Model 6

Universitatea Tehnică "Gheorghe Asachi" din Iași
Facultatea de Automatică și Calculatoare
Admitere – sesiunea iulie 2021
Domeniile: Calculatoare și tehnologia informației
Ingineria sistemelor (Automatică și informatică aplicată)

Subiecte la testul grilă de Informatică

1. Fie următoarele structuri:

```
struct candidat
{
    char nume[20];
    int nota;
};

struct examen
{
    candidat c[20];
    char sesiune[20];
}admitere2021;
```

Știind ca sunt 20 de candidați, care din secvențele de cod de mai jos afișează nota studenților al căror nume începe cu litera „A”?

(a)
```
for (i = 0; i < 20; i++)
    if (admitere2021.c[i].nume[0] == 'A')
        cout << admitere2021.c[i].nota;
```

(b)
```
for (i = 0; i < 20; i++)
    if (admitere2021.nume[0] == 'A')
        cout << admitere2021.c[i].nota;
```

(c)
```
for (i = 0; i < 20; i++)
    if (admitere2021.c[i].nume == 'A')
        cout << admitere2021.c[i].nota;
```

(d)
```
for (i = 0; i < 20; i++)
    if (admitere2021.candidat[i].nume[0] == 'A')
        cout << admitere2021.candidat[i].nota;
```

2. Care este rezultatul execuției următorului program?

```
int function1(int);
int main()
{
        int k=30;
        k = function1(k=function1(k=function1(k)));
        cout << "k = " << k << endl;
        return 0;
}
int function1(int k)
{
        k++;
        return k;
}
```

(a) k = 30; (b) k = 31; (c) k = 32; (d) k = 33.

3. Știind că variabilele x și y sunt de tip întreg, care dintre expresiile de mai jos are valoarea 1 dacă și numai dacă valorile nenule ale variabilelor x și y sunt egale?

(a) (x<=y) && (y<x)
(b) (x%y==0) && (y%x==0) && (x*y>0)
(c) x*x==y*y
(d) (x<=y) || (y<=x)

4. Fie următorul cod:

```
char text[] = "Admitere AC";

for (i = 0; i < strlen(text) - 1; i++)
    text[i + 1] = text[i];

cout << text;
```

Ce se va afișa pe ecran?

(a) AAAAAAAAAAA (b) AAdmitere A (c) AAdmitere AC (d) CAdmitere A

5. Se consideră o matrice M cu 7 linii care conține doar elemente de 0 și X. Matricea se completeaza astfel:

- prima linie conține un singur element X;
- linia i conține de două ori mai multe elemente X decât linia i-1, pentru $i \in \{2, 3, 4, 5, 6, 7\}$
- ultima linie contine un singur element O.

Câte elemente de O sunt în matrice?

(a) 127 (b) 328 (c) 455 (d) 256

6. Se consideră o matrice pătratică M de dimensiune n x n, cu $n \in [3, 100]$ și funcția $f(M, n, i, j)$ care pune caracterul '*' pe anumite pozitii din matrice. Parametrii i și j sunt numere naturale în intervalul $[1, n]$, iar DIV din codul de mai jos reprezintă câtul împărțirii.

funcția f(M,n,i,j)
  dacă i <= n *DIV 2* **atunci**
    dacă j <= *n-1* **atunci**
      $M[i][j] \leftarrow$ '*';
      f($M, n, i, j+1$);
    altfel
      f($M, n, i+1, i+2$)

De câte ori se autoapelează f dacă n=7, i=2 și j=4?

(a) de o infinitate de ori (b) de 5 ori (c) de 4 ori (d) de 7 ori

7. Fie matricea m cu 5 linii și 5 coloane, cu elementul de pe prima linie și prima coloană aflat pe poziția 0, 0 și un vector v cu 5 elemente inițializate cu 0, cu indexul primului element 0:

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

15	14	12	9	5
0	1	2	3	4

Cum va arăta vectorul v după rularea codului următor?

```
for (i = 0; i < 5; i++)
    for (j = i; j < 5; j++)
        v[i] = v[i] + m[i][j];
```

(a) 15 14 12 9 5 (b) 14 12 9 5 0 (c) 1 3 6 10 15 (d) 15 15 15 15 15

8. Ce se va afișa la rularea codului următor:

```
char text[] = "Admitere AC";

for (i = 0; i < strlen(text); i++)
    if (text[i] > 'A' && text[i] < 'Z')
        text[i] = text[i] - 'A' + 'a';

cout << text;
```

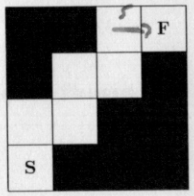
(a) Admitere Ac (b) admitere ac (c) ADMITERE AC (d) aDMITERE ac

9. Fie un robot care trebuie să se deplaseze printr-un labirint, de la punctul de start S la punctul de final F (vezi figura).

