

### Subiecte la testul grilă de Informatică

- Secvența de numere întregi  $S = (7, 1, 2, 5, 8, 10, 12, 3, 0, 9, 6)$  urmează a fi sortată crescător prin metoda bulelor. Care este rezultatul obținut după prima parcurgere a secvenței?
  - $S = (0, 1, 2, 3, 5, 6, 7, 8, 9, 10, 12)$
  - $S = (1, 2, 5, 7, 8, 10, 3, 0, 9, 6, 12)$
  - $S = (7, 1, 2, 5, 8, 10, 6, 3, 0, 9, 12)$
  - $S = (1, 7, 2, 5, 8, 10, 12, 3, 0, 9, 6)$
- Se consideră un graf neorientat cu 7 noduri și 3 componente conexe. Care este numărul minim de muchii ce trebuie adăugate astfel încât să devină conex?
  - 4
  - 6
  - 2
  - 5
- Cele 18 noduri ale unui graf neorientat sunt partiționate în trei mulțimi disjuncte, nevide, notate cu A, B și C. Aceste mulțimi au câte 5, 6, respectiv 7 noduri, astfel încât orice muchie din graf are extremitățile în mulțimi diferite. Care este numărul maxim de muchii pe care îl poate avea un astfel de graf?
  - 21
  - 210
  - 54
  - 107
- Fie matricea pătratică  $M$ :

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

Care este valoarea elementelor matricei  $M$ , după executarea funcției  $update(M)$ ?

```
void update(int M[4][4]) {  
    int i, j, tmp;  
    for (i = 0; i < 2; i++) {  
        for (j = 2; j < 4; j++) {  
            tmp = M[j][i];  
            M[j][i] = M[i][j];  
            M[i][j] = tmp;  
        }  
    }  
}
```

(a)  $\begin{bmatrix} 1 & 2 & 13 & 9 \\ 5 & 6 & 14 & 10 \\ 7 & 3 & 11 & 12 \\ 8 & 4 & 15 & 16 \end{bmatrix}$

(b)  $\begin{bmatrix} 1 & 2 & 9 & 13 \\ 5 & 6 & 10 & 14 \\ 3 & 7 & 11 & 12 \\ 4 & 8 & 15 & 16 \end{bmatrix}$

(c)  $\begin{bmatrix} 1 & 2 & 13 & 14 \\ 5 & 6 & 9 & 10 \\ 7 & 8 & 11 & 12 \\ 3 & 4 & 15 & 16 \end{bmatrix}$

(d)  $\begin{bmatrix} 1 & 2 & 9 & 10 \\ 5 & 6 & 13 & 14 \\ 3 & 4 & 11 & 12 \\ 7 & 8 & 15 & 16 \end{bmatrix}$

5. Se consideră două mulțimi  $A$  și  $B$ , fiecare fiind formată din câte 5 elemente. Numărul funcțiilor bijectiv definite pe  $A$  cu valori în  $B$  este egal cu:

- (a) 32   (b) 5   (c) 120   (d) 3125

6. Fie următoarele afirmații:

1. Într-un graf neorientat conex cu numărul de noduri  $n \geq 2$ , cel puțin două noduri au același grad.
2. Un graf neorientat cu  $2n$  noduri, în care fiecare nod are gradul  $n$  este format din cel puțin două componente conexe.
3. Într-un graf neorientat conex cu 115 noduri, fiecare nod este conectat cu exact alte 11 noduri.
4. Într-un graf neorientat există lanț de la fiecare nod de grad impar la un alt nod de grad impar.

Care dintre aceste afirmații nu sunt adevărate?

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

7. Care dintre următoarele secvențe realizează ordonarea crescătoare a elementelor tabloului  $a$ , cu  $n$  elemente:

```
(a)
for (i = 0; i < n ; i++)
  for (j = i + 1; j < n; j++)
    if (a[i] < a[j]) {
      x = a[i];
      a[i] = a[j];
      a[j] = x;
    }
```

```
(c)
for (i = 0; i < n - 1; i++)
  for (j = i + 1; j < n; j++)
    if (a[i] < a[j]) {
      x = a[i];
      a[i] = a[j];
      a[j] = x;
    }
```

```
(b)
for (i = 0; i < n - 1; i++)
  for (j = i + 1; j < n; j++)
    if (a[i] > a[j]) {
      x = a[i];
      a[i] = a[j];
      a[j] = x;
    }
```

```
(d)
for (i = 0; i < n; i++)
  for (j = i + 1; j < n; i++)
    if (a[i] > a[j]) {
      x = a[i];
      a[i] = a[j];
      a[j] = x;
    }
```

8. Fie următoarea funcție:

```
int f(int &X) {
  int sum = 0;
  while(X != 0) {
    sum += X % 2;
    X /= 2;
  }
  return sum;
}
```

Considerând o variabilă globală de tip întreg  $n = 32$ , ce se va afișa în urma instrucțiunii: `cout << f(n) + f(n) + 1`?

