

Subiecte la testul grilă la Informatică

Atenție! Dacă nu este specificat altfel, primul element al unui tablou are indexul 0.

1. Fie graful neorientat conex $G = (V, E)$, unde mulțimea nodurilor este $V = \{1, 2, 3, 4, 5\}$, iar mulțimea muchiilor este $E = \{(1, 2), (2, 3), (1, 3), (3, 4), (4, 5), (3, 5)\}$. Care dintre următoarele afirmații referitoare la graful G este adevărată?

- (a) Nu conține niciun ciclu eulerian și niciun ciclu hamiltonian.
- (b) Conține un ciclu hamiltonian, dar nu conține un ciclu eulerian.
- (c) Conține un ciclu eulerian, dar nu conține un ciclu hamiltonian.
- (d) Conține atât un ciclu eulerian, cât și un ciclu hamiltonian.

2. Fie a un tablou unidimensional cu n elemente. Pentru a determina lungimea maximă a unei secvențe (termeni pe poziții consecutive din tabloul a) cu suma cel mult S , un elev propune următoarea soluție:

```
int st = 0, lmax = 0;
int suma = 0;
for (int dr = 0; dr < n; dr++) {
    suma += a[dr];
    while (suma > S) { suma -= a[st]; st++; }
    if (dr - st + 1 > lmax) lmax = dr - st + 1;
}
cout << lmax;
```

Pentru ce restricție asupra valorilor din tablou soluția produce întotdeauna rezultatul corect?

- (a) Numai dacă toți termenii tabloului sunt ≥ 0 .
- (b) Numai dacă toți termenii tabloului sunt strict pozitivi.
- (c) Pentru orice numere întregi din tablou, fără nicio restricție.
- (d) Numai dacă tabloul este ordonat crescător.

3. Într-un sistem de control al unei flote de mașini autonome, se lucrează cu trei senzori care generează valori întregi. Acestea sunt salvate în variabilele a , b , c . Expresiile următoare reprezintă condiții de siguranță care decid dacă o mașină este „autorizată să pornească” (valoarea 1 = adevărat, 0 = fals).

- i) $((a > b) \ \&\& \ (b < c)) \ || \ (a < c)$
- ii) $!((a < b) \ || \ (b < c)) \ \&\& \ (a > c)$
- iii) $((a > b) \ \&\& \ !(b < c)) \ || \ (a < c)$
- iv) $(a > b) \ \&\& \ (b > c) \ \&\& \ !(a < c)$
- v) $(a > b) \ \&\& \ (b > c) \ \&\& \ (a < c)$

Câte dintre expresiile date au valoarea 1 pentru orice valori ale variabilelor a , b , c care respectă relația $a > b > c$?

- (a) 2 (b) 3 (c) 4 (d) 1

4. Fie următoarea secvență de cod C/C++:

```
char s[] = "examen de admitere la facultate";
char t[100] = "2026";
int x = strlen(strcat(t, strchr(strstr(s, "te"), 'f')));
```

Care este valoarea variabilei x?

- (a) 13 (b) 12 (c) 21 (d) 16

5. Pentru un tablou unidimensional **a** care conține **n** numere întregi, se caută valoarea maximă a diferenței $a_j - a_i$ cu condiția $i < j$ (descăzutul apare în vector după scăzător). Următoarea secvență de cod rezolvă cerința într-o singură parcurgere. Ce valoare se afișează pentru tabloul $a = \{7, 1, 5, 3, 6, 4\}$?

```
int minim = a[0], dif = 0;
for (int i = 1; i < n; i++) {
    if (a[i] - minim > dif) dif = a[i] - minim;
    if (a[i] < minim) minim = a[i];
}
cout << dif;
```

- (a) 3 (b) 5 (c) 7 (d) 6

6. Se dă următoarea secvență de cod C/C++:

```
int m[4][4] = {
    {181, 168, 152, 125},
    {201, 227, 239, 205},
    {392, 389, 313, 341},
    {434, 495, 402, 478}
};

int fun(int mat[][4]) {
    int ret = 0, i = 0, j;
    while (i < 4) {
        for (j = i+1; j < 4; j+=1)
            if ( ... )
                ret += 1;
        i++;
    }
    return ret;
}
```

Pentru care dintre următoarele condiții evaluate de instrucțiunea `if`, apelul funcției `fun(m)` returnează valoarea minimă?

