

## Subiecte la testul grilă de Informatică

1. Se consideră următoarea declarație în C: `char nume[10]`; Cât spațiu de memorie se ocupă dacă de la tastatură se citește valoarea „Ana” (ghilimele se exclud) dacă presupunem că un caracter este reprezentat pe un octet?  
(a) 9 octeți. (b) 3 octeți. (c) 10 octeți. (d) 4 octeți.

2. Fie funcția:

```
int f1(int a, int b)
{
    int r;
    r = a % b;
    while(r)
    {
        a = b;
        b = r;
        r = a % b;
    }
    return b;
}
```

Ce se afișează la execuția următorului program:

```
int main()
{
    int a = 259, b = 70;
    int c;
    c = f1(a, b);
    cout << a << " " << b << " " << c << endl;
    return 0;
}
```

- (a) 259 70 7 (b) 21 7 7 (c) 21 0 0 (d) 7 0 7

3. Fie `s` un șir ce poate memora maxim 30 de caractere. Care este valoarea următoarei expresii dacă inițial șirul `s` are valoarea: "admitere2022" ?

```
strlen(strcat(s, strchr(s, 'e')))
```

- (a) 19 (b) 13 (c) 8 (d) 12

4. De câte ori se execută corpul următoarei bucle de program? Inițial variabila `x` are valoarea 2021.

```
do {
    x /= 100;
} while (x%10);
```

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

5. Care din cele doua bucle continute de functia main calculeaza corect lungimea sirului s

```
main()
{
    int j;
    char s[] = "Universitatea Tehnica Iasi";

    for(j=0; s[j]; ++j);
        cout << j << endl;

    j=0;
    while(s[j++]);
        cout << j << endl;
}
```

- (a) niciuna (b) bucla for (c) amandoua (d) bucla while

6. Se consideră următoarea funcție:

```
int f(int x, int y) {
    int ret = 0;
    while (x-- && y--) {
        ret += x + y;
    }
    return ret;
}
```

Care este valoarea returnată de următorul apel al funcției f(675,3) ?

- (a) 1350 (b) 2022 (c) 2028 (d) 2700

7. Dan și Mihai au de rezolvat următoarea problema: se dă un vector ordonat de n elemente. Să se elimine din acesta, fără a crea alt vector, valorile duplicate. Ei au scris următorul cod, dar au senzația că în locul comentat ar trebui să mai adauge ceva pentru ca programul să funcționeze.

```
int a[]={1,1,1,1,2,3,4,4,4,4,5}, lung=11,i,j;

for(i=0;i<=lung;i++)
{
    if(a[i]==a[i+1])
    {
        for(j=i+1;j<=lung;j++)
            a[j]=a[j+1];
        //...
        lung--;
    }
}
```

Ce ar trebui adaugat?

- (a) decrementarea variabilei j (b) decrementarea variabilei i (c) nimic, programul e corect (d) incrementarea variabilei lung, pentru că nu trebuie decrementată la rândul imediat următor

8. Pentru un număr natural  $n$ , se definește funcția  $\phi(n)$  a cărei valoare este egală cu numărul numerelor naturale mai mici decât  $n$  și prime cu el. Prin definiție  $\phi(1)=1$ , iar  $\phi(2)=1$ ,  $\phi(3)=2$  (numerele prime cu 3 sunt 1 și 2),  $\phi(12)=4$  (numerele prime cu 12 și mai mici decât 12 sunt 1, 5, 7, 11).

Care dintre funcțiile următoare calculează și memorează într-un vector valorile funcției  $\phi$  pentru un șir de  $m$  numere naturale și care este secvența de cod care realizează apelul corect al acestei funcții?

Se cunoaște că șirul de numere naturale este memorat în tabloul  $x$ , valorile funcției  $\phi$  sunt memorate în vectorul  $fi$ , funcția *prim* returnează valoarea 1 dacă argumentul funcției este un număr prim și 0 în caz contrar, iar funcția *cmmdc* returnează cel mai mare divizor comun al celor două numere transmise ca argumente.

(a) Funcția este:

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

iar apelul corect al funcției *functie* este:

```
functie(x, fi, m);
```

(b) Funcția este:

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

```

    }
    fi[i] = contor;
}
}

```

iar apelul corect al funcției *functie* este:

```
functie(unsigned int x[], unsigned int fi[], unsigned int m);
```

(c) Funcția este:

```

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

```

iar apelul corect al funcției *functie* este:

```
functie(x, fi, m);
```

(d) Funcția este:

```

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

```

iar apelul corect al funcției *functie* este:

```
fi = functie(x, fi, m);
```

9. Fie un tablou de numere întregi inițializat prin următoarea definiție:

```
int x[] = {1, 2, 3, 1, 3, 1, 1, 3};
```

Câte elemente ale tabloului x își cresc valoarea după apelul funcției `trans(x, 8)`?

```
void trans(int x[], int len) {
    int i, j;
    for (i = 1; i < len - 1; i++) {
        x[i] = (x[i - 1] + x[i + 1]) / 2;
    }
}
```

