Universitatea Tehnică "Gheorghe Asachi" din Iași
Facultatea de Automatică și Calculatoare
Admitere – sesiunea iulie 2022
Domeniile: Calculatoare și tehnologia informației
Ingineria sistemelor (Automatică și informatică aplicată)

*

Subiecte la testul grilă de Informatică

1. Se consideră următoarea secvență de cod C++:

```cpp
int x = 13;

void f( void ) {
    int x;
    x = 10;
    cout << "x: " << x << endl;
}
```

Este corect scrisă funcția f()? Dacă da, ce valoare se afișează? Dacă nu, care este greșeala?

(a) Funcția este scrisă corect. Se afișează valoarea 10.

(b) Funcția nu este scrisă corect pentru că variabilele locale nu pot avea același nume cu cele globale.

(c) Funcția este scrisă corect. Se afișează valoarea 13.

(d) Funcția nu este scrisă corect pentru că variabilele globale nu pot fi inițializate în momentul definirii.

2. Se consideră funția de mai jos:

```cpp
int Calc(int n)
{
    return ( n / 1000 ) * 100 + n % 100 ;
}
```

Care este valoarea care se va returna la apelul Calc(98765); ?

(a) 9765 (b) 9876 (c) 98065 (d) 9865

3. Ce valoare afiseaza pe ecran urmatoarea secventa de program

```cpp
int x, y = 5, z = 5;
x = y == z;
cout<< x;
```

(a) 5 (b) 1 (c) 0 (d) nedefinit

4. Ce se afișează la execuția următoarei secvențe de cod?

```cpp
int v[] = {0, 1};
int w[] = {0, 1};
int i, j, p, q, expresie;
for(i=0; i<2; i=i+1)
{
    for(j=0; j<2; ++j)
    {
        p = v[i];
```

```
            q = w[j];
            expresie = !(p || q) || (p && !q);
            cout << p << " " << " " << q << " " << expresie << endl;
        }
    }
```

(a)

0	0	1
0	1	0
1	0	1
1	1	0

(b)

0	0	0
0	1	0
1	0	0
1	1	0

(c)

0	0	0
0	1	1
1	0	0
1	1	1

(d)

0	0	0
0	1	1
1	0	1
1	1	1

5. Fie programul de mai jos. De cate ori se va afișa pe ecran "Admitere AC@TUIASI" ?

```
int main()
{
    int x;
        for(x=-1; x<=10; x++)
        {
                if(x < 5)
                    continue;
                else
                    break;
            cout << ("Admitere AC@TUIASI");
    }
    return 0;
}
```

(a) 10 ori; (b) 11 ori (i porneste de la -1); (c) 0 ori; (d) de un numar infinit de ori.

6. Se consideră două numere naturale a și b ($1 \leq a < b \leq 1000$) si secventa de mai jos:

```
m ← 0;
pentru n ← a la b execută
    c ← 0;
    pentru d ← 1 la n execută
        dacă n MOD d = 0 atunci
            c ← c + 1;
    dacă c > m atunci
        m ← c;

pentru n ← a la b execută
    c ← 0;
    pentru d ← 1 la n execută
        dacă n MOD d = 0 atunci
            c ← c + 1;
    dacă c = m atunci
        scrie n;
```

Precizați care din afirmațiile de mai jos este corectă.

(a) Algoritmul afișează numărul de divizori pentru fiecare număr natural din intervalul [a, b].

(b) Algoritmul afișează numerele naturale din intervalul [a, b] care au proprietatea că au cel mai mare număr de divizori proprii.

(c) Algoritmul afișează numerele naturale din intervalul [a, b] care au proprietatea că au cel mai mare număr de divizori

(d) Algoritmul afișează maximul dintre numărul de divizori ai lui a și numărul de divizori ai lui b.

