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

*

Subiecte la testul grilă de Informatică

1. Se consideră următoarea expresie logică (X OR Z) AND (NOT ((X OR Y) AND Z)). Prin OR se înțelege *sau* logic, AND înseamnă *și* logic, NOT înseamnă negare. Fie valorile pentru X, Y, Z:

   I. X ← False; Y ← True; Z ← False ;
   II. X ← True; Y ← False; Z ← False;
   III. X ← False; Y ← False; Z ← False;
   IV. X ← False; Y ← False; Z ← True;

   Alegeți combinația corectă astfel încât evaluarea expresiei să dea rezultatul True.

   (a) I (b) II (c) I și III (d) II și IV

2. Care este valoarea afișată după execuția următoarei secvențe de cod?

```
char sirA[] = "Admitere2024";
char sirB[100] = "B";
char p[100];
strcpy(p, sirA + 3);
strcat(sirB, p + 2);
cout << strlen(sirB);
```

   (a) 10 (b) 8 (c) 9 (d) 7

3. Ce se afișează în urma rulării secvenței:

```
char s[100];
int i;
strcpy(s, "Testinformatica");
for(i = 0; i < strlen(s) - 1; i++)
    if(strchr("AutomaticasiCalculatoare", s[i + 1]))
        strcpy(s + i, s + i + 1);
cout << s;
```

   (a) Tsinfraia (b) etinomtca (c) etnfomtca (d) etinomtc

4. Fie secvența de mai jos. Indicați un algoritm care să facă același lucru, știind că vectorul inițial are valori generate aleatoriu și că se dorește ca algoritmul ales să fie cât mai eficient. Prin $\leftrightarrow$ se înțelege interschimbarea celor 2 valori.

```
pentru i = n, 2 execută
    pentru j = 1, n − 1 execută
        dacă a[j] < a[j + 1] atunci
            a[j + 1] ↔ a[j]
```

   (a) Ciurul lui Eratostene. (b) Sortarea prin inserție (c) Sortare prin interclasare (d) Algoritmul lui Euclid

5. Câte comparații ale valorilor elementelor vectorului v sunt efectuate de funcția process la apelul process(v, 8, 85)?

```
int v[] = {10, 25, 30, 45, 50, 65, 80, 95};

int process(int v[], int n, int x) {
    int i;
    for (i = 0; i < n; i++) {
        if (v[i] == x) return i;
    }
    return i;
}
```

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

6. Un cămin de studenți are 4 etaje, iar pe fiecare etaj sunt 5 camere. În fiecare cameră stau maxim 4 studenți. Pentru a ține această evidență se folosește o matrice mat, cu indecșii primului element 0, 0. Matricea, precum și numărul total de studenți se salvează într-o structură Camin, definită astfel:

```
struct Camin
{
    int mat[4][5];
    int nrStudenti;
};
```

Dacă se dorește afișarea numărului de studenți aflați în camerele de la parter, ce instrucțiuni se pot folosi? Pentru accesarea elementelor matricii a fost declarată o variabilă c de tip Camin, iar elementele structurii au fost inițializate corect.

(a)
```
for (i = 0; i < 5; i++)
{
    cout << c.mat[i][0] << " ";
}
```

(b)
```
for (i = 0; i < 5; i++)
{
    cout << c.mat[0][i] << " ";
}
```

(c)
```
for (i = 0; i < 5; i++)
{
    cout << mat[i][0] << " ";
}
```

(d)
```
for (i = 0; i < 5; i++)
{
    cout << c.mat[i][0].nrStudenti << " ";
}
```

7. Care este indexul elementului cu valoarea 5 în tabloul v după apelul funcției transform(v, 8) ?

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

void transform(int v[], int n) {
    int j = n - 1;
    for (int i = 0; i < j; i++) {
        int temp = v[i];
        v[i] = v[j];
        v[j] = temp;
        j--;
    }
}
```

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

8. Vlad vrea să joace un joc cu prietenul său Mihai, și se gândește să aplice un algoritm pentru amestecarea filmelor preferate ale lui Mihai. Acestea sunt stocate inițial în următorul vector:

```
char filme[][50]={ "film_actiune" , "film_comedie" , "film_drama" , "film_western" } ;
```

Fie funcțiile:

