ACADEMIA ZECELAINFO

Simulare Examen Admitere UBB 2023 - 27 mai 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. Precizati care dintre urmatoarele expresii verifica corect daca numarul memorat in **n** (**n** numar real) **NU** apartine multimii: **{-12} U [-3, 7]**?

A. (**n** ≤ -12) **SI** (**n** ≥ -12) **SI** (**n** ≤ -3) **SI** (**n** ≥ 7)
B. (**n** < -12) **SAU** (**n** > -12 **SI** **n** < -3) **SAU** (**n** > 7)
C. (**n** < -12) **SAU** (**n** > -12) **SI** (**n** < -3) **SAU** (**n** > 7)
D. (**n** < -3) **SI** (**n** ≠ -12) **SAU** (**n** > 7)

2. Fie subalgoritmul f(a, b) unde **a** si **b** sunt 2 numere naturale cu (1 ≤ **a**, **b** ≤ 10000).

```
Subalgorithm f(a, b):
    If a = b then
        return a
    EndIf
    If a > b then
        return f(a - b, b)
    Else
        return f(a, b - a)
    EndIf
EndSubalgorithm
```

Alegeti variantele care completeaza corect spatiul subliniat din subalgoritmul de mai jos astfel incat cei doi algoritmi sa returneze mereu aceiasi valoare.

```
Subalgorithm f1(a, b):
    If b = 0 then
        return a
    Else
        r ← a DIV b
        return ________________
    EndIf
EndSubalgorithm
```

A. f1(b, r)
B. f1(b, a **MOD** b)
C. f1(b, a **DIV** b)
D. f1(a **MOD** b, b)

3. Care dintre urmatorii algoritmi pot fi implementari in asa fel incat sa aiba complexitate de timp liniara (O(**n**))?

A. Algoritmul de calcul a sumei elementelor de pe diagonala secundara a unei matrici.
B. Algoritmul de cautare binara a unui element intr-un sir ordonat crescator cu n elemente numere naturale.
C. Algoritmul de descompunere in factori primi a unui numar natural nenul.
D. Sortarea unui vector cu **n** elemente numere naturale - cifre.

4. Care este rezultatul conversiei numarului binar 10100011011101 in baza 10?

A. 11997
B. 10461
C. 11973
D. Niciuna dintre variantele A, B sau C.

5. Se considera subprogramul ce_face(n, a) unde **n** este un numar natural nenul si **a** reprezinta un sir cu n elemente numere naturale nenule ($1 \leq \mathbf{n} \leq 10000$, $1 \leq \mathbf{a}[i] \leq 10^9$, oricare ar fi $1 \leq \mathbf{i} \leq \mathbf{n}$).

```
Subalgorithm ce_face(n, a):
    s ← 0
    For i ← 1, n execute
        nr ← 1
        While a[i] > 9 execute
```

```
                nr ← nr * 10
                a[i] ← a[i] DIV 10
            EndWhile
            s ← s + a[i] * nr
        EndFor
        return s
    EndSubalgorithm
```

Precizati care dintre urmatoarele afirmatii sunt adevarate referitoare la subalgoritmul ce_face(n, a).

A. ce_face(5, [222, 2043, 29, 2, 20035]) returneaza valoarea 2222.
B. ce_face(3, [1, 2, 3, 4, 5]) returneaza valoarea 10.
C. ce_face(5, [34, 254, 21, 543, 123]) returneaza valoarea 850.
D. ce_face(2, [123, 123]) returneaza valoarea 200.

6. Se considera subalgoritmul f(x), unde **x** este numar natural ($1 \leq \mathbf{x} \leq 10000$).

```
    Subalgorithm f(x):
        If x = 0 then
            return 0
        Else
            If x MOD 7 = 0 then
                return f(x DIV 10) + 1
            Else
                return f(x DIV 10)
            EndIf
        EndIf
    EndSubalgorithm
```

Pentru ce valori ale lui **x**, subalgoritmul de mai sus nu va returna valoarea 3?

A. x = 7140
B. x = 71
C. x = 71400
D. x = 7777

7. Se considera urmatorul subalgoritm function(n) unde **n** este un numar natural nenul ($1 \leq$ **n** ≤ 100000):

```
Subalgorithm function(n):
    i ← n
    While i > 0 execute
        j ← i
        While j ≥ 0 execute
            j ← j - 1
        EndWhile
        i ← i DIV 2
    EndWhile
EndSubalgorithm
```

Precizati in ce clasa de complexitati se incadreaza algoritmul de mai sus.