- (a) 4   (b) 2   (c) 3   (d) 1

9. Care este valoarea returnată de funcția de mai jos atunci când este apelată cu valoarea 20002021?

```

int do_it(int n) {
    int r = 0;
    while (n) {
        r += !(n%100);
        n /= 10;
    }
    return r;
}

```

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

10. Care este valoarea returnată după apelul următoarei funcții recursive? La apelul inițial al funcției, tabloul de numere întregi folosit pentru primul argument ( $x$ ) are valorile: 10, 20, 30, 10, 20, 30 și al doilea argument ( $i$ ) are valoarea 4.

```

int fun(int x[], int i) {
    if (i >= 1)
        return x[i] + fun(x, i - 1);
    else
        return x[0];
}

```

(a) 90 (b) 70 (c) 120 (d) 110

11. Se consideră toate șirurile de lungime  $l \in \{1, 2, 3\}$  formate din litere din mulțimea  $\{e, f, g, h, i\}$ . Precizați câte dintre aceste șiruri au elementele ordonate strict crescător (alfabetic) și un număr impar de vocale.

(a) 14 (b) 10 (c) 28 (d) 7

12. Fie vectorul  $v$  și funcția recursivă  $f$ :

```

int v[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

```

```

void f(int x) {
    if (x < 9) {
        v[x] = 0;
        f(v[x + 1]);
    }
}

```

Cum va arăta vectorul, dacă funcția  $f$  primește la apel valoarea 1 (obs. primul element al vectorului are indicele 0)?

- (a) 0 2 0 4 0 6 0 8 0 10  
 (b) 1 0 3 0 5 0 7 0 9 0  
 (c) 0 0 0 0 0 0 0 0 0 0  
 (d) 1 0 3 0 5 0 7 0 9 10

13. Se consideră următorul algoritm în care  $nr$ ,  $x$  și  $y$  sunt numere naturale, unde  $[a]$  reprezintă partea întreagă numărului  $a$ :

```

citeste x, y ;
nr ← 0;
d ← 2;
cât timp d ≤ x and d ≤ y execută
┌   dacă x % d = 0 and y % d = 0 atunci
│   │   nr ← nr + 1;
│   │   x ← [x/d];
│   │   y ← [y/d];
│   altfel
│   └   d ← d + 1;
└
Scrie nr, x, y

```

Determinați câte perechi  $(x, y)$  de valori de cel mult 2 cifre există, astfel încât să se afișeze valorile 1, 7, 11.

- (a) 8   (b) 5   (c) 2   (d) 4

14. O imagine alb-negru este codificată sub forma unei matrice cu  $N \times N$  elemente de 0 și 1, unde 0 reprezintă un pixel negru și 1 un pixel alb. Asupra acestei imagini se pot aplica următoarele operații:

- **R** - rotire cu  $90^\circ$  în sensul acelor de ceasornic;
- **I** - inversare, adică valorile de 1 se transformă în 0 și invers;
- **D** - dublare, se va obține o nouă imagine cu dimensiunea  $2N \times 2N$  prin dublarea fiecărei linii și după aceea prin dublarea fiecărei coloane din matricea aferentă imaginii.

Dacă asupra imaginii  $\text{Img} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$  se aplică următoarele operații: **R, R, R, I, R, D, I**, care va fi matricea corespunzătoare imaginii rezultat?

- (a)  $\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$    (b)  $\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$    (c)  $\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$    (d)  $\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$

15. Un palindrom este un șir care citit de la stânga la dreapta și de la dreapta la stânga arată identic. Determinați câte numere prime de tip palindrom sunt în intervalul  $[10, 300]$ .

- (a) 5   (b) 6   (c) 10   (d) 4

16. Pentru un graf neorientat  $G$  se definește o secvență în care fiecare număr reprezintă gradul unui nod al grafului. Care dintre secvențele următoare pot fi asociate unui graf neorientat conex?

