

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.

- | | |
|---|---|
| <p>1. Câte atribuiri se execută, în total, în secvența alăturată, dacă n și p sunt variabile de tip întreg? (4p.)</p> | <pre>p=1; n=279; while (n>=100) { p=p*10; n=n-100; }</pre> |
| a. 4 | b. 6 |
| c. 2 | d. 8 |

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 numerele care se vor afișa în urma executării algoritmului dacă se citesc valorile **a=312** și **b=1354**. (6p.)
- b) Scrieți câte o valoare care poate fi citită pentru variabila **a**, respectiv **b**, astfel încât algoritmul să afișeze exact **2** valori. (4p.)
- c) Scrieți programul **C/C++** corespunzător algoritmului dat. (10p.)
- d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura **pentru...execută** cu o structură repetitivă de alt tip. (6p.)

```
citește a,b
      (numere naturale)
a←[a/10]%10*a%10
b←[b/10]%10*b%10
pentru i←a,b execută
|   dacă[i/10]=i%10 atunci
|       scrie i%10
|   ■
|   ■
```

Subiectul II (30 de puncte)

Pentru fiecare dintre itemii următori, scrieți pe foaia de examen litera care corespunde răspunsului corect.

1. Un graf neorientat este complet dacă oricare două noduri distincte ale sale sunt adiacente. Care este numărul de muchii care trebuie eliminate dintr-un graf neorientat, complet, cu 7 noduri, astfel încât graful parțial obținut să fie arbore? (4p.)
a. 15 b. 1 c. 6 d. 21
2. Fiecare element al unei liste liniare, simplu înlanțuite, alocată dinamic, reține în câmpul **nr** un număr întreg, iar în câmpul **adr** adresa elementului următor din listă sau **NULL** dacă nu există un element următor. Dacă **p** reține adresa primului element, iar lista are cel puțin două elemente, care dintre următoarele secvențe de instrucțiuni copiază în câmpul **nr** al celui de-al doilea element al listei, conținutul câmpului **nr** al primului element din listă? (4p.)
a. **p->nr=p->adr->nr;** b. **p->adr=p->nr;**
c. **p->adr->nr=p->nr;** d. **p->adr->adr->nr=p->nr;**

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

3. Ce se va afișa în urma executării secvenței de program alăturate, știind că variabila **x** memorează un șir cu cel mult **100** de caractere, iar variabila **i** este de tip întreg ? (6p.)

```
strcpy(x,"bac2009");  
cout<<x<<endl; | printf("%s\n",x);  
for(i=0;i<strlen(x);i++)  
    if (strchr("0123456789",x[i])==0)  
        cout<<x[i]; | printf("%c",x[i]);
```
4. Fiecare dintre variabilele **a** și **b**, declarate alăturat, memorează simultan coordonatele reale ale câte unui punct în planul **xOy**.
Completați punctele de suspensie din secvența următoare de program, astfel încât, în urma executării ei, să se afișeze cuvântul **DA** dacă dreapta, determinată de punctele cu coordonatele memorate în variabilele **a** și **b**, este paralelă cu axa **Ox**, respectiv cuvântul **NU** în caz contrar.

```
if(...) cout<<"DA"; | printf("DA");  
else cout<<"NU" | printf("NU");
```

 (6p.)
5. Scrieți un program **C/C++** care citește de la tastatură un număr natural **n** ($2 < n < 20$), construiește în memorie și afișează pe ecran o matrice cu **n** linii și **n** coloane, numerotate de la **1** la **n**, în care fiecare element aflat pe o linie impară este egal cu suma dintre indicii liniei și coloanei pe care se află și fiecare element aflat pe o linie pară este egal cu cel mai mic dintre elementele aflate pe linia anterioară și pe aceeași coloană cu el sau pe linia anterioară și pe una dintre coloanele vecine cu cea pe care se află el.
Elementele matricei vor fi afișate pe ecran, câte o linie a matricei pe câte o linie a ecranului cu câte un spațiu între elementele fiecărei linii.
Exemplu: pentru **n=5** se va afișa matricea alăturată. (10p.)

2	3	4	5	6
2	2	3	4	5
4	5	6	7	8
4	4	5	6	7
6	7	8	9	10

Subiectul III (30 de puncte)

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

1. Funcția **F** are definiția alăturată. Ce valoare are **F(18)**? (4p.)

```
int F(int x){  
    if (x<=1) return x;  
    else return x+F(x-2);  
}
```

a. 90

b. 171

c. 91

d. 18

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare

2. Un algoritm generează în ordine crescătoare, toate numerele de **n** cifre (**n<9**), cu cifre distincte, care nu au două cifre pare alăturate. Dacă pentru **n=5**, primele cinci soluții generate sunt **10325**, **10327**, **10329**, **10345**, **10347**, precizați care sunt următoarele **trei** soluții generate, în ordinea obținerii lor. (6p.)
3. Subprogramul **aranjare** are doi parametri: **a** prin care primește un tablou unidimensional cu maximum **100** de numere reale nenule și **n**, numărul de elemente din tablou. Subprogramul rearanjează elementele tabloului unidimensional astfel încât toate valorile negative să se afle pe primele poziții, iar valorile pozitive în continuarea celor negative. Ordinea în cadrul secvenței de elemente pozitive, respectiv în cadrul secvenței de elemente negative, poate fi oricare. Tabloul modificat va fi furnizat tot prin intermediul parametrului **a**.
Exemplu: dacă tabloul are **6** elemente și este de forma **(12, -7.5, 6.5, -3, -8, 7.5)**, după apel, acesta ar putea fi: **(-7.5, -3, -8, 12, 6.5, 7.5)**.
Scrieți definiția completă a subprogramului **aranjare**. (10p.)
4. În fișierul **nr1.txt** este memorată pe prima linie o valoare naturală **n** de cel mult **8** cifre, iar pe linia următoare sunt memorate **n** numere naturale, cu maximum **4** cifre fiecare, ordonate strict crescător și separate prin câte un spațiu. În fișierul **nr2.txt** este memorată pe prima linie o valoare naturală **m** de cel mult **8** cifre, iar pe linia următoare sunt memorate **m** numere naturale, cu maximum **4** cifre fiecare, ordonate strict crescător și separate prin câte un spațiu. Se cere afișarea pe ecran, separate prin câte un spațiu, în ordine strict crescătoare, a tuturor numerelor aflate pe a doua linie în cel puțin unul dintre cele două fișiere. În cazul în care un număr apare în ambele fișiere, el va fi afișat o singură dată. Alegeți un algoritm de rezolvare eficient din punct de vedere al memoriei utilizate și al timpului de executare.
Exemplu: pentru următoarele fișiere:
nr1.txt
5
3 6 8 9 12
se va afișa **2 3 5 6 7 8 9 12 13**.
nr2.txt
6
2 3 5 7 9 13
a) Descrieți succint, în limbaj natural, strategia de rezolvare și justificați eficiența algoritmului ales. (4p.)
b) Scrieți programul **C/C++** corespunzător algoritmului ales. (6p.)