

EXAMENUL DE BACALAUREAT – 2009
Proba scrisă la INFORMATICĂ
PROBA E, limbajul C/C++
Specializarea Matematică-informatică intensiv 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. Variabilele **x** și **y** sunt de tip întreg, **x** memorând valoarea 8, iar **y** valoarea 6. Care dintre expresiile C/C++ de mai jos are valoarea 0? **(4p.)**
- a. $3*x-4*y==0$ b. $(x+y)/2 > x*y+1$
c. $!(x/2+2==y)$ d. $x-y+3!=0$

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 valoarea care se va afișa dacă se citește pentru **n** valoarea 296385, iar pentru **k** valoarea 3. **(6p.)**
- b) Dacă se citește pentru **k** valoarea 4, scrieți cea mai mare valoare de 5 cifre care poate fi citită pentru **n** astfel încât numărul afișat în urma executării algoritmului să fie 1. **(4p.)**

```
citește n,k (numere naturale)
p←1
cât timp n>0 și k>0 execută
|   c←n%10
|   dacă c%2=1 atunci
|       p←p*c
|   n←[n/10]
|   k←k-1
scrie p
```

- c) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d) Scrieți în pseudocod un algoritm echivalent cu algoritmul dat, în care să se înlocuiască structura repetitivă **cât timp...execută**, cu o structură repetitivă **pentru...execută**. **(6p.)**

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 graful orientat reprezentat prin matricea de adiacență alăturată. Care este lungimea maximă a unui drum, de la vârful 4 până la vârful 6, format din vârfuri distincte două câte două (lungimea unui drum este egală cu numărul de arce care compun acel drum)? **(4p.)**
- | | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 |
- a. 4 b. 3 c. 1 d. 5
2. Câte grafuri neorientate, distincte, cu 5 vârfuri, se pot construi? Două grafuri se consideră distincte dacă matricele lor de adiacență sunt diferite. **(4p.)**
- a. 5^4 b. 5^2 c. 2^{10} d. 4^{10}

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

3. Ce se afișează pe ecran în urma executării secvenței de program alăturate, în care variabila *s* memorează un șir cu cel mult 12 caractere, iar variabila *i* este de tip întreg? **(6p.)**
- ```
strcpy(s,"abracadabra");
i=0;
cout<<strlen(s); | printf("%d",strlen(s));
while (i<strlen(s))
 if (s[i]=='a')
 strcpy(s+i,s+i+1);
 else
 i=i+1;
cout<<" "<<s; | printf(" %s",s);
```
4. Într-o listă liniară simplu înlănțuită, alocată dinamic, cu cel puțin 3 noduri, fiecare element reține în câmpul *nr* un număr real, iar în câmpul *urm* adresa următorului element din listă sau valoarea *NULL* în cazul în care este ultimul nod al listei. Dacă *p* reține adresa primului element din listă, scrieți o **expresie C/C++** a cărei valoare este egală cu suma valorilor reale reținute în primele trei noduri ale listei. **(6p.)**
5. Scrieți un program C/C++ care citește de la tastatură două valori naturale nenule *m* și *n* ( $m \leq 10$ ,  $n \leq 10$ ) și apoi *m*\**n* numere naturale nenule cu cel mult 4 cifre fiecare, reprezentând elementele unei matrice cu *m* linii și *n* coloane. Programul determină apoi valorile minime de pe fiecare linie a matricei și afișează pe ecran cea mai mare valoare dintre aceste minime.

**Exemplu:** pentru *m*=3, *n*=5 și matricea  $\begin{pmatrix} 3 & 6 & 5 & 4 & 7 \\ 9 & 6 & 12 & 9 & 10 \\ 5 & 13 & 7 & 2 & 3 \end{pmatrix}$ , se afișează pe ecran valoarea 6

(cea mai mică valoare de pe prima linie a matricei este 3, cea mai mică valoare de pe linia a doua este 6, cea mai mică valoare de pe linia a treia este 2. Cea mai mare dintre aceste trei valori este 6). **(10p.)**

**Subiectul III (30 de puncte)**

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

1. Un algoritm de tip backtracking generează, în ordine lexicografică, toate șirurile de 5 cifre 0 și 1 cu proprietatea că nu există mai mult de două cifre 0 pe poziții consecutive. Primele 7 soluții generate sunt: 00100, 00101, 00110, 00111, 01001, 01010, 01011. Care este a 8-a soluție generată de acest algoritm? **(4p.)**
- a. 01110                      b. 01100                      c. 01011                      d. 01101

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

2. Pentru funcția  $f$  definită alăturat, stabiliți care este valoarea  $f(5)$ . Dar  $f(23159)$ ? **(6p.)**
- ```
int f(int n){
    int c;
    if (n==0) return 9;
    else
    {c=f(n/10);
      if (n%10<c) return n%10;
      else return c;
    }
}
```
3. Fișierul text **numere.txt** conține pe prima linie un număr natural n ($n < 30000$), iar pe a doua linie n numere întregi având maximum 4 cifre fiecare. Se cere să se afișeze pe ecran un șir de n numere întregi, cu proprietatea că valoarea termenului de pe poziția i ($i=1, 2, \dots, n$) din acest șir este egală cu cea mai mare dintre primele i valori de pe a doua linie a fișierului **numere.txt**.
- a) Descrieți pe scurt un algoritm de rezolvare, eficient din punct de vedere al timpului de executare și al spațiului de memorie utilizat, explicând în ce constă eficiența sa. **(4p.)**
- b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**
- Exemplu:** dacă fișierul **numere.txt** are conținutul
- | | |
|---------------------------|--------------------------|
| 12 | 12 |
| 4 6 3 7 8 1 6 2 7 9 10 8 | 4 6 3 7 8 1 6 2 7 9 10 8 |
| 4 6 6 7 8 8 8 8 8 9 10 10 | |
4. a) Scrieți doar antetul funcției **sum** care primește ca parametru un număr natural nenul x cu maximum 9 cifre și returnează suma divizorilor numărului x . **(3p.)**
- Exemplu:** **sum(6)** are valoarea 12 ($=1+2+3+6$).
- b) Să se scrie un program C/C++ care citește de la tastatură un număr natural n ($0 < n < 25$) și apoi n numere naturale nenule cu maximum 9 cifre fiecare. Programul calculează, folosind apeluri ale funcției **sum**, și afișează pe ecran câte numere prime conține șirul citit.
- Exemplu:** pentru $n=5$ și valorile 12 3 9 7 1 se va afișa pe ecran valoarea 2 (în șirul dat există două numere prime și anume 3 și 7). **(7p.)**