1. (1,1,2,3,3,4,6)
2. (2,2,3,3,3,4,4,4,4)
3. (1,2,3,4,6)
4. (1,1,2,3,3,0,4,6)

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

17. Se consideră șirul  $(1, 2, 3, 2, 5, 2, 3, 7, 2, 4, 3, 2, 5, 11, \dots)$  format astfel: plecând de la șirul numerelor naturale, se înlocuiesc numerele care nu sunt prime cu divizorii lor proprii, fiecare divizor  $d$  fiind considerat o singură dată pentru fiecare număr. Care dintre funcțiile de mai jos determină al  $n$ -lea element al acestui șir ( $n$  - număr natural,  $n \in [2, 100]$ )? (DIV calculează câtul împărțirii și MOD calculează restul împărțirii)

**funcția  $V1(n)$**

```
a ← 1; b ← 1; c ← 1;
cât timp c < n execută
  a ← a + 1; b ← a;
  c ← c + 1; d ← 2; f = false;
  cât timp c ≤ n și d ≤ a DIV 2 execută
    dacă a MOD d = 0 atunci
      c ← c + 1; b ← d; f ← true;
      d ← d + 1;
    dacă f atunci
      c ← c - 1;
returnează b;
```

**funcția  $V2(n)$**

```
a ← 1; b ← 1; c ← 1;
cât timp c < n execută
  c ← c + 1; d ← 2;
  cât timp c ≤ n și d ≤ a DIV 2 execută
    dacă a MOD d = 0 atunci
      c ← c + 1; b ← d;
      d ← d + 1;
returnează b;
```

**funcția  $V3(n)$**

```
a ← 1; b ← 1; c ← 1;
cât timp c < n execută
  b ← a; a ← a + 1;
  c ← c + 1; d ← 2;
  cât timp c ≤ n și d ≤ a DIV 2 execută
    dacă a MOD d = 0 atunci
      c ← c + 1; b ← d;
      d ← d + 1;
returnează b;
```

(a)  $V2$  și  $V3$    (b)  $V2$    (c)  $V1$  și  $V2$    (d)  $V1$

18. Se dă următoarea secvență de cod:

```
int i, j, k, n=5, m=5;
int a [[5] = {{1,1,1,1,1},{2,2,2,2,2},{3,3,3,3,3},{4,4,4,4,4},{5,5,5,5,5}};
for(k=0; k < n/2; k++)
{
  i = k;
  j = k;
  for(; j < m-k; j++)
    cout << a[i][j] << " ";
  --j;
  ++i;
  for(; i < n-k-1; i++)
    cout << a[i][j] << " ";
  for(; j >= k; j--)
    cout << a[i][j] << " ";
  j++;
  --i;
  for(; i > k; --i)
    cout << a[i][j] << " ";
}
if(n % 2)
  for(++j; j < m-k; ++j)
    cout << a[k][j] << " ";
```

Ce se afișează la execuție?

- (a) 1 1 1 1 1 2 3 4 5 5 5 5 5 4 3 2 2 2 2 3 4 4 4 3 2
- (b) 1 1 1 1 1 2 3 4 5 5 5 5 5 4 3 2 2 2 2 3 4 4 4 3 3
- (c) 1 2 3 4 5 5 5 5 5 4 3 2 1 1 1 1 2 3 4 4 4 3 2 2 3
- (d) 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5

19. Fie funcția *fff* în care se apelează funcția *cmmdc* care returnează cel mai mare divizor comun al celor două numere transmise ca argumente.

```
void fff(unsigned int a[], unsigned int fi[], unsigned int m) {
    int i, j, contor;
    for(i = 0; i < m; i++) {
        contor = 1;
        for(j = 2; j < a[i]; j++)
            if(cmmdc(j, a[i]) == 1)
                contor++;
        fi[i] = contor;
    }
}

int main() {
    unsigned int x[] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
    unsigned int fi[50];
    unsigned int m = sizeof(x) / sizeof(x[0]);
    ... //trebuie completat
    int i;
    for(i = 0; i < m; i++)
        cout << fi[i] << " ";
    cout << endl;
    return 0;
}
```

Ce trebuie scris în funcția *main* în locul celor 3 puncte pentru ca funcția *fff* să fie apelată corect și ce se afișează în urma execuției programului?

- (a) Apelul corect este: `fff(x, fi, m);`  
pe monitor se afișează: 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
- (b) Apelul corect este: `fff(x, fi, m);`  
pe monitor se afișează: 1 1 2 2 4 2 6 4 6 4 10 4 12 6 8
- (c) Apelul corect este: `fi = fff(x,fi, m);`  
pe monitor se afișează: 1 1 2 2 4 2 4 4 6 4 8 8 10 6 8
- (d) Apelul corect este: `fff(x[], fi[], m);`  
pe monitor se afișează: 1 1 2 2 4 2 6 4 6 4 10 4 12 6 8

20. Se consideră următoarea structură care stochează numele și media studenților dintr-o grupă, în ordine alfabetică.

```
typedef struct{
    char nume[256];
    float media;
}student;

typedef student grupa[30];
grupa A;
```

Specificați care este varianta corectă pentru a accesa inițiala numelui primului student din grupă.