7. Se consideră o matrice pentru care primul element (prima linie si prima coloana) este $a_{1,1}$. Elementele matricei de sub diagonala principală, în parcurgere pe linie, se copie intr-un vector. Se considera ca primul element din vector este pe pozitia 1. Care este relația dintre pozitia k in vector și indecsii elementelor matricii $a_{i,j}$ (i linie, j coloana)?
   (a) k=(i-1)*(j-2)/2+j
   (b) k=(i-2)*(i-1)/2+j
   (c) k=(j-2)*(j-1)/2+i
   (d) k=(i-2)*(j-1)/2+i

8. Se consideră definițiile pentru tablourile și funcția de mai jos:

```
char TA[][4] = {
  {'R', 'C', 'A', 'C'},
  {'D', 'D', 'R', 'R'},
  {'C', 'D', 'C', 'C'},
  {'C', 'C', 'A', 'D'},
};

int vx[] = {0, 1, 0, -1} ;
int vy[] = {1, 0, -1, 0} ;

int f (char c, int N) {
    int i, j, k, flag, vi, vj, ret = 0;
    for (i = 0; i < N; i++) {
        for (j = 0 ; j < N; j++) {
            flag = 0;
            for (k = 0; k < 4; k++) {
                vi = i + vx[k];
                vj = j + vy[k];
                if ((vi >=0 ) && (vi < N) && (vj >=0 ) && (vj < N)) {
                    if (TA[vi][vj] == c) {
                        flag = 1;
                    }
                }
            }
            if (TA[i][j] == c && !flag) {
                ret++;
            }
        }
    }
    return ret;
}
```

Care este valoarea returnată de apelul funcției f('C', 4) ?

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

9. Fie patru vectori v_1, v_2, v_3 și v_4 de dimensiune n, m, m și, respectiv, n, sortați crescător. Se dorește obținerea unui vector sortat care să conțină elementele celor patru vectori. Pentru aceasta se interclasează vectorii doi câte doi.

Considerăm că $\oplus$ reprezintă operația de interclasare, iar interclasarea vectorilor se realizează după relația:

$((v_1 \oplus v_2) \oplus v_3) \oplus v_4$.

Care este complexitatea timp a interclasării celor patru vectori?

(a) $\mathcal{O}(n+m)$ (b) $\mathcal{O}(n \cdot m)$ (c) $\mathcal{O}(n^2+m^2)$ (d) $\mathcal{O}(n^2 \cdot m^2)$

10. Care este numărul de interschimbări necesare pentru a sorta n elemente utilizând metoda selecției, în cel mai defavorabil caz?

    (a) $O(n)$ (b) $O(n * log(n))$ (c) $O(n * n)$ (d) $O(log(n))$

11. Numărul de submulțimi al unei mulțimi finite $M = \{a_1, a_2, \ldots, a_7\}$ este:

    (a) 128 (b) 49 (c) 7 (d) 1

12. Fie 7 tipuri de fructe (prune, portocale, mere, mandarine, ananas, smochine, clementine) și un program care aplică algoritmul backtracking pentru a genera un compot alcătuit din minim 3 fructe. Știind că niciodată nu putem avea două fructe care încep cu aceeași literă în compot, să se indice ce variante de amestec urmează în șirul generat de algoritmul în descris:
    - șir 1 : prune, mere, ananas
    - șir 2 : prune, mere, ananas, smochine
    - șir 3 : prune, mere, ananas, smochine, clementine

    (a) **șir 4**: portocale, mere, ananas **șir 5**: portocale, mere, ananas, smochine
    (b) **șir 4**:prune, mere, ananas, clementine **șir 5**: prune, mere, ananas, clementine, smochine
    (c) **șir 4**: prune, mandarine, ananas **șir 5**: prune, mandarine, ananas, smochine
    (d) **șir 4**: prune, mere, clementine **șir 5**: prune, mere, clementine, smochine

13. Care din următoarele variante de funcții determină restul împărțirii întregi a lui x la y?

$$f1(x,y) = \begin{cases} x & \text{pentru x<y} \\ f1(x-y,y) & \text{pentru x>=y} \end{cases}$$

$$f2(x,y) = \begin{cases} y & \text{pentru x>y} \\ f2(x-y,y) & \text{pentru x<=y} \end{cases}$$

$$f3(x,y) = \begin{cases} y & \text{pentru x<=y} \\ f3(y,x-y) & \text{pentru x>y} \end{cases}$$

$$f4(x,y) = \begin{cases} x & \text{pentru x<=y} \\ f4(x,x-y) & \text{pentru x>y} \end{cases}$$