Robotul poate efectua următoarele acțiuni: *deplasareInainte*(), *rotireDreapta*() și *rotireStanga*(). La acțiunea de deplasare înainte, robotul se deplasează cu un pătrățel în direcția spre care este îndreptat. Cele două tipuri de rotiri sunt de 90 de grade, iar robotul nu se deplasează.

Fie următorul pseudocod:

funcția f()
- **pentru** $i \leftarrow 1$ **la** 5 **execută**
  - [acțiunea 1]
  - **dacă** *i este impar* **atunci**
    - [acțiunea 2]
  - **altfel**
    - [acțiunea 3]
  - [acțiunea 4]
- [acțiunea 5]



În condițiile în care robotul este inițial orientat în sus, ce trebuie de completat în locurile [acțiunea 1], [acțiunea 2], [acțiunea 3], [acțiunea 4] și [acțiunea 5] astfel încât, în urma execuției pseudocodului de mai sus, robotul să ajungă la finalul labirintului din figură.

(a) deplasareInainte(), rotireDreapta(), rotireStanga(), -, deplasareInainte()
(b) -, rotireDreapta(), rotireStanga(), deplasareInainte(), rotireStanga()
(c) -, deplasareInainte(), rotireDreapta(), rotireStanga(), deplasareInainte()
(d) deplasareInainte(), rotireDreapta(), rotireStanga(), deplasareInainte(), -

da → $\hat{J}_{st}$

$\hat{J}$

10. Care din următoarele probleme referitoare la o mulțime de numere întregi $M = \{x_1, x_2, \ldots, x_n\}, (n > 1000)$, poate fi rezolvată cu un algoritm care are un număr minim de pași?

(a) Sortarea elementelor mulțimii
(b) Generarea tuturor permutărilor mulțimii M
(c) Generarea elementelor produsului cartezian M x M
(d) Determinarea elementului maxim al mulțimii M

11. Într-o sală de examen, scaunele sunt etichetate cu o literă urmată de o cifră. Se consideră că literele sunt $\{a, b, c, d, e\}$ și cifrele sunt $\{0, 1, 2, \ldots, 9\}$. Scaunele trebuie să aibă etichete diferite între ele. Care este numărul minim de scaune care trebuie să fie în sală pentru a folosi toate combinațiile posibile?

(a) 50 (b) 15 (c) 100 (d) 45

12. Care este complexitatea de timp a următorului program C++?

```
int i, j, k;
for(i = 0; i < n; i++) {
    k = n;
    for(j = 0; j < k; j++) {
        k = k / 2;
    }
}
```

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

13. Fie secvența de cod de mai jos. Ce trebuie de completat în locul celor trei puncte (...) pentru ca în matricea c să se obțină combinările de n luate câte k (C_n^k).

```
int c[n+1][n+1];
for (int i = 0; i <= n; i++)
    for (int j = 0; j <= n; j++)
        c[i][j] = 0;
c[0][0] = 1;
for (int i = 1; i <= n; i++) {
    c[i][0] = 1;
    for (int j = 1; j <= i; j++)
        c[i][j] = ...
}
```

(a) c[i - 1][j - 1] * c[i - 1][j] / c[i][j - 1]
(b) c[i - 1][j] + c[i - 1][j - 1]
(c) c[i][j - 1] + c[i - 1][j]
(d) c[i - 1][j - 1] + c[i][j - 1]

D pascal

14. Se considera următorul algoritm cu nr, x și y numere naturale:

citește x, y ;
$nr \leftarrow 0$;
$d \leftarrow 2$;
cât timp $d<=x$ *și* $d<=y$ **execută**
- **dacă** $x \% d = 0$ *și* $y \% d=0$ **atunci**
  - $nr \leftarrow nr + 1$;
  - $x \leftarrow [x/d]$;
  - $y \leftarrow [y/d]$;
- **altfel**
  - $d \leftarrow d + 1$;

scrie nr, x, y

se va afișa dacă se citesc valorile x=720 și y=495 ?

(a) 3, 48, 33
(b) 3, 16, 11
(c) 2, 48, 33
(d) 3 16 33

15. Se consideră mulțimea $A = \{1, 2, \ldots, n\}$ și mulțimea $B = \{1, 2, \ldots, m\}$, $m > n$. O funcție injectivă definită pe A cu valori în B poate fi considerată:

(a) un aranjament de n elemente luate câte m
(b) un aranjament de m elemente luate câte n
(c) o combinare de m elemente luate câte n
(d) o combinare de n elemente luate câte m

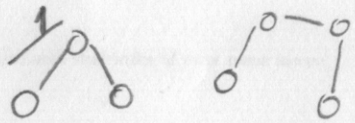
$f: \{1, 2, 3\} \to \{1, 2, 3, 4, 5\}$?

$\frac{5 \cdot 4}{2} = 10 - 4 = 6$

16. Care este ordinul de complexitate timp pentru algoritmul următor?

funcția *sum*
Date de intrare: n
Date: $s = 0$
dacă $n == 0$ **atunci**
  returnează 1
pentru i **de la** 0 **la** $n - 1$ **execută**
  $s = s + sum(n - 1)$
returnează s

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



$\frac{4 \cdot 3}{2} = 6 - 3 = 3$

17. Numărul de *subgrafuri* ale unui graf cu *23* noduri este:

(a) $2^{22} - 1$ (b) 2^{24} (c) 2^{23} (d) $2^{23} - 1$

18. 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 rămână același?