```
void swap_element ( char filme[][50] , int index_1 , int index_2 )
{
    char aux[50];
    strcpy(aux,filme [index_1]) ;
    strcpy(filme [ index_1 ], filme [ index_2 ]) ;
    strcpy(filme [ index_2 ], aux) ;
}

void procesare ( char filme [ ][50] )
{
    int valoare = 32 ;
    for ( int index = 0 ; index < 4 ; index++)
    {
        valoare = valoare * 4 ;
        valoare = valoare = 5 ;
        valoare = valoare % 50 ;
    }
    swap_element ( filme , valoare % 4 , ( valoare % 4 + 2) % 4 ) ;
    swap_element ( filme , ( valoare % 4 + 1) % 4 , ( valoare % 4 + 3) % 4 ) ;
    swap_element ( filme , ( valoare % 4 + 2) % 4 , valoare % 4 ) ;
    swap_element ( filme , ( valoare % 4 + 3) % 4 , ( valoare % 4 + 1) % 4 ) ;
}
```

Care este ordinea elementelor din vectorul *filme* după rularea funcției *procesare*(*filme*)?

(a) `filme[]={"film_comedie", "film_actiune", "film_drama", "film_western"}`

(b) `filme[]={"film_actiune", "film_drama", "film_comedie", "film_western"}`

(c) `filme[]={"film_western", "film_comedie", "film_drama", "film_actiune"}`

(d) `filme[]={"film_actiune", "film_comedie", "film_drama", "film_western"}`

9. Se dorește procesarea următorului șir de 8 numere naturale:

```
int sir[8] = {0, 1, 2, 3, 4, 5, 6, 7};
```

Care este valoarea returnată de apelul funcției *procesare_sir*(*sir*)?

```
void procesare_sir(int sir[])
{
    int sir_2[8];
    int sir_3[8];

    for (int index = 0; index < 8; ++index)
    {
        sir_2[7 - index] = sir[index];
    }
    for (int index = 0; index < 4; ++index)
    {
        sir_3[index] = ((sir[index] * 2) + sir_2[index]);
    }
    for (int index = 4; index < 8; ++index)
```

```
    {
        sir_3[index] = (sir[index] + (sir_2[index] * 4));
    }
    for (int index = 0; index < 8; ++index)
    {
        sir[index] = (2 * sir_2[index] + 3 * sir_3[index]);
    }
}
```

(a) `sir[] = {35, 36, 37, 38, 54, 43, 32, 21};`

(b) `sir[] = {35, 36, 38, 37, 54, 43, 32, 21};`

(c) `sir[] = {35, 36, 37, 38, 54, 43, 21, 32};`

(d) `sir[] = {35, 37, 36, 38, 54, 43, 32, 21};`

10. Care sunt valorile tabloului v după apelul funcției tranform(v, 8) ?

```
int v[] = {0, 1, 2, 3, 4, 5, 6, 7};

void transform(int v[], int n) {
    int temp[n];

    for (int i = 0; i < n; i++) {
        temp[i] = v[(i+3) % n];
    }

    for (int i = 0; i < n; i++) {
        v[i] = temp[(i+2) % n];
    }
}
```

(a) $\{5,6,7,1,2,3,4,0\}$

(b) $\{5,6,7,4,3,2,1,0\}$

(c) $\{1,2,3,4,5,6,7,0\}$

(d) $\{5,6,7,0,1,2,3,4\}$

11. Care este valoarae returnată de apelul funcției process(mat) ?

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

int process(int mat[4][4]) {
    int m = 0;

    for (int i = 1; i < 3; i++) {
        m += mat[i][0] + mat[i][3];
    }

    for (int i = 0; i < 4; i++) {
```

```
        m += mat[0][i] + mat[3][i];
    }

    return m;
}
```

(a) 67 (b) 63 (c) 50 (d) 57

12. Ordinul de complexitate timp al algoritmului clasic de înmulţire a două matrici $A_{m\times n}$ şi $B_{n\times p}$ este:
(a) $\mathcal{O}(mp)$ (b) $\mathcal{O}(mn^2p)$ (c) $\mathcal{O}(mnp)$ (d) $\mathcal{O}(n^2)$

13. O cofetărie oferă gratuit 3 sortimente de topping pentru înghețată. Dacă sunt disponibile în total 10 tipuri de topping, câte variante are Bogdan de a alege gratuit cele 3 topping-uri?
(a) 100 (b) 110 (c) 150 (d) 120

14. Se consideră toate șirurile de lungime 1, 2 și 3 formate din elemente din șirul 1, 2, 3, 4, 5. Câte dintre aceste șiruri au elementele ordonate strict descrescător și un număr impar de elemente impare.

