 1000$).

```

Subalgorithm suma(n):
    If n = 0 then
        return 0
    EndIf
    return -1 * suma(n - 1) + n * (n - 1) / 2
EndSubalgorithm

```

Precizati care dintre urmatoarele afirmatii sunt adevarate:

- A. $\text{suma}(n) = 0 - 1 + 3 - 6 + \dots + (-1)^{n+1} * n * (n-1) / 2$
- B. $\text{suma}(n) = 0 - 1 + 3 - 6 + \dots + (-1)^n * n * (n-1) / 2$
- C. $\text{suma}(n) = 0 + 1 + 3 + 6 + \dots + (-1)^n * n * (n-1) / 2$
- D. $\text{suma}(n) = 0 - 1 + 3 - 6 + \dots + (-1)^{n-1} * n * (n-1) / 2$

16. Se considera subprogramele f1(a, b) si f2(a, b) unde a si b sunt 2 numere naturale nenule ($1 \leq a, b \leq 1000$).

```

Subalgorithm f1(a, b):
    If b = 0 then
        return 1
    EndIf
    return a * f1(a, b-1)
EndSubalgorithm

```

```

Subalgorithm f2(a, b):
    If b = 0 then
        return 1

```

```

EndIf
m ← f2(a, b DIV 2)
If b MOD 2 = 0 then
    return m * m
Else
    return a * f1(a, b - 1)
EndIf
EndSubalgorithm

```

Precizati care dintre urmatoarele informatii referitoare la cei 2 algoritmi sunt corecte:

- A. Ambii algoritmi au aceiasi complexitate timp.
- B. Ambii algoritmi calculeaza rezultatul ridicarii la puterea b a numarului a.
- C. Algoritmul f2 are o complexitate mai buna decat f1.
- D. Algoritmul f2 nu calculeaza corect rezultatul ridicarii la puterea b a numarului a.

17. Referitor la cele 2 subprograme de la grila **16.**, precizati care dintre informatiile de mai jos sunt adevarate:

- A. f2(2, 14) provoaca executia subprogramului **f1** de 7 ori.
- B. f2(2, 28) provoaca executia subprogramului **f1** de 7 ori.
- C. f2(5, 15) provoaca executia unui singur auto-apel al functiei **f2**.
- D. Niciunul dintre raspunsurile A, B sau C nu sunt corecte.

18. Se considera subprogramul f(a, n, k) care primeste ca si parametrii un sir **a** cu **n** element numere naturale si un numar **k** ($1 \leq n \leq 1000$, $a[1], a[2], \dots, a[n]$, $1 \leq a[i] \leq 1000$). Subprogramul max(a, b) returneaza valoarea maxima dintre **a** si **b**.

```

Subalgorithm f(a, n, k):
    If k > n then
        return 0
    EndIf
    x ← f(a, n, k+1)
    y ← f(a, n, k+2)
    return max(x, y + a[k])
EndSubalgorithm

```

Pentru care dintre urmatoarele date de intrare se va returna valoarea 21 stiind ca k ia mereu valoarea 1 la apelul initial?

- A. $a = [1, 2, 3, 4, 5, 6], n = 6$
- B. $a = [5, 14, 21, 20, 11], n = 5$
- C. $a = [7, 3, 2, 7, 3, 7, 3], n = 7$
- D. $a = [11, 2, 5, 3, 5, 1], n = 6$

19. Se considera subalgoritmul verificare(s, n) care primeste ca parametri de intrare un sir de caractere s cu n elemente indexate de la 1 ($1 \leq n \leq 1000, s[1], s[2], \dots, s[n], s[i] = \{(' ', ' ')\}$). Sirul contine doar caracterele '(' si ')'. Subprogramul verificare(s, n) returneaza true daca sirul s cu n elemente reprezinta un sir corect parantezat, respectiv false in caz contrar. Precizati care dintre urmatoarele implementari sunt corecte:

A. **Subalgorithm** verificare(s, n):

```

cnti ← 0
cntd ← 0
For i ← 1, n execute
    If s[i] = '(' then
        cntd ← cntd + 1
    Else
        cnti ← cnti + 1
    EndIf
EndFor
return cntd = cnti
EndSubalgorithm

```

B. **Subalgorithm** verificare(s, n):

```

cnti ← 0
cntd ← 0
For i ← 1, n execute
    If s[i] = '(' then
        cntd ← cntd + 1
    Else
        cnti ← cnti + 1
    EndIf
    If cnti > cntd then
        return false
    EndIf

```

```
    EndFor
    return true
EndSubalgorithm
```

C. **Subalgorithm** verificare(s, n):

```
    cnti ← 0
    cntd ← 0
    For i ← 1, n execute
        If s[i] = '(' then
            cntd ← cntd + 1
        Else
            cnti ← cnti + 1
        EndIf
        If cnti > cntd then
            return false
        EndIf
    EndFor
    return cnti = cntd
EndSubalgorithm
```

D. **Subalgorithm** verificare(s, n):

```
    cnti ← 1
    cntd ← 1
    For i ← 1, n execute
        If s[i] = '(' then
            cntd ← cntd + 1
        Else
            cnti ← cnti + 1
        EndIf
        If cnti > cntd then
            return false
        EndIf
    EndFor
    If cnti != cntd then
        return false
    Else
        return true
    EndIf
EndSubalgorithm
```

20. O persoana are de urcat n trepte. Stiind ca initial este pe treapta 1 si poate urca cate 1, cate 2 sau cate 3 trepte o data, precizati care dintre urmatoarele afirmatii sunt adevarate.

