JAȘI IULIE 2022

23/25

A

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

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

1. Într-o secvență de cod corectă există instrucțiunea

```
if (!((x >= 3) && !((x >= 3) && (y < 5))))
```

Expresia de test este echivalentă cu:

(a) `(x >= 3) && (y >= 5)`

(b) `(x < 3) || (y < 5)`

(c) `(x < 3) && (y >= 5)`

(d) `(x >= 3) || (y < 5)`

2. Fie A un arbore cu o singură rădăcină și n noduri ($n > 3$), în care fiecare nod care nu este frunză are exact doi fii (fiu stâng și fiu drept) și toate frunzele sunt pe același nivel. Dacă la toate nodurile interne se taie muchia către fiul drept, ce procent din numărul total de arbori care se formează reprezintă arbori cu un singur nod?

(a) 100% (b) 12.5% (c) 50% (d) 25%

3. Se consideră definițiile pentru matricea și funcția de mai jos:

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

void f (int M[4][4], int n, int k) {
    int i, t;
    for(i = k; i<n-k; i++) {
        t = M[i][i];
        M[i][i] = M[i][n-i-1];
        M[i][n-i-1] = t;
    }
}
```

Care este suma elementelor de pe diagonala principală a matricei A, după următorul apel al funcției f: f(A, 4, 1)?

(a) 18 (b) 17 (c) 19 (d) 16

4. Care este numărul maxim de frunze ale unui arbore cu o rădăcină și un număr total de 67 de noduri, în care fiecare nod are cel mult doi fii?

(a) 38 (b) 36 (c) 32 (d) 34

5. Se consideră secvența de prelucrări de mai jos în care i = 1, k = 0 și n este un număr întreg pozitiv oricât de mare:

Știind că operatorul [x] are semnificația părții întregi a numarului x, care este valoarea variabilei k în urma execuției secvenței următoare?

cât timp $i <= n$ **execută**
  $i \leftarrow 2 * i$
  $k \leftarrow k + 1$

(a) $2 * n$ (b) $[log_2(n)] + 1$ (c) n/2 (d) $[log_2(n)]$

6. Care este ordinul de complexitate timp pentru următoarea secvență?

```
for (i = 0; i < n * n; i++) {
    for (j = 0; j < n; j++)
        s = s + i * j;
    for (k = 0; k < n; k = k + 2)
        s = s + k;
}
```

(a) $n^3 \log n$ (b) n^3 (c) n^4 ~~(d) $n^2 \log n$~~

7. În limbajul C/C++, care dintre următoarele expresii este adevărată pentru oricare două numere întregi x și y diferite de zero, atunci când se verifică dacă rezultatul împărțirii este un număr întreg?

(a) x%y==(int)(x/y) (b) (double) x/y==x%y (c) x/y==x%y (d) x/y ==(double) x/y

8. Fie următoarea funcție:

```
int f(int a) {
    int x, y;
    x = a % 100;
    y = x % 10;
    x = x / 10;
    if (x % 2 == y % 2)
        return 1;
    return 0;
}
```

Știind că parametrul a are cel puțin patru cifre, funcția f:

(a) verifică dacă ultimele două cifre ale numărului transmis ca parametru au aceeași valoare.
(b) verifică dacă numărul transmis ca parametru este par.
(c) verifică dacă toate cifrele numărului transmis ca parametru au aceeași paritate.
(d) verifică dacă ultimele două cifre ale numărului transmis ca parametru au aceeași paritate.

9. Se consideră funcția recursivă de mai jos:

funcția $f(n)$
- **dacă** $n <= 3$ **atunci**
  - **returnează** 1;
- **altfel**
  - **returnează** $f(n-1) + f(n-2) + f(n-3)$

Dacă primul apel este f(10), de câte ori se calculează f(4)?

(a) 24 (b) 21 (c) 35 (d) 15

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

```
int v[] = {0, 1};
int w[] = {0, 2};
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 2 0
1 0 1
1 2 0
```

(b)
```
0 0 0
0 2 2
1 0 1
1 2 3
```

(c)
```
0 0 0
0 2 1
1 0 0
1 2 1
```

(d)
```
0 0 1
0 2 0
1 0 0
1 2 0
```

11. Se consideră două matrice de numere întregi: A are m linii și n coloane, B are n linii și p coloane. Numărul de înmulțiri elementare între elementele matricelor A și B, realizate de algoritmul clasic de înmulțire a celor două matrice (bazat pe definiția produsului a două matrice) este:

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

12. Un designer vrea să vopsească cele 7 elemente ale unui loc de joacă folosind un număr minim de culori, dar își dorește ca două elemente vecine să aibă culori diferite. Pentru aceasta el utilizează un graf neorientat care modelează componența locului de joacă, graf ce are următoarele muchii: [1,4], [1,5], [1,6], [1,7], [2,3], [2,5], [2,6], [3,4], [5,6]. Care este numărul minim de culori ce poate fi utilizat?

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

13. Se consideră definițiile următoarelor variabile:

```
char s[] = "102030405060";
char d[30];
```

Care este valoarea afișată de următoarea instrucțiune?