- (a) `mat[i][j] % 10 != mat[j][i] % 10`
(b) `mat[i][j] % 10 > mat[j][i] % 10`
(c) `mat[i][j] % 10 < mat[j][i] % 10`
(d) `mat[i][j] % 10 == mat[j][i] % 10`

7. Se dă următoarea secvență de instrucțiuni C/C++:

```
int x = 2026;
int a = x / 100 + x % 1000;
int b = x / 100 + x % 10;
int c = x / 1000 + x % 100;
```

Care este relația de ordine corectă între valorile variabilelor **a**, **b**, **c**?

- (a) $c < b < a$ (b) $b < a < c$ (c) $c < a < b$ (d) $b < c < a$

8. Se consideră următoarele afirmații referitoare la metodele de sortare crescătoare a unui vector:

- i) La sortarea prin metoda bulelor (Bubble Sort), după prima trecere asupra vectorului $x = \{4, 2, 5, 1, 3\}$, ultimul element al acestuia este 3.
- ii) La sortarea prin metoda selecției (Selection Sort), după fiecare trecere se fixează pe poziția curentă cel mai mic element dintre cele încă nesortate.
- iii) Pentru vectorul $x = \{6, 3, 5, 2, 4\}$, după prima trecere la sortarea prin metoda selecției, vectorul devine $x = \{2, 4, 5, 6, 3\}$.
- iv) La sortarea prin metoda inserției (Insertion Sort), după fiecare pas, partea din stânga a vectorului deja procesată este ordonată crescător.
- v) Metoda de sortare prin numărare (Counting Sort) este eficientă atunci când valorile elementelor aparțin unui interval relativ mic.
- vi) Pentru un vector deja ordonat crescător, metoda de sortare prin inserție poate parcurge algoritmul fără a deplasa niciun element.
- vii) Sortarea prin metoda bulelor compară și interschimbă doar elemente aflate pe poziții consecutive.

Câte dintre afirmațiile de mai sus sunt corecte?

- (a) 7 (b) 5 (c) 4 (d) 6

9. Ce se afișează după execuția secvenței de cod C/C++?

```
int c = 0;
for (int i = 1; i <= 5; i++)
    for (int j = 1; j <= 5; j++)
        if ((i + j) == 6)
            c++;
cout << c;
```

- (a) 5 (b) 3 (c) 6 (d) 10

10. Care este complexitatea timp pentru secvența de cod C/C++ din funcția fun?

```
void fun(int n, int arr[]) {
    int i = 0, j = 0;
    for (; i < n; ++i)
        while (j < n && arr[i] < arr[j])
            j++;
}
```

- (a) $\mathcal{O}(n^2)$ (b) $\mathcal{O}(\log^2 n)$ (c) $\mathcal{O}(n)$ (d) $\mathcal{O}(n \log n)$

11. Care este valoarea variabilei sum la finalul execuției următoarei secvențe de cod C/C++?

```
char s[] = "3a5b";
int sum = 0;
for (int i = 0; i < strlen(s); i++)
    if (s[i] >= '0' && s[i] <= '9')
        sum += s[i] - '0';
cout << sum;
```

- (a) 35 (b) 104 (c) 8 (d) 0

12. Se consideră următoarea funcție recursivă, definită în limbajul C/C++:

```
void f(int n) {
    if (n > 0) {
        cout << n % 10;
        f(n / 100);
        cout << n % 100;
    }
}
```

Câte cifre impare se vor afișa pe ecran în urma apelului $f(30016)$?

(a) 3 (b) 1 (c) 2 (d) 4

13. Se consideră următoarea secvență de cod C/C++:

```
char x[] = "mississippi";
int g(char s[], int k) {
    int c = 0, i, L;
    L = strlen(s);
    for(i = 0; i < L; i++)
        if(s[i] == s[k])
            c++;
    return c;
}

int f(char s[]) {
    int i, m = 0, L;
    L = strlen(s);
    for(i = 0; i < L; i++)
        if(g(s, i) > g(s, m))
            m = i;
    return m;
}
```

Ce valoare returnează apelul $f(x)$?

