

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

1. Câte grafuri neorientate cu 5 noduri există astfel încât fiecare nod să aibă gradul 3 ? (4p.)

- a. 0 b. 3 c. 5 d. 10

2. Care este numărul maxim de frunze pe care le poate avea un arbore binar cu 11 vârfuri ? (4p.)

- a. 5 b. 6 c. 8 d. 10

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

3. Se consideră două cozi $q1$ și $q2$, inițial vide. În prima coadă sunt adăugate trei elemente oarecare. Să se scrie o secvență de instrucțiuni ce utilizează apeluri ale funcțiilor **add** și **out** pentru a muta elementele cozii $q1$ în coada $q2$, astfel încât, la final, suita de operații: **out**($q2$), **out**($q2$), **out**($q2$) să returneze în ordine inversă cele trei valori adăugate inițial în coada $q1$. (Funcția **add**(q, x) adaugă elementul x în coada q . Funcția **out**(q) extrage și returnează un element din coada q .) (6p.)

4. Scrieți un program C/C++ care citește din fișierul standard de intrare (tastatura) șiruri de caractere de forma *cuvânt#tip*, unde *cuvânt* este un șir oarecare de litere iar *tip* poate fi una din literele S,P sau C, semnificația fiind *subiect*, *predicat* sau *complement*. Aplicația va afișa pe ecran toate propozițiile având structura *subiect predicat complement* ce pot fi formate cu ajutorul cuvintelor citite. Datele de intrare se consideră a fi corecte.

Exemplu: dacă la intrare s-au introdus șirurile: **Ion#S Vasile#S alearga#P repede#C scrie#P**, atunci pe ecran se va afișa: **Ion scrie repede, Ion alearga repede, Vasile scrie repede, Vasile alearga repede** (6p.)

5. Scrieți un program C/C++ care citește din fișierul standard de intrare (tastatura) un număr natural n ($n \geq 2$) și o matrice pătratică A de dimensiune $n \times n$, elementele acesteia putând avea doar valorile 0 sau 1. Două elemente $A(i1, j1)$ și $A(i2, j2)$ sunt *adiacente* dacă sunt "vecine" pe o aceeași linie sau coloană: ($i1=i2$ și $|j1-j2|=1$) sau ($j1=j2$ și $|i1-i2|=1$). Un *grup* reprezintă fie un singur element al matricii având valoarea 1, neadiacent cu niciun alt element cu valoarea 1, fie o mulțime de elemente având valoarea 1, fiecare dintre ele fiind adiacent cu cel puțin un alt element din mulțimea respectivă și neadiacent cu niciun alt element din alt grup. Programul trebuie să afișeze pe ecran numărul de grupuri conținute de matrice și coordonatele elementelor acestora.

Exemplu: Numărul de grupuri din matricea 4×4 de mai jos este 3 iar cele trei grupuri sunt: $G1=\{(0,0)\}$, $G2=\{(0,3), (1,2), (1,3), (2,3)\}$, $G3=\{(2,1), (3,0), (3,1)\}$.

1001
0011
0101
1100

(10p.)

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

1. În câte dintre permutările elementelor mulțimii $\{B, I, N, A, R\}$, literele A și B apar pe poziții consecutive, indiferent de ordinea acestora (AB sau BA)? (4p.)

a. 120

b. 72

c. 48

d. 24

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

2. Pentru funcția F definită alăturat, ce valoare va returna apelul $F(1234)$? (6p.)

```
int F(int x) {  
    if(x == 0) return 0;  
    if (x%10%2 == 0) return 2 + F(x/10);  
    return 10 - F(x/10);  
}
```

3. Se consideră o tablă de șah de dimensiune $n \times n$, unde $n \geq 3$ și n piese de tip regină. Două regine sunt în *conflict* dacă ele se găsesc fie pe aceeași linie a tablei, fie pe aceeași coloană, fie pe aceeași diagonală. *Problema celor n regine* constă în plasarea tuturor reginelor pe tabla de joc astfel încât oricare două dintre ele să nu se afle în conflict una cu cealaltă. Ținând cont că în orice soluție pe fiecare linie sau coloană a tablei de joc se va găsi exact o regină, o soluție a problemei va fi un tablou s cu n elemente, astfel încât regina cu numărul i se va găsi la linia i și coloana $s[i]$.

a) Descrieți în limbaj natural o metodă pentru rezolvarea problemei reginelor. Specificați modul în care veți verifica existența unui conflict între două regine. (4p.)

b) Scrieți în limbajul C/C++ o funcție care primește ca argumente dimensiunea n a tablei de joc, un număr de regine k deja plasate și un tablou de numere întregi cuprinse între 0 și $k-1$, reprezentând pozițiile la care au fost plasate cele k regine și construiește o matrice a de dimensiune $n \times n$ astfel:

- $a(i, j) = 0$, dacă nicio regină nu se găsește pe linia i și coloana j și nici nu atacă această poziție;
- $a(i, j) = 1$, dacă există o regină plasată pe linia i și coloana j ;
- $a(i, j) = -p$, dacă nicio regină nu se găsește pe linia i și coloana j dar există $p \geq 1$ regine care atacă această poziție. (6p.)

c) Scrieți în limbajul C/C++ un program care citește din fișierul standard de intrare (tastatura) un număr natural n ($n \geq 3$) și afișează o soluție a problemei celor n regine sau un mesaj corespunzător în cazul în care nu există niciuna. Pentru a eficientiza procesul de căutare a soluției, folosiți matricea a definită la punctul anterior. (10p.)

Exemplu: Pentru $n=4$, o soluție este dată de tabloul $(1, 3, 0, 2)$ corespunzător reprezentării de mai jos (cu 1 au fost marcate reginele plasate):

0100
0001
1000
0010