```
cout << strcat(strcpy(d, s + strlen(s) - 4), s + 8);
```

(a) 50605060 (b) 60506050 (c) 60505060 (d) 50506060

14. Fie următoarele structuri:

```
struct Punct {
    int x,y;
};

struct Triunghi {
    Punct p[3];
};
```

Știind că variabila t este o variabilă de tipul *Triunghi* declarată cu *Triunghi t*; și că punctele triunghiului au fost inițializate anterior, care dintre următoarele instrucțiuni verifică dacă primul punct al triunghiului se află în primul cadran (ambele coordonate pozitive)?

(a) `if(t[0].p.x > 0 && t[0].p[1].y > 0)`

(b) `if(t.p[0] > 0)`

(c) `if(p[0].x > 0 && p[1].y > 0)`

(d) `if(t.p[0].x > 0 && t.p[0].y > 0)`

15. Un beduin trebuie să parcurgă distanța dintre două oaze din deșert, având la dispoziție o cămilă și n recipiente, fiecare recipient având 20 de litri de apă. Cămila este extrem de îndărătnică și nu acceptă să care mai mult de un singur recipient pe lângă beduin și numai după ce bea 20 de litri de apă, apoi poate merge exact 105 km. La ce distanță maximă trebuie să fie oaza destinație dacă beduinul are 5 recipiente? Indicație: Beduinul folosește un recipient pentru a potoli setea cămilei cu scopul de a căra restul recipientelor la o distanță pe care o poate parcurge cămila dus-întors de câte ori este nevoie.

(a) 176 (b) 325.5 (c) 157.5 (d) 281

16. Fie secvența de cod de mai jos. Dacă în funcția *main*, v este un vector cu n elemente întregi pozitive, x are o valoare pozitivă, iar i are valoarea 0, ce returnează $f(v, n, x, i)$ apelată din *main*?

```
int f(int v[], int n, int x, int i) {
    if (i == n)
        return -1;
    if (v[i] == x)
        return x;
    return f(v, n, x, i + 1);
}
```

(a) Apelul returnează întotdeauna valoarea lui x.
(b) Valoarea lui x, dacă x se găsește în vectorul v, iar v este un vector cu elemente nesortate.
(c) Programul este blocat într-o buclă infinită.
(d) Valoarea lui x, dacă x se găsește în vectorul v, iar v este neapărat un vector sortat crescător.

17. Fie elementele matricei m cu 5 linii și 5 coloane, cu elementul de pe prima linie și prima coloană aflat pe poziția 0, 0:

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

Având în vedere că n are valoarea 5 și variabila s a fost inițializată cu 0, ce valoare se va afișa după execuția codului următor?

```
for (int i = n/2 + 1; i < n; i++)
    for (int j = n - i/2 - 1; j < n - i/2 + 1; j++)
        s = s + m[i][j];
cout << s;
```

(a) 42 (b) 82 (c) 40 (d) 126

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

```
char s[20] = "Admitere AC 2022";
char v[20] = "vara";
int w[20], i, j, z;
for(i=0; s[i] != '\0'; i += 1)
    ;
z = i;
cout << i << endl;
for(i = 0, j = 0; s[i]; i++) {
    if(s[i] >= '0' && s[i] <= '9') {
        v[j] = s[i];
        w[j] = v[j] - '0';
        s[i] = 0;
        ++j;
    }
}
v[8] = 'H';
cout<<s<<endl;
cout<<v<<endl;
for(i=0; w[i]; i++) {
    cout << w[i] << "\n";
}
```

(a) 30
Admitere AC 2022
2022
2022

(b) 16
Admitere AC
2022 H
2

(c) 16
Admitere AC
2022
2022

(d) 16
Admitere AC
2022
2

19. Fie codul de mai jos:

```
int g(int y);

int f(int x) {
    if (x==0) return 5;
    else return g(x/2)+1;
}

int g(int y) {
    if (y<0) return 0;
    else return f(y-1)+2;
}

int main() {
   cout << f(5);
   return 0;
}
```

recursivitate indirectă

De câte ori se apelează funcția g?

(a) programul este incorect, deoarece nu se pot apela funcțiile în acest mod
(b) de o infinitate de ori
(c) niciodată, deoarece o funcție nu poate fi declarată de două ori
(d) de două ori

20. O rețetă de chec necesită următoarele ingrediente cu gramajele aferente: 200g zahăr, 150g unt, 120g ouă, 300g banane, 250g făină, 5g praf de copt și 20g nuci.

Ingredientele pot fi combinate separat între ele sau pot fi integrate în compoziții. O compoziție se poate obține prin combinarea a două elemente. Un element reprezintă un ingredient sau o altă compoziție.

Timpul de combinare a două elemente se calculează ca fiind suma gramajelor elementelor care se combină, transformată în secunde. De exemplu, pentru a combina 200g zahăr cu 120g ouă sunt necesare 320 secunde, iar pentru a combina această compoziție cu 20g nuci vor fi necesare încă 340 secunde.

La început, se combină untul cu zahărul, și, separat, făina și praful de copt. Nucile trebuie adăugate/combinate la final. Când se combină două elemente trebuie folosită întreaga cantitate a celor două elemente. Care este timpul minim de preparare a rețetei?

(a) 3865 secunde (b) 3700 secunde (c) 1045 secunde (d) 2655 secunde

21. Fie următoarea declarație asociată unei fracții:

```
struct FRACTIE {
    int numarator;
    int numitor;
};
```

și prototipul funcției care calculează suma a două fracții:

```
struct FRACTIE suma(struct FRACTIE, struct FRACTIE);
```

Funcția *suma* adună fracțiile notate cu $p1$ și $p2$. Care din variantele de mai jos este corectă?

(a)
```
rez = suma(p1, p2);
cout << rez.numarator << "/" << rez.numitor << endl;
```

(b)
```
rez = suma(p1, p2);
cout << rez << endl;
```

(c)
```
rez = suma(FRACTIE p1, FRACTIE p2);
cout << rez.numarator << "/" << rez.numitor << endl;
```

(d)
```
rez = suma(FRACTIE p1, FRACTIE p2);
cout << rez << endl;
```

22. Fie definiția următoarei funcții:

```
int f(int a[], int n) {
    int k = 0, pk = 1, m = 0;
    while(k>=0 && k<n && pk!=k && m<n ) {
        pk = k;
        k = a[k];
        m++;
    }
    return m;
}
```

Se consideră apelul f(a, 10). Pentru care dintre următoarele tablouri funcția f returnează cea mai mare valoare?

(a) int a[] = {4, 6, 7, 8, 5, 1, 6, 8, 0, 9};
(b) int a[] = {5, 1, 7, 3, 9, 1, 2, 4, 0, 8};
(c) int a[] = {4, 6, 7, 6, 5, 1, 2, 7, 0, 3};
(d) int a[] = {5, 2, 7, 8, 4, 5, 6, 8, 0, 3};

23. Ce efect are codul următor asupra matricei a?

```
int a[3][3] = {{-1, 0, 1}, {2, 3, 1}, {0, -1, -2}};
int f1, f2, i, j, ok, n = 3;
int k = n-1;
do {
    ok = 1;
    for(i=0; i < k; i++) {
        if(i == 0)
            f1 = f(a[i], n);
        else
            f1 = f2;
        f2 = f(a[i+1], n);
        if(f1 > f2)
            for(j=0; j < n; j++) {
                int v;
                v = a[i][j];
                a[i][j] = a[i+1][j];
                a[i+1][j] = v;
                ok = 0;
            }
    }
    k = k - 1;
} while(ok == 0);
```

Funcția f apelată în secvența de cod dată este de forma:

```
int f(int a[], int n) {
    int i, F;
    F = a[0];
    for(i=1; i<n; i++)
        if(a[i] > F)
            F = a[i];
    return F;
}
```

(a) Se face ordonarea matricei a în ordinea crescătoare a elementului maxim de pe fiecare coloană, folosind metoda bulelor;
(b) Se face ordonarea matricei a în ordinea crescătoare a elementului maxim de pe fiecare linie, folosind metoda selecției;
(c) Se face ordonarea matricei a în ordinea crescătoare a elementului maxim de pe fiecare linie, folosind metoda bulelor;
(d) Codul va genera erori la compilare pentru că nu putem transmite $a[i]$ ca parametru pentru funcția f definită în enunțul problemei.

24. Fie G un graf neorientat cu n noduri. Dacă G este complet, câte muchii pot fi tăiate astfel încât structura rezultată să fie formată din doi arbori distincți?

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

25. Fie funcțiile de mai jos:

```
int fct1 (int a, int b) {
  while (a!=b)
    if (a<b)
      return fct1(a,b-a);
    else
      return fct1(a-b,b);
  return a;
}
```

```
int fct2 (int a, int b) {
  while (a!=b)
    if (a<b)
      b=b-a;
    else
      a=a-b;
  return a;
}
```

```
int fct3 (int a, int b) {
  while (a!=b)
    if (a<b)
      return b=b-a;
    else
      return a=a-b;
  return a;
}
```

```
int fct4 (int a, int b) {
  while (a!=b)
    if (a<b)
      b=fct4(a,b-a);
    else
      a=fct4(a-b,b);
  return a;
}
```

Care dintre funcțiile de mai sus au același rezultat indferent de valorile lui a și b, a>0 și b>0?

(a) fct1, fct2 și fct4 (b) fct2, fct3 și fct4 (c) fct1, fct2 și fct3 (d) fct1, fct3 și fct4

a=5 b=10

f1(5,10) → f(5,5) → 5

f2(5,10) → b=5 a=5 → 5

f3(5,10) →