(a) 10 (b) 1 (c) 2 (d) 4

14. Ce afișează următoarea secvență de cod C/C++?

```
struct nod {
    int val;
    int leg;
};

nod a[6] = {
    {2, 2}, {1, 0}, {4, 4},
    {3, 0}, {8, 0}, {5, 1}
};

int main() {
    int i = 0, s = 0, p = 0;
    while (p < 6 && a[i].val % 2 == 0) {
        s += a[i].val++;
        i = a[i].leg;
        p++;
    }
    cout << s << " " << p << " " << i;
    return 0;
}
```

(a) 14 3 0 (b) 28 6 0 (c) 6 2 4 (d) 14 3 4

15. Se dă următoarea secvență de cod C/C++:

```
int s = 0;
for (int i = n; i > 0; i /= 2)
    for (int j = i; j < 2 * i; j++)
        s++;
```

Care este complexitatea timp a algoritmului descris de secvență?

(a) $\mathcal{O}(n^2)$ (b) $\mathcal{O}(n \log n)$ (c) $\mathcal{O}(n)$ (d) $\mathcal{O}(\log n)$

16. În matricea de adiacență a unui graf orientat cu 8 noduri sunt 14 valori nenule. Care este numărul maxim posibil de componente tare conexe ale grafului?

(a) 6 (b) 5 (c) 8 (d) 7

17. Într-un joc video, personajul principal trebuie să deblocheze 6 abilități speciale pe parcursul aventurii sale. Deblocarea acestor abilități se realizează respectând următoarele reguli:

- Abilitatea „Agilitate” trebuie deblocată înaintea abilităților „Salt Dublu” și „Eschivă”.
- „Salt Dublu” trebuie deblocat înaintea abilității „Lovitură Aeriană”.
- „Eschivă” trebuie deblocată înaintea abilităților „Lovitură Aeriană” și „Contraatac”.
- Atât „Lovitură Aeriană”, cât și „Contraatac”, trebuie deblocate înaintea abilității supreme „Furia Zeilor”.

În câte moduri distincte (ordini de deblocare) poate jucătorul să obțină toate cele 6 abilități, respectând regulile jocului?

(a) 8 (b) 6 (c) 5 (d) 4

18. Ancuța dorește să își planifice vacanțele, zburând direct din Iași către diverse destinații europene. Pentru a lua o decizie optimă, ea a notat durata totală a zborului dus-întors și a asociat fiecărui oraș un „grad de satisfacție”. Ancuța ia în considerare următoarele 7 destinații și informațiile asociate:

- | | |
|---|--------------------------------------|
| • București: zbor 2 ore, satisfacție 7 | • Paris: zbor 6 ore, satisfacție 20 |
| • Viena: zbor 3 ore, satisfacție 7 | • Londra: zbor 7 ore, satisfacție 24 |
| • Roma: zbor 4 ore, satisfacție 15 | • Madrid: zbor 8 ore, satisfacție 24 |
| • Bruxelles: zbor 5 ore, satisfacție 18 | |

Ancuța vrea să aleagă o submulțime de destinații distincte dintre cele 7 considerate pe care să le viziteze. Considerând că suma timpilor de zbor trebuie să fie de cel mult 13 ore, care este suma maximă a gradelor de satisfacție pe care o poate obține Ancuța?

(a) 44 (b) 40 (c) 45 (d) 46

19. Fie G un graf neorientat conex cu 7 noduri și muchiile: $(1, 2)$, $(2, 3)$, $(3, 1)$, $(3, 4)$, $(4, 5)$, $(5, 6)$, $(6, 4)$, $(4, 7)$. În câte moduri distincte putem elimina muchii din G astfel încât graful parțial obținut să fie un arbore?

(a) 8 (b) 9 (c) 6 (d) 2

20. Fie următoarea secvență de cod C/C++:

```
char multime_litere[] = { 'a', 'b', 'c', 'd', 'e', 'f' };
void inlocuire(int index, char caracter) {
    multime_litere[index] = caracter;
}
void prelucrare_elemente_multime() {
    char multime_auxiliara[] = { 'w', 'x', 'y', 'z' };
    for (int index = 0; index < 4; index++) {
        inlocuire((index * 10) % 6, multime_auxiliara[index]);
    }
    for (int index = 3; index >= 0; index--) {
        inlocuire(((index + 5) * 10) % 6, multime_auxiliara[index]);
    }
}
```

Care este conținutul vectorului `multime_litere` după apelul funcției `prelucrare_elemente_multime()`?