- A. Numarul de moduri in care pot urca 10 trepte este 150.
- B. Numarul de moduri in care pot urca 9 trepte este 81.
- C. Numarul de moduri in care pot urca 8 trepte este 24.
- D. Numarul de moduri in care pot urca 10 trepte este 149.

21. Se considera subalgoritmul cautare(a, n, val) unde a este un sir cu n elemente numere naturale ordonate strict crescator, iar val este un numar natural nenul ($1 \leq val \leq 1000, 1 \leq n \leq 1000, a[1], a[2], \dots, a[n]$).

Subalgorithm cautare(a, n, val):

$st \leftarrow 1$

$dr \leftarrow n$

While $st \leq dr$ **execute**

$mij \leftarrow (st + dr) / 2$

If $a[mij] > val$ **then**

$dr \leftarrow mij - 1$

Else

$st \leftarrow mij + 1$

EndIf

EndWhile

return $st - 1$

EndSubalgorithm

Precizati care dintre informatiile de mai jos descriu corect efectul executiei subalgoritmului cautare(a, n, val).

- A. Subalgoritmul determina pozitia celui mai mare element din sir care este mai mic sau egal cu val .
- B. Subalgoritmul determina pozitia - 1 a celui mai mic element din sir care este mai mare strict ca val .
- C. Subalgoritmul determina pozitia celui mai mic element din sir care este mai mare sau egal cu val .
- D. In cazul in care val apare in sir, se returneaza pozitia pe care acesta apare.

22. Se considera subalgoritmul dei(a, n) care primeste ca si parametrii de intrare o matrice a cu n linii si n coloane. Elementele matricei sunt numere naturale

nenule ($1 \leq n \leq 1000$, $a[1][1]$, $a[1][2]$, ... $a[n][n-1]$, $a[n][n]$, $-1000 \leq a[i][j] \leq 1000$).

```
Subalgorithm dei(a, n):  
    sum  $\leftarrow$  0  
    For i  $\leftarrow$  1, n execute  
        For j  $\leftarrow$  i, n execute  
            If  $a[i][j] \bmod 2 = 1$  then  
                sum  $\leftarrow$  sum +  $a[i][j]$   
            Else  
                sum  $\leftarrow$  sum -  $a[i][j]$   
            EndIf  
        EndFor  
    EndFor  
    return sum  
EndSubalgorithm
```

Precizati care dintre urmatoarele afirmatii descriu corect efectul acestui subalgorithm.

- A. Subalgorithmul determina suma tuturor elementelor din matrice.
- B. Subalgorithmul determina diferenta dintre suma elementelor impare si suma elementelor pare din matrice.
- C. Subalgorithmul determina diferenta dintre suma elementelor impare si suma elementelor pare din matrice situate deasupra de diagonala principala.
- D. Nicio afirmatie nu descrie corect efectul subalgorithmului.

23. Se considera subalgorithmul $f(n)$ care primeste ca si parametru un numar natural nenul n ($1 \leq n \leq 1000$).

```
Subalgorithm f(n):  
    p  $\leftarrow$  1  
    cn  $\leftarrow$  n  
    While cn > 0 execute  
        If  $cn \bmod 3 = 2$  then  
            n  $\leftarrow$  0  
        EndIf  
        cn  $\leftarrow$  cn DIV 3
```

```

EndWhile
While  $n > 0$  execute
    If  $n \bmod 3 = 1$  then
        Write  $p$ , " "
    EndIf
     $p \leftarrow p * 3$ 
     $n \leftarrow n \text{ DIV } 3$ 
EndWhile
EndSubalgorithm

```

Precizati care dintre urmatoarele informatii sunt false:

- A. Algoritmul afiseaza o modalitate de a-l scrie pe n ca suma de puteri ale lui 3.
- B. Algoritmul afiseaza (daca exista) o modalitate de a-l scrie pe n ca suma de puteri distincte ale lui 2.
- C. Algoritmul afiseaza (daca exista) o modalitate de a-l scrie pe n ca suma de puteri distincte ale lui 3.
- D. Algoritmul are o complexitate logaritmica.

24. Se considera subalgoritmul $\text{modif}(a, n, st, dr)$ care primeste ca si parametrii de intrare un sir a cu n elemente numere naturale nenule si 2 indici, st si dr . ($1 \leq n \leq 1000$, $a[1], a[2], \dots, a[n]$, $1 \leq st \leq dr \leq n$).

```

Subalgorithm  $\text{modif}(a, n, st, dr)$ :
     $i \leftarrow st$ 
     $j \leftarrow dr$ 
    While  $j < n$  execute:
         $j \leftarrow j + 1$ 
         $a[i] \leftarrow a[j]$ 
         $a[j] \leftarrow a[j] + 1$ 
         $i \leftarrow i + 1$ 
    EndWhile
     $n \leftarrow n - 1$ 
     $n \leftarrow n - dr$ 
     $n \leftarrow n + st$ 
EndSubalgorithm

```

Precizati care dintre informatiile de mai jos descriu modificarile produse asupra tabloului a cu n elemente.

- A. Se elimina elementele din intervalul (st, dr) din sir.
- B. Se elimina elementele din intervalul $[st, dr)$ din sir.
- C. Sirul se scurteaza indiferent de valorile parametrilor st si dr daca acestia sunt in conformitate cu restrictiile impuse de enunt.
- D. Subalgoritmul elimina o secventa de unul sau mai multe elemente din sir si are grija sa adapteze lungimea sirului stocata in variabila n la noul sir modificat.