A. $O(n * log_2 n)$
B. O(n)
C. $O(n^2)$
D. $O(n + log_2 n)$

8. Folosind metoda backtracking, se determina in ordine lexicografica, toate cuvintele de 6 litere distincte din multimea {**a**, **e**, **i**, **o**, **u**, **b**, **c**, **d**, **m**, **n**, **p**}, oricare doua litere alaturate neputand fi vocale. Primele 5 solutii sunt: **abecid**, **abecim**, **abecin**, **abecip**, **abecod**. Indicati care cuvant este generat inaintea solutiei **ebacid**.

A. apnmdc
B. acdmnp
C. apunom
D. eadmnp

9. Se considera urmatorul subprogram f(A, n) unde **A** este o matrice patratica cu **n** * **n** elemente numere naturale nenule, iar **n** este un numar natural strict pozitiv ($1 \leq$ **n** ≤ 1000, $0 \leq$ **A**[i][j] $\leq 10^9$ oricare ar fi **i**, **j** cu $1 \leq$ **i**, **j** $\leq$ **n**).

```
Subalgorithm f(A, n):
    For i ← 1, n DIV 2 execute
        For j ← i + 1, n - i + 1 execute
```

```
            A[i][j] ↔ A[n-j+1][n-i+1]
        EndFor
    EndFor
EndSubalgorithm
```

Precizati care este efectul acestui subalgoritm:

A. Subalgoritmul interschimba elementele din matrice situate deasupra de diagonala principala si deasupra de diagonala secundara cu elementele situate deasupra de diagonala principala si sub diagonala secundara.
B. Subalgoritmul interschimba elementele situate deasupra de diagonala secundara cu cele situate sub diagonala secundara.
C. Subalgoritmul interschimba elementele situate deasupra diagonalei principale si deasupra diagonalei secundare cu elementele simetrice fata de diagonala secundara.
D. Niciuna dintre variantele A, B si C nu exprima corect efectul.

10. Se considera subalgoritmul f(a, x, y) unde **a** este un vector cu **n** elemente numere naturale, iar **x** si **y** sunt 2 numere naturale, strict pozitive ($1 \le \mathbf{n} \le 1000$, $0 \le \mathbf{a}[i] \le 10^9$, $1 \le \mathbf{i} \le \mathbf{n}$, $1 \le \mathbf{x} \le \mathbf{y} \le \mathbf{n}$).

```
Subalgorithm f(a, x, y):
    If x = y then
        return 1
    EndIf
    z ← (x + y) DIV 2
    If f(a, x, z) = 1 AND f(a, z+1, y) = 1 then
        If a[z] MOD 2 ≠ a[z+1] MOD 2 then
            return 1
        Else
            return 0
        EndIf
    Else
        return 0
    EndIf
EndSubalgorithm
```

Precizati care dintre urmatoarele informatii sunt **false** referitoare la subprogramul de mai sus, considerand apelul initial f(a, 1, n).

A. Subalgoritmul returneaza 1 daca si numai daca toate elementele sirului **a** au aceiasi paritate.
B. Subalgoritmul returneaza 1 daca si numai daca elementele sirului **a** sunt in ordine crescatoare.
C. Subalgoritmul returneaza mereu 0.
D. Subalgoritmul returneaza 1 daca si numai daca oricare ar fi 2 elemente consecutive in sirul **a**, acestea au paritati diferite.

11. Referitor la algoritmul de la punctul **10.** , precizati care dintre urmatoarele informatii sunt adevarate:

A. Algoritmul foloseste conceptul de Cautare Binara.
B. Algoritmul foloseste conceptul de Divide Et Impera.
C. Algoritmul are o complexitate O($log_2 n$).
D. Algoritmul are o complexitate liniara (O(**n**)).

12. Se considera urmatorul subalgoritm recursiv cautare(val, x, y) unde **val**, **x** si **y** sunt numere naturale ($1 \leq \mathbf{val} \leq 10^6$, $1 \leq \mathbf{x} \leq \mathbf{y} \leq \mathbf{val}$).

```
Subalgorithm cautare(val, x, y):
    If x = y then
        return x
    EndIf
    mij ← (x + y) DIV 2
    If mij * mij ≥ val then
        return cautare(val, x, mij)
    Else
        return cautare(val, mij+1, y)
    EndIf
EndSubalgorithm
```

Precizati care dintre urmatoarele afirmatii sunt corecte considerand ca apelul initial al functie este cautare(val, 1, val).

A. Subalgoritmul determina cel mai mic numar al carui patrat este mai mare sau egal cu **val**.
B. Subalgoritmul determina cel mai mare numar al carui patrat este mai mic sau egal cu **val**.

C. In cazul in care **val** este patrat perfect, subalgoritmul returneaza radacina acestuia.
D. Patratul numarului returnat de subalgoritmul cautare(val, 1, val) este mereu mai mare sau egal cu **val**.