- (a) { 'x', 'b', 'z', 'd', 'y', 'f' }
- (b) { 'w', 'b', 'w', 'd', 'y', 'f' }
- (c) { 'x', 'b', 'x', 'd', 'y', 'f' }
- (d) { 'x', 'b', 'w', 'd', 'y', 'f' }

21. Managerul unei companii decide să angajeze 3 asistenți. Pentru fiecare asistent se stochează într-un tablou, în această ordine: costul orar, eficiența și disponibilitatea zilnică (numărul de ore de lucru pe zi). Tablourile corespunzătoare celor trei asistenți sunt:

```
int asistent_1[] = {10, 50, 2};
int asistent_2[] = {20, 60, 6};
int asistent_3[] = {15, 80, 4};
```

Productivitatea zilnică reprezintă produsul dintre eficiență și disponibilitatea zilnică. Costul zilnic al unui asistent este egal cu produsul dintre costul orar și productivitate.

Managerul stabilește următoarele moduri de lucru: modul normal (când se respectă valorile din tablouri), modul avarie (se înjumătățește valoarea eficienței) și modul suprasolicitare (se dublează valoarea eficienței).

Toți asistenții lucrează 20 de zile astfel: asistentul 1 lucrează în modul normal toate cele 20 de zile, asistentul 2 lucrează 10 zile în modul normal și 10 zile în modul avarie, iar asistentul 3 lucrează 10 zile în modul suprasolicitare și 10 zile în modul avarie.

Care este costul total (suma costurilor totale) pentru cei 3 asistenți pe durata celor 20 de zile?

(a) 250000 (b) 330000 (c) 248000 (d) 246000

22. Într-un sistem de compresie a textului se folosește un algoritm care construiește un arbore binar după următoarea regulă:

- pentru fiecare caracter distinct din text se creează inițial un nod-frunză, etichetat cu frecvența de apariție a caracterului în text;
- cât timp există cel puțin două noduri în mulțime:
 - se aleg două noduri cu frecvențele cele mai mici;
 - se creează un nou nod părinte, având frecvența egală cu suma frecvențelor celor două noduri alese;
 - noul nod părinte devine tatăl celor două noduri și este reintrodus în mulțime, în locul lor;
- la final rămâne un singur nod, care este rădăcina arborelui.

În arbore, muchia spre fiul stâng este notată cu 0, iar muchia spre fiul drept cu 1. Codul unui caracter este șirul de valori 0 și 1 de pe drumul de la rădăcină la frunza corespunzătoare acelui caracter.

Se consideră un text care folosește doar caracterele A, B, C, D, E, cu numărul de apariții:

$$A - 5, \quad B - 7, \quad C - 10, \quad D - 15, \quad E - 20.$$

Se aplică algoritmul descris mai sus, respectând exact pașii indicați. La fiecare pas, dacă există mai multe noduri cu aceeași frecvență minimă, se pot alege orice două dintre ele: pentru frecvențele date, toate modurile de alegere duc la aceeași lungime totală a textului comprimat.

Care este lungimea textului comprimat folosind codurile obținute din acest arbore?

(a) 126 (b) 120 (c) 130 (d) 114

23. Ce valoare are variabila s după execuția secvenței de cod C/C++?

```
int n = 0, s = 0, i;
for (i = 1; i <= 8; i++)
    if (++n % 3 == 0 || i % 2 == 0)
        s += i;
```

(a) 20 (b) 28 (c) 23 (d) 15

24. Ce valoare afișează următoarea secvență de cod C/C++?

```
int a[] = {3, 1, 4, 1, 5, 9, 2, 6}, n = 8;
int f(int &x) {
    x = x % n;
    return a[x]++;
}
int main() {
    int i = 0, s = 0, k = 0;
    while(k < 3) {
        s += f(a[i]);
        i = (i + a[i]) % n;
        k++;
    }
    cout << s;
}
```

(a) 9 (b) 13 (c) 6 (d) 11

25. Se dă următoarea secvență de cod C/C++:

```
int m[5][5] = {
    {1, 0, 1, 0, 1},
    {1, 1, 0, 0, 1},
    {0, 0, 0, 1, 0},
    {1, 1, 0, 1, 1},
    {0, 0, 1, 1, 0}
};
int a, b;

void fun(int mat[][5], int &i , int &j) {
    int done = 0;
    while (!done) {
        if (mat[i][j]) {
            j++;
        } else {
            i++;
        }
        if (i >= 5 || j > 4) done = 1;
    }
}
```

Care sunt valorile inițiale ale variabilelor a și b astfel încât, după apelul funcției `fun(m, a, b)`, expresia `a + b` are valoare maximă?

(a) `a = 1; b = 2;` (b) `a = 0; b = 2;` (c) `a = 1; b = 3;` (d) `a = 2; b = 3;`