(a) 10 (b) 13 (c) 25 (d) 85

15. Alex vrea să descopere un cod secret folosit pentru accesarea unui seif digital. El știe că acest cod este format din 4 caractere, fiecare reprezentând o cifră între 0 și 9, din care doar cifrele pare 0, 2, 4, 6, 8 sunt utilizate.
Alex decide să folosească un brute force attack, testând toate combinațiile posibile până când descoperă codul corect.
Care este numărul maxim de încercări eșuate pe care le poate face Alex (dar care respectă condițiile) înainte de a ghici codul corect?
(a) 120 (b) 624 (c) 626 (d) 625

16. Fie funcțiile:

```
void fct_aux_1(int index, int array[])
{
    array[index] = 10 * index + array[index] + 1;
    array[index] = array[index] / 2;
}
void fct_aux_2(int array[])
{
    for (int index = 0; index < 8; ++index)
    {
        array[index] = array[index] * 3 + 5;
        array[index] = array[index] / 2;
    }
}
int cautare_binara(int arr[], int low, int high, int x)
{
    var_globala = (var_globala + 1) * 2;

    if (high >= low) {
        int mid = low + (high - low) / 2;

        if (arr[mid] == x)
            return mid;
```

```
        if (arr[mid] > x)
            return cautare_binara(arr, low, mid - 1, x);

        return cautare_binara(arr, mid + 1, high, x);
    }
    return -1;
}
```

Din programul principal *main*() sunt rulate următoarele instrucțiuni:

```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8};

for (int index = 7; index >= 0; index--)
{
    fct_aux_1(7 - index, array);
}
fct_aux_2(array);
cautare_binara(array, 0, 7, 53);
```

Precizați care este valoarea variabilei globale *var_globala* dupa rularea codului de mai sus, știind ca aceasta a fost inițializată cu valoarea 5.

(a) 12 (b) 26 (c) 54 (d) 110

17. Fie codul de mai jos:

```
int v[] = { 8, 3, 5, 2, 4 };
int n=5;
for (int i = 1; i < n; i++)
{
    int aux = v[i];
    int j = i;
    while (j > 0 && v[j - 1] < aux)
    {
        v[j] = v[j - 1];
        j--;
    }
    v[j] = aux;
}
```

Câte din afirmațiile de mai jos sunt adevărate?

i) Metoda ordonează descrescător un vector folosind sortarea prin inserție
ii) Metoda ordonează descrescător un vector folosind sortarea prin selecție
iii) Metoda ordonează crescător un vector folosind sortarea prin metoda bubblesort
iv) Metoda ordonează crescător un vector folosind sortarea prin metoda numărării
v) După a treia iterație a lui i, vectorul arată astfel: 8, 5, 4, 3, 2
vi) După a patra iterație a lui i, vectorul arată astfel: 8, 5, 3, 2, 4

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

18. În graful orientat complet cu 6 noduri, numărul ciclurilor hamiltoniene distincte (dacă au cel puțin o muchie diferită) este:

(a) 6 (b) 120 (c) 5 (d) 720

19. Într-un graf conex cu 60 de vârfuri și 120 de muchii, care este numărul maxim muchii ce pot fi eliminate astfel încât graful să rămână conex:

(a) 60 (b) 61 (c) 59 (d) 1

20. Fie G un graf neorientat cu m noduri, care este complet. Presupunem că avem un număr de n astfel de grafuri, notate cu $G_1, G_2, ...G_n$. Aceste grafuri sunt conectate într-o structură S de tip lanț de grafuri: $G_1 - G_2 - ... - G_n$. Conectarea a două grafuri adiacente din structura S se face printr-o singură muchie. Presupunem că pentru toate grafurile G_k din structura S, conectarea cu grafurile adiacente se face prin același nod.
Care este diametrul structurii S?