(a) 0 (b) 2 (c) 3 (d) 1

10. Dan are de efectuat o problemă cu numere întregi. Pentru aceasta el scrie următorul cod:

```
int functie (int n, int m)
{
    while (n!=m)
        if (n>m)
            n=n-m;
        else
            m=m-n;
    return n;
}

int calcul(int n, int m, int z);

int main()
{
    int n=10, m=15;
    int z=functie(n,m);
    int q=calcul(n,m,z);
    cout<<q;
    return 0;
}
```

Știind că rezultatul a fost 90, să se indice ce ar putea reprezenta implementarea funcției `calcul`:

- (a) Ar determina cel mai mare multiplu comun de 2 cifre al numerelor m și n.
- (b) Ar determina un divizor al numerelor n și m.
- (c) Ar determina cel mai mic multiplu comun al numerelor.
- (d) Ar determina produsul divizorilor comuni ai celor doua numere.

11. Care este numărul de interschimbări de elemente în urma execuției algoritmului Bubble Sort pentru vectorul: 15, 7, 6, 11, 3, 18, 9, 23, 2.

(a) 20 (b) 23 (c) 22 (d) 19

12. Secvența de numere întregi  $S = (7, 1, 2, 5, 8, 11, 12, 3, 0, 9, 6)$  urmează a fi sortată crescător prin metoda bulelor. Care este rezultatul obținut după prima parcurgere a secvenței?
- (a)  $S = (1, 2, 5, 7, 8, 11, 3, 0, 9, 6, 12)$     (b)  $S = (0, 1, 2, 3, 5, 6, 7, 8, 9, 11, 12)$     (c)  $S = (7, 1, 2, 5, 8, 11, 6, 3, 0, 9, 12)$   
 (d)  $S = (1, 7, 2, 5, 8, 11, 12, 3, 0, 9, 6)$

13. Pentru a scrie valoarea 9 ca sumă de numere prime se folosește metoda backtracking și se generează, în această ordine, sumele distincte:  $2+2+2+3$ ,  $2+2+5$ ,  $2+7$ ,  $3+3+3$ . Folosind exact aceeași metodă, se scrie valoarea 10 ca sumă de numere prime. Care sunt ultimele trei soluții, în ordinea generării lor?
- (a)  $2+2+2+2+2$ ,  $2+2+3+3$ ,  $2+3+5$   
 (b)  $2+2+3+3$ ,  $2+3+5$ ,  $3+7$   
 (c)  $2+2+2+2+2$ ,  $3+7$ ,  $5+5$   
 (d)  $2+3+5$ ,  $3+7$ ,  $5+5$

14. Dintr-o echipă de 5 persoane: Mihai, Oana, Dan, Ioan, Alina șeful trebuie să aleagă câteva persoane pentru a-i trimite într-o delegație. Delegația trebuie să aibă cel puțin doi membri, dar Mihai și Alina nu pot face parte amândoi din echipă. Astfel, șeful încearcă să creeze mai multe variante. Care este numărul total de variante posibile?
- (a) 6    (b) 19    (c) 14    (d) 18

15. Se consideră funcția  $Calcul(n)$  unde  $n$  este un număr natural,  $n \in [1, 10000]$ :

```

funcția  $Calcul(n)$ 
┌
   $x \leftarrow 0;$ 
   $z \leftarrow 1;$ 
  cât timp  $z \leq n$  execută
  ┌
     $x \leftarrow x + 1;$ 
     $z \leftarrow z + 2 * x;$ 
     $z \leftarrow z + 1;$ 
  └
└ returnează  $x;$ 

```

Care afirmație de mai jos este falsă?

- (a) Funcția calculează și returnează numărul pătratelor perfecte strict pozitive și strict mai mici decât  $n$   
 (b) Funcția calculează și returnează partea întregă a radicalului numărului  $n$   
 (c) Dacă  $n = 1$ , funcția returnează 1  
 (d) Dacă  $n \in [125, 140]$ , atunci  $Calcul(n)$  returnează 11
16. Să se determine câte numere naturale prime reprezentate în baza 2 pe cel mult 5 cifre conțin exact 3 cifre de 1.  
 (a) 10    (b) 8    (c) 3    (d) 4
17. Pentru a testa algoritmul de sortare prin selecție, în cazul cel mai favorabil, un student folosește un tablou de intrare cu 100 de elemente. Câte comparații de elemente va efectua algoritmul?  
 (a) 5000    (b) 4999    (c) 4950    (d) 100
18. Care este complexitatea de timp a următorului program C++?

```

int i, j, k;
for (i = 1; i <= n; i++) {
  k = i;
  for (j = 1; j <= k; j++) {
    k = k / 2;
  }
}

```

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

19. Dintr-un fișier se citesc  $n$  numere ( $n < 100000$ ) naturale de maxim trei cifre. Asupra acestora se aplică următoarele prelucrări: dacă numărul este impar acesta este eliminat din șir. Dacă numărul este par, acesta se împarte la 2. Procedura se repetă până când nu mai există nici un număr în șir. Știind că s-a citit șirul: 12 32 18 44 11 22 15, care din afirmațiile de mai jos este adevărată.