(a) 19 (b) 60 (c) 12 (d) 10

19. Dacă se elimină o muchie dintr-un arbore complet cu cel puțin două niveluri se obține:

(a) un graf neconex
(b) cel putin două noduri izolate
(c) tot un arbore
(d) cel putin un ciclu

20. Fie G un graf neorientat cu n noduri. Fiecare nod are un atribuit un identificator $1, 2, \ldots, n$. 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 par, care este numărul maxim de muchii care pot fi tăiate astfel încât graful G să rămână format dintr-o singură componentă conexă?

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

21. Un arbore binar complet este un arbore binar în care toate nivelurile, cu excepția ultimului sunt ocupate în întregime. Considerând faptul că toate nodurile de pe ultimul nivel sunt grupate în partea stânga a acestuia, un arbore binar complet cu n noduri frunză are:

(a) $n + 1$ noduri. (b) $\log n$ noduri. (c) $2n - 1$ noduri. (d) 2^n noduri.

22. O trupa de genești încearcă să mineze o suprafată de formă patrată, având la dispozitie n mine. Acești trebuie să plaseze minele astfel încât, dacă explodeaza o mină să nu explodeze și celelalte din apropiere. Pentru aceasta ei împart suprafața în nXn pătrate astfel încât într-un pătrat poate fi pusă în siguranță o singură mină. Acești apeleaza la un program de tip backtracking care să genereze toate combinațiile posibile pentru a acoperi toate rândurile și coloanele formei create cu măcar o mină. Care din secvențele de mai jos verifică corect o astfel de plasare a minelor?

Problema reginelor

(a)
```
int pozitioneaza (int n)
{
  for(i=1;i<n;i++)
    if (v[i]==v[n] || abs(v[i]-v[n])==abs(i-n)) return 0;
  return 1;
}
```

(b)
```
int pozitioneaza (int n)
{
  for(i=1;i<k;k++)
    if (v[i]==v[n]) return 0;
  return 1;
}
```

(c)
```
int pozitioneaza (int n)
{
  for(i=1;i<n;n++)
    if (v[i]==v[n] || abs(v[i]-v[n])==abs(i-n)) return 0;
  return 1;
}
```

(d)
```
int pozitioneaza (int n)
{
  for(i=1;i<n;n++)
    if (v[i]==v[n] || abs(i-n)==0) return 0;
  return 1;
}
```

23. Ce calcul realizează funcția următoare?

```
int fun(int x, int y)
{
    if (y == 0) return 0;
    return (x + fun(x, y-1));
}
```

(a) x+y (b) x*y (c) x + x*y (d) x^y

24. Fie subprogramul:

```
void functie (int n, int b, int &m)
{

    if (n!=0){
        if (n%b!=0) m++;
        functie(n/b,b,m);
        cout<<n%b;}
}
```

Pentru un apel de forma functie(100,5,m) unde m=0, funcția va avea ca efect:
(a) transformă numărul în baza b și contorizează în m numărul de cifre ale lui n transformat în baza b.
(b) numără câte cifre diferite de 0 în baza b are numărului n.
(c) transformă numărul în baza m.
(d) numără de cate ori apare o cifră b în numărul n.

25. Pentru verificarea cunoștințelor de limba germană, o grupă de 8 elevi au susținut un examen alcătuit din două teste de verificare. La fiecare dintre cele două teste unii dintre elevi au fost absenti. Codul de identificare pentru un elev și nota obținută la un test sunt stocate într-o structura de date. Notele obținute de elevii care au fost prezenți la fiecare dintre cele două teste sunt stocate în tablourile test1 si test2. Pentru a trece examenul, un elev trebuie să obțină o notă mai mare sau egală cu 5 la fiecare dintre cele două teste. Daca un elev nu este prezent la un test atunci nota pentru acel test se consideră 0.

```
struct elev {
    int cod;
    int nota;
};
struct elev test1[5], test2[7];
int fun(void) {
    int ret = 0;
    int i, j;
    for (i = 0; i < 5; i++) {
        for (j = 0; j < 7; j++) {
            if(...) {
                ret = ret + 1;
            }
        }
    }
    return ret;
}
```

Care este condiția care trebuie verificată în locul marcat cu ... pentru ca funcția să returneze numărul de elevi care au trecut examenul?

(a) test1[i].cod == test2[j].cod && test1[i].nota > 4 && test2[j].nota > 4
(b) test1[i] == test2[j] && test1[i].nota >= 5 && test2[j].nota >= 5
(c) test1[i].cod == test2[j].cod && test1[i].nota > 5 && test2[j].nota > 5
(d) i == j && test1[i].nota > 4 && test2[j].nota > 4