13. Se considera urmatoarea secventa de algoritm, unde **a** este un vector de **n** numere naturale (**a**[1], **a**[2], …, **a**[n], $1 \leq$ **a**[i] $\leq 10^4$, pentru i = 1,2,...,**n**), iar **n** este un numar natural nenul ($1 \leq$ **n** $\leq 10^4$):

```
For i ← 1, n-1 execute
    For j ← n, i+1, -1 execute
        If a[j] < a[j-1] then
            a[j] ↔ a[j-1]
        EndIf
    EndFor
EndFor
```

Precizati care dintre urmatoarele informatii descriu corect algoritmul de mai sus.
A. Algoritmul sorteaza in ordine crescatoare elementele sirului **a** utilizand Sortarea prin Insertie.
B. Algoritmul sorteaza in ordine crescatoare elementele sirului **a** utilizand Sortarea prin Metoda Bulelor.
C. Algoritmul nu sorteaza corect sirul **a**.
D. Algoritmul sorteaza in ordine descrescatoare elementele sirului **a** utilizand Sortarea prin Insertie.

14. Se considera functia recursivitate(n) unde **n** este un numar natural nenul ($1 \leq n \leq 1000$).

```
Subalgorithm recursivitate(n):
    If n = 1 then
        return 1
    EndIf
    return recursivitate(n-1) + recursivitate(n-1) + 1
EndSubalgorithm
```

Precizati care dintre urmatoarele informatii sunt adevarate referitoare la functia recursivitate(n):

A. Algoritmul ar returna aceiasi valoare daca linia 5 ar fi inlocuita cu **return** 2 * recursivitate(n-1) + 1
B. In cazul inlocuirii prevazute la punctul A, complexitatea ramane aceiasi.
C. Complexitatea functiei recursivitate(**n**) este liniara.
D. Pentru **n** = 9, se returneaza valoarea $2^9 - 1$.

15. Se considera urmatoarea problema: Considerand ca variabila **x** este egala cu 1, asupra ei se pot efectua 3 operatii: **x** ← **x** + 1, **x** ← **x** * 2, **x** ← **x** * 3. Care este numarul minim de operatii pe care trebuie sa le efectuam asupra variabilei **x**, in asa fel incat aceasta sa devina egala cu un numar **n** dat.

Care dintre urmatoarele afirmatii sunt adevarate?

A. Pentru **n** = 5, se efectueaza 2 pasi.
B. Pentru **n** = 74, se efectueaza 6 pasi.
C. Pentru **n** = 1232, se efectueaza 11 pasi.
D. Pentru **n** = 711, se efectueaza 10 pasi.

16. Se considera urmatorul subprogram recursiv nr_mod(n, a), unde **n** este un numar natural nenul (1 ≤ **n** ≤ 1000), iar **a** este un numar intreg.

```
Subalgorithm nr_mod(n, a):
    If n = 0 then
        return 1
    EndIf
    cnt ← 0
    For i ← a + 2, n execute
        cnt ← cnt + nr_mod(n - i, i)
    EndFor
    return cnt
EndSubalgorithm
```

Precizati care dintre urmatoarele afirmatii sunt corecte considerand apelul initial al functiei nr_mod(n, -1).

A. Algoritmul calculeaza numarul de moduri in care poate fi scris **n** ca suma de numere naturale impare.

B. Algoritmul calculeaza numarul de moduri in care poate fi scris **n** ca suma de numere naturale nenule ne-consecutive.
C. Algoritmul calculeaza numarul de moduri in care poate fi scris **n** ca suma de numere naturale nenule pare.
D. Pentru **n** = 15, se returneaza 14.

17. Ne dorim implementarea unui soft pentru un bancomat. Utilizatorul va cere o anumita suma de bani si noi trebuie sa determinam o posibilitate de a-i da utilizatorului suma dorita. Bancnotele disponibile sunt: {1, 5, 10, 20, 50, 100, 200, 500}, iar cantitatea de bancnote din fiecare tip disponibile se considera a fi infinita. Cel mai bun algoritm pe care l-am putea implementa este:

A. Vom scrie un algoritm backtracking care determina toate permutarile bancnotelor si vedem in care situatie se atinge exact suma dorita.
B. Vom scrie un algoritm backtracking care determina toate modalitatiile de a plati suma dorita si alegem una dintre ele.
C. Folosind Metoda Greedy, putem determina o succesiune de bancnote care ating exact suma dorita intr-o complexitate liniara.
D. Este imposibila rezolvarea acestei probleme intrucat este imposibil sa scriem un algoritm care sa managerieze un numar infinit de bancnote.

