

**EXAMENUL DE BACALAUREAT – 2009**  
**Proba scrisă la INFORMATICĂ**  
**PROBA E, limbajul C/C++**  
**Specializarea Matematică-informatică**

- ◆ Toate subiectele 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  $y$  și cu  $[z]$  partea întreagă a numărului real  $z$ .
- a) Scrieți numărul afișat dacă se citește valoarea  $n=12939$ . **(6p.)**
- b) Scrieți toate valorile care pot fi citite pentru variabila  $n$  astfel încât, în urma executării algoritmului, valoarea afișată 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.)**
- ```
citește n (număr natural)
nr ← 0
p ← 1
cât timp n ≠ 0 execută
  c ← n % 10
  dacă c > 0 și c < 9 atunci
    c ← c + 1
  nr ← nr + c * p
  p ← p * 10
  n ← [n / 10]
scrie nr
```

**Subiectul II (30 de puncte)**

**Pentru fiecare dintre itemii 1 și 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 și 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 și 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, câte dintre nodurile grafului au gradul exterior strict mai mare decât gradul interior? **(4p.)**
- a. 1                                      b. 2                                      c. 4                                      d. 3
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 declarația alăturată, câmpurile  $x$  și  $y$  ale înregistrării pot memora numărătorul, respectiv numitorul unei fracții. Scrieți secvența de instrucțiuni prin executarea căreia se construiește în variabila  $f$  o fracție obținută prin însumarea fracțiilor memorate în variabilele  $f1$  și  $f2$ . **(6p.)**
- |  |  |  |
|--|--|--|
| <pre>struct fractie {     int x,y; }f,f1,f2;</pre> |  | <pre>struct fractie {     int x,y; }f,f1,f2;</pre> |
|--|--|--|
4. În secvența de instrucțiuni de mai jos, variabila  $s$  memorează un șir de caractere format doar din litere ale alfabetului englez, iar variabilele  $i$  și  $n$  sunt de tip `int`. Știind că în urma executării secvenței s-a afișat succesiunea de caractere `eeleeeeee` scrieți care este șirul de caractere memorat de variabila  $s$ . **(6p.)**
- ```
n=strlen(s);
for(i=0;i<n;i++)
    printf("%c%c",s[i], 'e'); | cout<<s[i]<<'e';
```
5. Scrieți un program C/C++ care citește de la tastatură un număr natural  $n$  ( $2 \leq n \leq 24$ ) și construiește în memorie o matrice cu  $n$  linii și  $n$  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 afișa 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). **(10p.)**
- |                                                                |  |                                            |
|----------------------------------------------------------------|--|--------------------------------------------|
| <pre>Exemplu: pentru n=4 se va afișa matricea alăturată.</pre> |  | <pre>0 3 2 1 4 0 2 1 4 3 0 1 4 3 2 0</pre> |
|----------------------------------------------------------------|--|--------------------------------------------|

**Subiectul III (30 de puncte)**

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

1. În timpul procesului de generare a permutărilor mulțimii  $\{1, 2, \dots, n\}$  prin metoda backtracking, în tabloul unidimensional  $x$  este plasat un element  $x_k$  ( $2 \leq k \leq n$ ). Acesta este considerat valid dacă este îndeplinită condiția: **(6p.)**
- a.  $x_k \notin \{x_1, x_2, \dots, x_{k-1}\}$                       b.  $x_k \neq x_{k-1}$
- c.  $x_k \notin \{x_1, x_2, \dots, x_n\}$                       d.  $x_k \neq x_{k-1}$  și  $x_k \neq x_{k+1}$

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

2. Considerăm subprogramul recursiv definit alăturat. Ce se va afișa în urma apelului de mai jos?  
`f('B');` **(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 unui subprogram, **nz**, cu un parametru întreg, **n** ( $0 < n \leq 12$ ), care returnează numărul zerourilor de la sfârșitul numărului  $n! = 1 \cdot 2 \cdot \dots \cdot n$ . **(6p.)**  
b) Scrieți programul **C/C++** care citește de la tastatură un număr natural **k** ( $0 < k \leq 2$ ) și determină, folosind apeluri ale subprogramului **nz**, cel mai mic număr natural **n** pentru care  $n!$  are cel puțin **k** zerouri la sfârșit. Numărul determinat se afișează pe ecran. **(4p.)**
4. Scrieți programul **C/C++** care citește din fișierul text **BAC.TXT** numărul întreg **n** ( $1 \leq n \leq 10000$ ) și un șir de **n** perechi de numere întregi **a b** ( $1 \leq a \leq b \leq 32000$ ), fiecare pereche fiind scrisă pe o linie nouă a fișierului, cu un spațiu între cele două numere. Programul afișează 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 afișate 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 fișierul **BAC.TXT** conține numerele  
3  
2 69  
10 20  
19 25  
se va afișa: 64 16 0. **(10p.)**