    (a) f2
    (b) f1
    (c) f3
    (d) f4

14. Care este modalitatea cea mai compactă de a scrie complexitatea din punctul de vedere al duratei de execuție (complexitatea timp) a secvenței de cod de mai jos, dacă $m > n$?

```
int s = 100;
int i, j;
for (i = 0; i < n; ++i) {
    for (j = i + 1; j <= m; ++j) {
        s = s + j / 2;
    }
}
for (i = n; i < m; ++i) {
    s++;
}
```

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

15. Se consideră următorii algoritmi de sortare: algoritmul bulelor, algoritmul inserției directe şi algoritmul selecției prin numărare. Care dintre perechile următoare este formată din algoritmi de sortare care nu au complexitatea timp $O(n^2)$ în cazul cel mai nefavorabil?
    (a) Algoritmul bulelor, Algoritmul inserție directe
    (b) Algoritmul bulelor, Algoritmul selecției prin numărare
    (c) Algoritmul inserție directe, Algoritmul selecției prin numărare
    (d) Niciuna dintre perechile posibile

16. Pentru conectarea pe un site, orice utilizator trebuie sa se inregistreze cu o parola formata din n cifre. Ulterior, pentru logare, aplicatia genereaza aleator trei pozitii distincte din parola, p1, p2 și p3, astfel încât $1 \leq p1 < p2 < p3 \leq n$ iar utilizatorul trebuie să introducă doar cifrele de pe acele 3 poziții. De exemplu, dacă codul utilizatorului este 10234056789 și pagina generează aleator pozițiile 3, 6 și 8, utilizatorul trebuie să introducă cifrele 2, 0, 6. Pentru 9 logări valide pe acel site, un utilizator introduce urmatoarele cifre :

1, 4, 7	9, 3, 2	4, 4, 3
6, 3, 2	4, 3, 1	5, 6, 0
2, 0, 2	1, 2, 3	2, 9, 0

Daca codul utilizatorului nu a fost schimbat între timp, precizați care dintre următoarele afirmații sunt adevărate.
    (a) Codul utilizatorului sigur nu conține cifra 8.
    (b) Cel mai scurt cod posibil are 12 cifre.
    (c) Cel mai scurt cod posibil conține cifra 2 de minimum 3 ori.
    (d) Suma cifrelor în cel mai scurt cod posibil poate fi 38.

17. La un concert, pentru a crea atmosfera, liderul formației invită pe scenă spectatori selectați din primul rând cu n locuri, ocupat în întregime, după următoarea regulă: se numără fiecare al k-lea (0<k<n) spectator din cei n, numărând din k în k. Când numărătoarea ajunge la spectatorul de pe ultimul loc de pe rând se continuă cu primul spectator, numărând mai departe toți spectatorii, fie că au fost sau nu selectați. Numărătoarea se oprește atunci când ar trebui selectat un spectator deja selectat. Câți spectatori rămân neselectați dacă n=33 și k=6?
    (a) 11
    (b) 22
    (c) 25
    (d) 13

18. Fie codul de mai jos:

```
#include <iostream>
using namespace std;

int x[10],n=5, m=4;
char instrumente[][7]={"Lopata","Sapa","Topor","Cazma","Grebla"};

int Validare(int k) {
    for(int i=0;i<k;i++)
```

```
        if(x[i]>=x[k]) return 0;
        return 1;
}

void Afisare(int k) {
        for(int i=0;i<m;i++)
        cout<<instrumente[x[i] - 1]<<" ";
        cout<<endl;
}

void Functie(int k) {
        for(int i = 0 ; i < n ; ++i) {
                x[k]=n-m+i;
                if( Validare(k) )
                        if(k>=m-1)
                                Afisare(k);
                        else
                                Functie(k+1);
        }
}

int main(){
        Functie(0);
        return 0;
}
```

Care este cea de a treia soluție afișată?

