Examenul de bacalaureat 2010
PROBA E - d)
Proba scrisă la INFORMATICĂ, limbajul C/C++
Specializarea Matematică-informatică intensiv informatică

MODEL

- **Toate subiectele (I, II şi III) sunt obligatorii. Se acordă 10 puncte din oficiu.**
- **Timpul efectiv de lucru este de 3 ore.**
- **În rezolvările cerute, identificatorii utilizați trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată).**

Subiectul I **(30 de puncte)**

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

1. Care este numărul total de atribuiri efectuate la executarea secvenței de instrucțiuni alăturate? **(4p.)**

```
x=4; y=6;
while (y==6) y=y+1;
if (x==y) x=x+1;
```

a. 4 **b. 3** **c. 2** **d. 5**

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

2. Se consideră algoritmul alăturat descris în pseudocod.

S-a notat cu **x%y** restul împărțirii numărului natural **x** la numărul natural nenul **y** şi cu **[z]** partea întreagă a numărului real **z**.

```
citeşte n (număr natural)
nr←0
p←1
┌cât timp n≠0 execută
│ c ← n%10
│ ┌dacă c>0 şi c < 9 atunci
│ │ c←c+1
│ └■
│ nr ← nr+c*p
│ p ← p*10
│ n ← [n/10]
└■
scrie nr
```

a) Scrieți numărul afişat dacă se citeşte valoarea **n=12939**. **(6p.)**

b) Scrieți toate valorile care pot fi citite pentru variabila **n** astfel încât, în urma executării algoritmului, valoarea afişată să fie **2009**. **(4p.)**

c) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura **cât timp...execută** cu o structură repetitivă de un alt tip. **(6p.)**

d) Scrieți programul **C/C++** corespunzător algoritmului dat. **(10p.)**

Subiectul al II-lea (30 de puncte)

Pentru fiecare dintre itemii 1 şi 2 scrieți pe foaia de examen litera care corespunde răspunsului corect.

1. Se consideră un graf orientat cu **6** noduri numerotate de la **1** la **6** şi cu mulțimea arcelor formată **doar** din arcele:
- de la fiecare nod numerotat cu un număr neprim **i** (**i>1**) la toate nodurile numerotate cu numere ce aparțin mulțimii divizorilor proprii ai lui **i** (divizori diferiți de **1** şi de **i**)
- de la nodul numerotat cu **1** la nodul numerotat cu **6**
- de la fiecare nod numerotat cu un număr prim **i** la nodul numerotat cu **i-1**

Pentru graful dat, care este lungimea celui mai mare drum, format **doar** din noduri distincte? **(4p.)**

a. 6 **b. 5** **c. 3** **d. 4**

2. Câte frunze are arborele cu rădăcină descris prin următorul vector ”de tați”: **(6,5,5,2,0,3,3,3,8,7,7)**? **(4p.)**

a. 1 **b. 2** **c. 5** **d. 4**

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

3. În declararea alăturată, câmpurile **x** şi **y** ale înregistrării pot memora numărătorul, respectiv numitorul unei fracții. Scrieți secvența de instrucțiuni prin executarea căreia se construieşte în variabila **f** o fracție obținută prin însumarea fracțiilor memorate în variabilele **f1** şi **f2**. **(6p.)**

```
struct fractie
  {
    int x,y;
  }f,f1,f2;
```

4. În secvența de instrucțiuni de mai jos, variabila **s** memorează un şir de caractere format doar din litere ale alfabetului englez, iar variabilele **i** şi **n** sunt de tip **int**. Ştiind că în urma executării secvenței s-a afişat succesiunea de caractere **eied*eael*** scrieți care este şirul de caractere memorat de variabila **s**. **(6p.)**

```
//C
n=strlen(s);
for(i=0;i<n;i++)
  if (s[i]==’e’) printf(”%c”,’*’);
  else printf("%c%c",’e’,s[i]);
```

```
//C++
n=strlen(s);
for(i=0;i<n;i++)
  if (s[i]==’e’) cout<<’*’;
  else cout<<'e'<<s[i];
```

5. Scrieți un program **C/C++** care citeşte de la tastatură un număr natural **n** (**2≤n≤24**) şi construieşte în memorie o matrice cu **n** linii şi **n** coloane ale cărei elemente vor primi valori după cum urmează:
- elementele aflate pe diagonala principală a matricei vor primi valoarea **0**
- elementele de pe prima coloană, cu excepția celui aflat pe diagonala principală vor primi valoarea **n**
- elementele de pe a doua coloană, cu excepția celui aflat pe diagonala principală vor primi valoarea **n-1**
- ...
- elementele de pe ultima coloană, cu excepția celui aflat pe diagonala principală vor primi valoarea **1**

Programul va afişa matricea astfel construită pe ecran, câte o linie a matricei pe câte o linie a ecranului, cu câte un spațiu între elementele fiecărei linii (ca în exemplu).

Exemplu: pentru **n=4** se va afişa matricea alăturată. **(10p.)**

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

Subiectul al III-lea (30 de puncte)

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

1. In timpul procesului de generare a permutărilor mulțimii **{1,2,…,n}** prin metoda backtracking, în tabloul unidimensional **x** este plasat un element x_k (**2≤k≤n**). Acesta este considerat valid dacă este îndeplinită condiția: **(6p.)**

a. $x_k \notin \{x_1, x_2, \ldots, x_{k-1}\}$ **b.** $x_k \neq x_{k-1}$

c. $x_k \notin \{x_1, x_2, \ldots, x_n\}$ **d.** $x_k \neq x_{k-1}$ şi $x_k \neq x_{k+1}$

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

2. Considerăm subprogramul recursiv definit alăturat. Ce se va afişa în urma apelului de mai jos?
f('C'); **(4p.)**

```
void f(char c)
{ if (c>'A') f(c-1);
  cout<<c; | printf("%c",c);
  if (c>'A') f(c-1);
}
```

3. **a)** Scrieți definiția completă a unui subprogram, **nz**, cu un parametru întreg **n** (**0<n≤32000**), care returnează numărul zerourilor de la sfârşitul numărului **n!=1·2·…·n**. **(6p.)**
b) Scrieți programul **C/C++** care citeşte de la tastatură un număr natural **k** (**0<k≤1500**) şi determină, folosind apeluri ale subprogramului **nz**, cel mai mic număr natural **n** pentru care **n!** are cel puțin **k** zerouri la sfârşit. Numărul determinat se afişează pe ecran. **(4p.)**

4. Scrieți programul **C/C++** care citeşte din fişierul text **BAC.TXT** numărul întreg **n** (**1≤n≤10000**) şi un şir de **n** perechi de numere întregi **a b** (**1≤a≤b≤32000**), fiecare pereche fiind scrisă pe o linie nouă a fişierului, cu un spațiu între cele două numere. Programul afişează pe ecran pentru fiecare pereche **a,b** cel mai mare număr natural din intervalul închis **[a,b]** care este o putere a lui **2** sau numărul **0** dacă nu există nicio putere a lui **2** în intervalul respectiv. Numerele afişate pe ecran se scriu în linie, separate prin câte un spațiu. Un număr **p** este putere a lui **2** dacă există un număr natural **k** astfel încât $p=2^k$.
Exemplu: dacă fişierul **BAC.TXT** conține numerele
3
2 69
10 20
19 25
se va afişa: **64 16 0**. **(10p.)**