- (a) `grupa[0].nume`    (b) `A[0].student.nume`    (c) `A[0].nume[0]`    (d) `A[0].student.nume[0]`

21. Precizați câte dintre următoarele expresii au valoarea 1 (true) oricare ar fi numărul natural nenul  $x$ , care nu este divizibil cu 10:

- 1)  $x/10! = 0$
- 2)  $x\%10! = 0$
- 3)  $x\%2! = 0 \ \&\& \ x\%5! = 0$
- 4)  $x/10 + x\%10 > 0$
- 5)  $x\%2 + x\%5 > 0$
- 6)  $x\%10 > 10$

(a) 1   (b) 3   (c) 2   (d) 4

22. Fie programul de mai jos:

```
int n = 10, i = 5, d;
void functie (int n, int& d) {
    for(i = 2; i < n; i++)
        if(n%i == 0) d++;
}

int main() {
    functie(4, d);
    cout << d;
    return 0;
}
```

Care dintre afirmațiile de mai jos este adevărată:

- (a) Funcția nu este apelată corect.
- (b) Valoarea lui  $d$  se calculează în funcție și este afișată în funcția *main*.
- (c) Valoarea lui  $d$  se calculează în funcție, dar rezultatul afișat va fi imprevizibil.
- (d) Valoarea lui  $d$  nu este cunoscută la apelul funcției, deci rezultatul afișat va fi imprevizibil.

23. Fie  $G = (V, A)$  un graf neorientat, complet, cu 20 noduri. Care este numărul maxim de muchii care pot fi eliminate astfel încât  $G$  să aibă un ciclu elementar care conține toate nodurile?

(a) 95   (b) 180   (c) 20   (d) 170

24. Se consideră următoarea funcție care primește ca date de intrare un vector  $v$  cu  $n$  elemente, sortat, un indice de start  $i \in \{1 \dots n\}$ , un indice de stop  $j \in \{1 \dots n\}$  și o valoare  $x$  de căutat în vector:

```
funcția f( $v, i, j, x$ )
|
| dacă  $j \geq i$  atunci
| |  $m \leftarrow i + (j - i)/2$ ;
| | dacă  $v[m] = x$  atunci
| | | returnează  $m$ ;
| | altfel
| | | dacă  $C1$  atunci
| | | | returnează  $f(v, m+1, j, x)$ ;
| | | | returnează  $f(v, i, m-1, x)$ ;
| | returnează  $0$ ;
```

Care este condiția de test  $C1$  și care este modul în care vectorul este sortat, dacă dorim ca funcția  $f$  să implementeze corect *căutarea binară*?

- (a)  $v[m] > x$ , dacă vectorul este sortat descrescător.
- (b)  $v[m] < x$ , dacă vectorul este sortat descrescător.
- (c)  $v[m] > x$ , dacă vectorul este sortat crescător.
- (d)  $v[m] < x$ , indiferent de direcția de sortare a vectorului.

25. Într-o clasă sunt 25 de elevi, iar mediile la cele 6 materii dintr-un semestru sunt stocate într-un tablou de structuri în care fiecare element este definit astfel:

```
struct media {
    char nume[50];
    char materia[25];
    int nota;
};
struct media clasa[150];
```

Care este definiția corectă pentru o funcție care calculează media clasei la o anumită materie precizată ca parametru?

(a)

```
float calculeaza(char d[]) {
    int i, m = 0;
    for (i = 0; i <= 150; i++)
        if (strcmp(d,clasa[i].materia)==0)
            m += clasa.nota[i];
    return(m/25.0);
}
```

(c)

```
float calculeaza(char d[]) {
    int i, m = 0;
    for (i = 0; i < 150; i++)
        if (clasa[i].materia==d)
            m += clasa[i].nota;
    return(m/25.0);
}
```

(b)

```
float calculeaza(char d[]) {
    int i, m = 0;
    for (i = 0; i < 150; i++)
        if (strcmp(d,clasa[i].nume)==d)
            m += clasa.nota[i];
    return(m/25.0);
}
```

(d)

```
float calculeaza(char d[]) {
    int i, m = 0;
    for (i = 0; i < 150; i++)
        if (strcmp(d,clasa[i].materia)==0)
            m += clasa[i].nota;
    return(m/25.0);
}
```