(a) Sapa Topor Cazma Grebla

(b) Lopata Topor Cazma Grebla

(c) Sapa Topor Lopata Grebla

(d) Lopata Sapa Cazma Grebla

19. Fie un arbore cu o rădăcină și un număr total de 31 de noduri, în care fiecare nod poate avea 2, 1 sau 0 fii. Dacă în arbore sunt 12 noduri de grad 2, care este numărul nodurilor de grad 1? Gradul unui nod reprezintă numărul de fii ai acestuia.

(a) 4 (b) 9 (c) 11 (d) 6

20. Fie un arbore cu o rădăcină și mai multe niveluri, în care fiecare nod (inclusiv rădăcina) are zece descendenți direcți (fii). În acest arbore se introduc un milion de valori consecutive în ordine crescătoare începând cu valoarea 0. Prima valoare (0) va fi introdusă în rădăcină, iar următoarele vor ocupa, în ordine, nodurile descendente rădăcinei de la stânga la dreapta. După ce un nivel este complet, se trece la următorul nivel, unde nodurile vor fi populate cu valori, la fel, de la stânga la dreapta.

Dacă se dorește a căuta o valoare mai mică de un milion în acest arbore, în cazul cel mai defavorabil, un algoritm eficient de căutare câte valori din arbore ar trebui să acceseze?

Observație: Căutarea unei valori între fii unui nod se poate realiza accesând un singur element.

(a) 7 (b) 6 (c) 10 (d) 100000

21. Fie G un graf neorientat cu n noduri. Fiecare nod are un identificator notat cu $1, 2, ..., n$. În graf există muchii:

- între oricare două noduri cu identificatori pari
- între oricare două noduri cu identificatori impari

- între oricare două noduri cu identificatori de forma $2k$, respectiv $2k+1$

Dacă n este impar, câte muchii conține graful G?

(a) $\frac{n^2-1}{4}$ (b) $\frac{n^2}{4}-2$ (c) $\frac{n^2}{2}$ (d) n^2

22. Fie A un arbore binar plin cu n noduri. Dacă la toate nodurile interne se taie muchia către succesorul din dreapta, care este adâncimea celui mai adânc arbore dintre toți arborii care se formează?
Notă 1: Un arbore binar este plin dacă fiecare nivel al acestuia este complet ocupat.
Notă 2: În cadrul unui arbore, adâncimea unui nod este egală cu numărul de muchii de la acel nod până la rădăcină (rădăcina are adâncimea 0). Adâncimea arborelui este dată de adâncimea celui mai adânc nod.

(a) $\frac{n}{2}$ (b) $\frac{n+1}{2}$ (c) $log_2(n+1)$ (d) $log_2\frac{n+1}{2}$

23. Fie G un graf neorientat format din 3 componente conexe cu 3, 4 și, respectiv, 5 noduri. Dacă cele 3 componente conexe conțin minimul posibil de muchii, care este numărul maxim de muchii care pot fi adăugate în graf astfel încât numărul de componente conexe să fie 2?

(a) 19 (b) 30 (c) 12 (d) 10

24. Se consideră o stivă şi o coadă iniţial vide. Se introduc pe rând în coadă primele 7 pătrate perfecte nenule, în ordine descrescătoare. Se extrag apoi din coadă două elemente și se adaugă în stivă, în ordinea în care au fost extrase. Care este diferența dintre valoarea elementului din vârful stivei și valoarea primului element din coadă, după executarea acestor operaţii?

    (a) 24
    (b) 11
    (c) 9
    (d) 13

25. Fie codul de mai jos:

```
#include <iostream>
using namespace std;

unsigned fct (unsigned n){
   unsigned x;
    if (n < 0) return -n;
    else
       if (n == 0) return 5;
       else {
            x=fct(n/10)-1;
            cout<<"x="<<x<<" ";
            return x;
        }
  }

int main ( ) {
   unsigned x=1918;
   cout<<fct(x);
   return 0 ;
}
```

Care sunt valorile afișate pentru x în funcția fct?

(a) x=4 x=3 x=2 x=1 (b) x=5 x=5 x=5 x=5 1 (c) x=5 x=4 x=3 x=2 (d) x=4 x=3 x=2 x=1 0