Universitatea **Facultatea de Informatică**

"Al. I. Cuza", Iaşi **Admitere, septembrie 2012**

Test la INFORMATICĂ
Limbajul C/C++

Se acordă 10 puncte din oficiu. Timpul efectiv de lucru este de 3 ore.

SUBIECTUL I (30 de puncte)

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

1. Variabila întreagă `n` memorează un număr natural de exact 5 cifre. Instrucţiunea care determină eliminarea din numărul `n` a primei şi ultimei cifre este: **(4p.)**

a. `n=n%100/10;`
b. `n=n/10%1000;`
c. `n=n/10%10000;`
d. `n=n%10/1000;`

2. Se consideră algoritmul alăturat, descris în pseudocod.

a. Scrieţi valorile variabilelor `rezultat` şi `iteratii` după execuţia algoritmului dacă secvenţa de intrare este formată din numerele `4` şi `56`. **(6p.)**

b. Pentru un `a` fixat, care sunt cea mai mică, respectiv cea mai mare valoare pe care le poate lua variabila `b` astfel încât valoarea variabilei `iteratii` după execuţia algoritmului să fie exact `5`. **(6p.)**

c. Scrieţi în pseudocod un algoritm care să folosească doar operatorii aritmetici + (plus) şi – (minus) şi care să fie echivalent cu cel dat. **(4p.)**

d. Scrieţi programul `C/C++` corespunzător algoritmului alăturat. **(10p.)**

```
  citeşte a, b (numere naturale)
s ← a
r ← 0
iteratii ← 0
cât timp b > 1 execută
[
  iteratii ← iteratii+1
  dacă b%2=0 atunci
  [
    s ← s*2
    b ← b/2
  altfel
  [
    r ← r+s
    b ← b-1
rezultat ← s+r
```

SUBIECTUL al II-lea **(30 de puncte)**

Pentru fiecare dintre itemii 1 şi 2 scrieţi pe foaia de examen litera corespunzătoare 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`

Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.

3. Se consideră două cozi `q1` şi `q2`, iniţial vide. In prima coadă sunt adăugate trei elemente oarecare. Să se scrie o secvenţă de instrucţiuni ce utilizează apeluri ale funcţiilor `add` şi `out` pentru a muta elementele cozii `q1` în coada `q2`, astfel încât, la final, suita de operaţii: `out(q2), out(q2), out(q2)` să returneze în ordine inversă cele trei valori adăugate iniţial în coada `q1`. (Funcţia `add(q,x)` adaugă elementul `x` în coada `q`. Funcţia `out(q)` extrage şi returnează un element din coada `q`.)

(6p.)

4. Scrieţi un program `C/C++` care citeşte din fişierul standard de intrare (tastatura) şiruri de caractere de forma *cuvânt*`#`*tip*, unde *cuvânt* este un şir oarecare de litere iar *tip* poate fi una din literele S,P sau C, semnificaţia fiind *subiect, predicat* sau *complement*. Aplicaţia va afişa pe ecran toate propoziţiile 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 şirurile: `Ion#S Vasile#S alearga#P repede#C scrie#P`, atunci pe ecran se va afişa: `Ion scrie repede`, `Ion alearga repede`, `Vasile scrie repede`, `Vasile alearga repede` **(6p.)**

5. Scrieţi un program `C/C++` care citeşte din fişierul standard de intrare (tastatura) un număr natural `n` ($n \geq 2$) şi o matrice pătratică `A` de dimensiune `n`×`n`, elementele acesteia putând avea doar valorile `0` sau `1`. Două elemente `A(i1,j1)` şi `A(i2,j2)` sunt *adiacente* dacă sunt "vecine" pe o aceeaşi linie sau coloană: (`i1=i2` şi `|j1-j2|=1`) sau (`j1=j2` şi `|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 mulţime de elemente având valoarea `1`, fiecare dintre ele fiind adiacent cu cel puţin un alt element din mulţimea respectivă şi neadiacent cu niciun alt element din alt grup. Programul trebuie să afişeze pe ecran numărul de grupuri conţinute de matrice şi 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.)

SUBIECTUL al III-lea **(30 de puncte)**

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

1. In câte dintre permutările elementelor mulţimii {B, I, N, A, R}, literele A şi B apar pe poziţii consecutive, indiferent de ordinea acestora (AB sau BA)? **(4p.)**

a. `120` **b.** `72` **c.** `48` **d.** `24`

Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.

2. Pentru funcţia 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 şah de dimensiune `n`×`n`, unde `n` ≥ `3` şi `n` piese de tip regină. Două regine sunt în *conflict* dacă ele se găsesc fie pe aceeaşi linie a tablei, fie pe aceeaşi coloană, fie pe aceeaşi 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. Tinând cont că în orice soluţie pe fiecare linie sau coloană a tablei de joc se va găsi exact o regină, o soluţie a problemei va fi un tablou `s` cu `n` elemente, astfel încât regina cu numarul `i` se va găsi la linia `i` şi coloana `s[i]`.

a) Descrieţi în limbaj natural o metodă pentru rezolvarea problemei reginelor. Specificaţi modul în care veţi verifica existenţa unui conflict între două regine. **(4p.)**

b) Scrieţi în limbajul `C/C++` o funcţie care primeşte ca argumente dimensiunea n a tablei de joc, un număr de regine k deja plasate şi un tablou de numere întregi cuprinse între `0` şi `k-1`, reprezentând poziţiile la care au fost plasate cele `k` regine şi construieşte o matrice `a` de dimensiune `n`×`n` astfel:

- `a(i,j)=0`, dacă nicio regină nu se găseşte pe linia `i` şi coloana `j` şi nici nu atacă această poziţie;
- `a(i,j)=1`, dacă există o regină plasată pe linia `i` şi coloana `j`;
- `a(i,j)=-p`, dacă nicio regină nu se găseşte pe linia `i` şi coloana `j` dar există `p` ≥ `1` regine care atacă această poziţie. **(6p.)**

c) Scrieţi în limbajul `C/C++` un program care citeşte din fişierul standard de intrare (tastatura) un număr natural `n` (`n` ≥ `3`) şi afişează o soluţie a problemei celor `n` regine sau un mesaj corespunzător în cazul în care nu există niciuna. Pentru a eficientiza procesul de căutare a soluţiei, folosiţi matricea `a` definită la punctul anterior. **(10p.)**

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

```
0100
0001
1000
0010
```