18. Se considera subalgoritmul ce_face(n) unde **n** este un numar natural nenul ($1 \leq \mathbf{n} \leq 10^9$).

```
Subalgorithm ce_face(n):
    If n ≤ 10 then
        return 1
    EndIf
    If n MOD 2 ≠ (n DIV 10) MOD 2 then
        return ce_face(n DIV 10)
    Else
        return 0
    EndIf
EndSubalgorithm
```

Precizati care dintre urmatoarele afirmatii sunt adevarate.

A. ce_face(1234567890) returneaza 1.
B. ce_face(24689753) returneaza 1.
C. ce_face(123454321) returneaza 1.

D. ce_face(321233) returneaza 0.

19. Se considera subalgoritmul afis(n) unde **n** este un numar natural nenul (1 ≤ **n** ≤ 1000).

```
Subalgorithm afis(n):
    d ← 2
    While n > 1 execute
        p ← 1
        While n MOD d = 0 execute
            n ← n DIV d
            p ← p + 1
        EndWhile
        If p ≠ 0 then
            write d, '^', p, ' '
        EndIf
        d ← d + 1
    EndWhile
EndSubalgorithm
```

Precizati care dintre urmatoarele afirmatii sunt adevarate referiotare la subalgoritmul afis(n):

A. Algoritmul afiseaza scrierea in factori primi a numarului **n**.
B. Algoritmul are o complexitate O(√**n**)
C. Algoritmul nu afiseaza corect scrierea in factori primi a numarului **n**.
D. Algoritmul are o complexitate liniara.

20. Se considera urmatorul subalgoritm calcul(a, n) care primeste ca si parametrii un si **a** cu **n** elemente numere naturale (**a**[1], **a**[2], …, **a**[**n**]) si un numar natural nenul **n** (1 ≤ **n** ≤ 100).

```
Subalgorithm calcul(a, n):
    nr ← 0
    For i ← 0, 2^n execute
        j ← i
        sum ← 0
        For k ← 1, n execute
            If j MOD 2 = 1 then
```

```
                    sum ← sum + a[j]
                EndIf
                j ← j DIV 2
            EndFor
            If sum MOD n = 0 then
                nr ← nr + 1
            EndIf
        EndFor
        return nr
    EndSubalgorithm
```

Precizati care dintre urmatoarele afirmatii sunt adevatate in legatura cu calcul(a, n):

A. Subprogramul calcul(a, n) calculeaza numarul de partitii ale sirului a ale caror suma a elementelor este multiplu de **n**.
B. Subprogramul calcul(a, n) calculeaza numarul de submultimi ale sirului a caror suma a elementelor este divizibila cu **n**.
C. Subprogramul calcul(a, n) calculeaza suma elementelor submultimilor sirului a care au un numar de **n** elemente.
D. Subprogramul nu calculeaza nimic din ce a fost precizat la punctele anterioare.

21. Se considera subalgoritmul suma(n) unde **n** este un numar natural nenul ($1 \leq n \leq 10^9$).

```
    Subalgoritm suma(n):
        s ← 0
        For i ← 1, √n execute
            If n MOD i = 0 then
                s ← s + i
            EndIf
        EndFor
        return s
    EndSubalgorithm
```

Care dintre urmatoarele afirmatii sunt corecte?

A. Algoritmul calculeaza numarul de divizori ai lui **n**.
B. Oricare ar fi numarul **n**, daca suma(n) = 1, atunci numarul este prim.

C. Algoritmul calculeaza suma divizorilor numarului **n**.
D. Nicio afirmatie de la A, B si C nu este corecta.

22. Se considera urmatoarea secventa de cod unde **a** si **b** sunt 2 numere naturale nenule.

```
For i ← a * b, b, -1 execute
    If i MOD a = 0 AND i MOD b = 0 then
        d ← i
    EndIf
EndFor
```

Valoarea memorata de variabila **d** la finalul executiei secventei de cod de mai sus este:

A. Cel mai mare divizor comun.
B. Cel mai mic multiplu comun.
C. Numarul de divizori comuni.
D. Numarul de multipli comuni.

23. Se considera subalgoritmul f(x, y) unde **x** si **y** sunt 2 numere naturale nenule ($1 \leq \mathbf{x}, \mathbf{y} \leq 10^9$).

```
Subalgorithm f(x, y):
    If x > y then
        return f(f(y, x), x DIV y) - 1
    Else
        If x = y then
            return f(x + y, x) + 2
        Else
            return y - x
        EndIf
    EndIf
EndSubalgorithm
```

Precizati care este rezultatul apelului f(6, 2).

A. 0
B. 1
C. 2

D. f(4, 7) - 3

24. Un algoritm determina minimul si maximul dintr-un tablou unidimensional cu 100 de numere, prin operatii de comparare a elementelor. Numarul minim de comparari necesare este…

A. 200
B. 198
C. 148
D. 100

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