Definiții:
$G = (V, E)$ - graf neorientat cu setul de noduri V și setul de muchii E.
$d(u, v)$ - distanța minimă dintre nodurile u și v, egală cu numărul de muchii al celui mai scurt drum dintre u și v.
$\epsilon(u) = \max_{u \in V} d(u, v)$ - excentricitatea nodului u.
$radius(G) = \min_{u \in V} \epsilon(u)$ - raza grafului G.
$diameter(G) = \max_{u \in V} \epsilon(u)$ - diametrul grafului G.

(a) $2 * (n - 1)$ (b) $n - 1$ (c) $n + 1$ (d) $n * m - 1$

21. Fie următoarea funcție. Câte dintre afirmațiile de mai jos sunt adevărate, dacă inițial $x = 0$, vectorul a are toate cele n elementele nule ($n >= 4$), iar la apelarea funcției y are valoarea 4?

```
int fun(int &x, int y, int a[])
{
    if (y <= 0)
        return x;
    else {
        a[--y] = ++x;
        x = a[y];
        return fun(x, y-1, a);
    }
}
```

i) Valorile nenule din vector după apelarea funcției sunt egale cu y.
ii) Pentru apelul cerut funcția va returna valoarea 4.
iii) Pentru apelul cerut a[1] va avea la final valoarea 2.
iv) Există o eroare pentru că nu se precizează numărul de elemente a vectorului a.
v) a[–y] va genera un mesaj de eroare pentru ca y va ajunge la valori mai mici ca 0.
vi) Pentru apelul cerut funcția se va autoapela (nu se numără apelul initial) de 2 ori.
(a) 1 (b) 2 (c) 3 (d) 4

22. Care este valoarea returnată de apelul funcției process(20250, 0) ?

```
int process(int n, int m) {
    if (n == 0) return m;
    return process(n / 10, m * 10 + n % 10);
}
```

(a) 20250 (b) 52020 (c) 2025 (d) 5202

23. Pe un bulevard sunt cate N becuri (numerotate de la 1 la N) pe ambele trotuare, aşezate echidistant si simetric fata de axul drumului. Pentru economie, unele becuri sunt aprinse (stare 1), iar altele sunt stinse (stare 0). Becurile de pe oricare parte a bulevardului sunt legate între ele astfel încât apăsarea comutatorului unui bec produce modificarea

atât a stării lui, cât şi a becurilor vecine lui. Deci, dacă se atinge comutatorul becului i (2iN-1) atunci se modifică stările becurilor i-1, i și i+1. Dacă se atinge comutatorul becului 1, atunci se modifică stările becurilor 1 și 2, iar dacă se atinge comutatorul becului N, atunci se modifică stările becurilor N-1 și N.

Se doreste ca stările becurilor de pe ambele trotuare sa fie identice. Daca pe trotuarul A starile celor 10 becuri sunt: A=1, 0, 0, 1, 1, 0, 1, 1, 0, 1

Iar pe trotuarul B toate becurile sunt stinse (stare 0). Care este numarul minim de atingeri ale comutatoarelor de pe trotuarul B astfel incât becurile să aibă aceeași stare pe ambele trotuare (starea becurilor de pe trotuarul A nu se modifica)?

(a) 7 (b) 6 (c) 5 (d) Nu există o succesiune de atingeri care să satisfacă cerința

24. Se consideră funcția

```
void foo(int n) {
    if (n >= 1) {
        foo(n - 1);
        foo(n - 3);
        cout << n;
    }
}
```

De câte ori este invocată funcția *foo*(), dacă în *main* se apelează *foo*(6)? (a) 15 (b) 25 (c) 35 (d) 45

25. Ce valoare returnează apelul funcției fun(s, 4) din următoarea secvență de cod?

```
struct data {
    int idx[5];
    int val[5];
};

struct data t = {
    {1, 2, 1, 4, 3},
    {1, 2, 3, 2, 4}
};

int fun(struct data t, int n) {
    if (n < 0) return 1;
    return fun(t, n - 1) + t.val[t.idx[n]];
}
```

(a) 16 (b) 14 (c) 12 (d) 13