

SUBIECTUL al II-lea

(30 de puncte)

Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Se consideră un graf neorientat cu 6 noduri și 9 muchii. Numărul de muchii ce trebuie adăugate, pentru ca graful obținut să fie complet, este: **(4p.)**

a. 5 b. 6 c. 12 d. 15

2. Se consideră secvența de instrucțiuni alăturată, în care variabilele *i* și *j* sunt de tip întreg, variabila *a* este de tip *char*, iar variabila *s* poate memora un șir de cel mult 20 de caractere. Șirul afișat în urma executării secvenței este: **(4p.)**

```
strcpy(s, "bacalaureat");
j=strlen(s);
for(i=0; i<3; i++)
    if(s[i]!=s[j-i-1])
    { a=s[i];
      s[i]=s[j-i-1];
      s[j-i-1]=a;
    }
```

```
cout<<s; | printf("%s", s);
```

a. bacalaureab b. tacalaureab c. tacrlauaeab d. taealaurcab

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

3. Se consideră un arbore cu rădăcină având 8 noduri, numerotate de la 1 la 8, și muchiile [1, 2], [1, 3], [2, 4], [3, 5], [3, 6], [6, 7], [6, 8]. Considerând că rădăcina este nodul numerotat cu 1 și că ea este situată pe nivelul 0 al arborelui, scrieți toate nodurile situate pe nivelul 2 al arborelui dat. **(6p.)**

4. Variabilele *e1* și *e2*, declarate alăturat, memorează numele și data nașterii pentru câte un elev. Știind că cei doi elevi sunt născuți în ani diferiți, scrieți secvența de instrucțiuni care afișează pe ecran numele elevului de vârstă mai mare. **(6p.)**

```
struct data_n {
    int an, luna, zi;
};
struct elev{
    char nume[20];
    data_n d;
} e1, e2;
```

5. Scrieți un program C/C++ care citește de la tastatură un număr natural, *n* ($2 \leq n \leq 20$), și construiește în memorie tabloul bidimensional cu *n* linii și *n* coloane, având proprietățile:
- toate elementele situate pe diagonala principală sunt nule;
 - fiecare linie conține, începând cu diagonala principală, de la dreapta la stânga, un șir strict crescător de numere consecutive, iar începând cu diagonala principală, de la stânga la dreapta, tot un șir strict crescător de numere consecutive.

Programul afișează pe ecran tabloul construit, fiecare linie a tabloului pe câte o linie a ecranului, cu elementele aflate pe aceeași linie separate prin câte un spațiu.

Exemplu: dacă *n*=5 se afișează pe ecran tabloul alăturat.

(10p.)

0	1	2	3	4
1	0	1	2	3
2	1	0	1	2
3	2	1	0	1
4	3	2	1	0

SUBIECTUL al III-lea

(30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Utilizând metoda backtracking se generează toate numerele pare cu câte trei cifre, cifre care aparțin mulțimii {7, 8, 1, 6, 2, 3}. Primele 4 soluții generate sunt, în această ordine: 778, 776, 772, 788. Cea de a 8-a soluție generată este: **(4p.)**

a. 712 b. 716 c. 718 d. 782

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Se consideră subprogramul **f**, definit alăturat. Scrieți două numere naturale din intervalul deschis (20,30), care pot fi memorate în variabilele întregi **x1**, și respectiv **x2**, astfel încât valoarea lui **f(x1,3)** să fie 29, iar valoarea lui **f(3,x2)** să fie 1. **(6p.)**

```
int f(int a, int b)
{
    if(a>=b)
        return (a-b)+f(a/b,b);
    return 1;
}
```

3. Se consideră șirul **s**, construit după regula alăturată.

$$s_n = \begin{cases} 1 & \text{dacă } n \leq 2 \\ 3 \cdot s_{n-1} - s_{n-2} & \text{dacă } n > 3 \end{cases}$$

Subprogramul **sir** primește prin parametrul **n** un număr natural ($3 < n < 20$) și furnizează prin parametrul **a** un tablou unidimensional care memorează primii **n** termeni ai șirului **s**, definit după regula de mai sus, astfel încât numerele impare să ocupe primele poziții în tablou, iar cele pare să fie memorate în continuarea celor impare. Scrieți definiția completă a subprogramului **sir**.

Exemplu: dacă **n=6**, primii șase termeni ai șirului sunt 1, 1, 2, 5, 13, 34, iar, după apel, tabloul construit poate fi (1, 1, 5, 13, 2, 34). **(10p.)**

4. Fișierul **bac.in** conține pe prima linie un număr natural **n** ($3 < n < 1000$), iar pe următoarea linie, un șir de **n** numere naturale distincte, de cel mult nouă cifre fiecare. Numerele din șir sunt separate prin câte un spațiu și cel puțin trei dintre ele au ultima cifră egală cu 5.

a) Scrieți un program **C/C++** care citește toate numerele din fișier și, utilizând un algoritm eficient din punct de vedere al timpului de executare și al memoriei utilizate, determină și afișează pe ecran cele mai mari trei numere din șir care au ultima cifră egală cu 5. Numerele determinate sunt afișate în ordine crescătoare, separate prin câte un spațiu. **(6p.)**

Exemplu: dacă fișierul **bac.in** are conținutul

```
10
97 5 11 1 8 6 85 3 25 15
```

alăturat, pe ecran se vor afișa, în această ordine, numerele: 15 25 85

b) Descrieți succint, în limbaj natural (3-4 rânduri), algoritmul utilizat la punctul **a)** și justificați eficiența acestuia. **(4p.)**