- (a) nu se pot elimina toate elementele din șir (b) numărul de parcurgeri a elementelor din fișier este foarte mare  
(c) se poate parcurge fișierul o singură dată (d) sunt 5 etape până la eliminarea tuturor elementelor

20. Dată fiind o regiune pătratică de  $4 \times 4$  și punctele A, B, C, D, E de coordonate (1,1), (2,2), (3,1), (3,2) respectiv (1,2), se dorește partiționarea ei în 5 zone, după următorul algoritm:

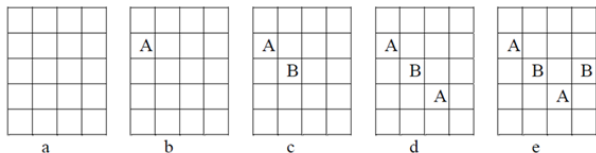
- În fiecare zonă se regăsește un singur punct dintre cele indicate.

- Fiecare zonă este mărginită de mediatoarea dusă pe dreapta dintre punctul zonei respective și celelalte puncte.

Zona cea mai mică obținută este cea care conține punctul?

- (a) A (b) B (c) C (d) D

21. Se consideră o matrice dreptunghiulară împărțită în  $n \times m$  căsuțe ( $n$  – numărul liniilor,  $m$  – numărul coloanelor,  $n, m$  – numere naturale în intervalul  $[2, 100]$ ). Doi jucători notați cu A și B hașurează pe rând câte o căsuță nehașurată vecină pe o diagonală cu cea hașurată la pasul anterior. Jucătorul care nu mai poate hașura respectând această regulă pierde. Jucătorul A începe jocul. O posibilă desfășurare a jocului este prezentată în secvența de imagini de mai jos.



Determinați condiția în care jucătorul A are strategie sigură pentru a câștiga jocul, oricare ar fi mutările jucătorului B. Care poate fi prima căsuță hașurată de jucătorul A pentru a câștiga jocul?

- (a) condiția: numărul  $m$  este impar; prima hașurare a jucătorului A: o casuța aflată pe prima linie de sus a matricei (linia 1) și pe o coloană de indice impar;  
(b) condiția: numărul  $n$  este impar; prima hașurare a jucătorului A: casuța de pe o linie de indice par și pe prima coloană din stânga tablei (coloana 1);  
(c) condiția: ambele numere  $n$  și  $m$  sunt pare; prima hașurare a jucătorului A: casuța din colțul stânga sus (de pe linia 1, coloana 1);  
(d) condiția: ambele numerele  $n$  și  $m$  sunt impare; prima hașurare a jucătorului A: casuța de pe linia 2, coloana 2.

22. În anul 2022, la examenul de admitere la Facultatea de Automatică și Calculatoare participă 721 candidați. Numărul maxim de candidați născuți în aceeași lună este cel puțin  $x$ . Care este valoarea lui  $x$ ?

- (a) 61 (b) 60 (c) 721 (d) 12

23. Fie  $G$  un graf neorientat format din 3 componente conexe cu 3, 4 și, respectiv, 5 noduri. Dacă cele 3 componente conexe reprezintă subgrafuri complete, care este diferența dintre

- numărul maxim de muchii care pot fi adăugate în graf astfel încât numărul de componente conexe să fie 1 și
- numărul minim de muchii care pot fi adăugate în graf astfel încât numărul de componente conexe să fie 1?

- (a) 19 (b) 45 (c) 47 (d) 57

24. Fie  $G$  un graf neorientat format dintr-un număr minim de noduri și dintr-un număr minim de muchii astfel încât să conțină simultan:

- cel puțin un nod cu un vecin
- cel puțin un nod cu doi vecini
- cel puțin un nod cu trei vecini
- cel puțin un nod cu patru vecini
- cel puțin un nod cu cinci vecini

Câte muchii trebuie adăugate la graful  $G$  astfel încât acesta să devină complet?

(a) 6   (b) 10   (c) 15   (d) 18

25. Fie  $G = (V, A)$  un graf complet cu  $n$  vârfuri. Care este numărul maxim de muchii care pot fi eliminate pentru a rămâne graf cu proprietatea că există un ciclu elementar care conține toate vârfurile?

(a)  $\frac{n(n-3)}{2}$    (b)  $n$    (c)  $\frac{n(n-1)}{4}$    (d)  $\frac{n(n-2)}{2}$