

Examenul de bacalaureat 2006
Probă scrisă la INFORMATICĂ, profil matematică-informatică
Varianta C/C++

Simulare 2006

Proba E

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

I. Pentru fiecare din itemii de la 1 la 8, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 5 puncte.

1. Nodurile grafului neorientat dat prin matricea de adiacență
- | | |
|--|-----------|
| alăturată sunt etichetate cu numere de la 1 la 5 corespunzător liniilor matricei. Este acest graf un arbore? | 0 1 0 0 0 |
| | 1 0 0 0 0 |
| | 0 0 0 1 0 |
| | 0 0 1 0 1 |
| | 0 0 0 1 0 |
- a. da, dacă se elimină muchia [1,2] b. da, dacă se adaugă muchia [3,5]
c. da, dacă se adaugă muchia [2,3] d. da
2. Pentru șirul format din cele 20 de valori citite, algoritmul alăturat determină și afișează:
- | | |
|--|-------------------------|
| | pentru i=1,20 execută |
| | citește a (număr real) |
| | dacă a≥0 atunci scrie a |
| | sfârșit pentru |
- a. toate componentele pozitive b. numărul de componente pozitive
c. cea mai mare componentă pozitivă d. toate componentele negative
3. Valoarea variabilei **x** este un număr natural. Reprezentarea în baza 10 a lui **x** este formată din exact 3 cifre dacă și numai dacă:
- a. `x/100==0 && x%1000==0` b. `x/1000>0 && x/100==0`
c. `x%100>0 && x/1000==0` d. `x/1000==0 && x/100>0`
4. Care dintre următoarele variante este o secvență corectă de declarare a unui tablou bidimensional format din 200 de componente numere reale?
- a. `float a[199]` b. `float a[10][20]`
c. `float a[200]` d. `float a[9][19]`
5. Un graf neorientat conex are **n** noduri și **2*n** muchii dacă și numai dacă:
- a. **n>5** b. **n<10** c. **n<9** d. **n>4**
6. Nodurile unei liste dublu înlănțuite rețin în câmpurile **info**, **adp** și **adu** o informație numerică, adresa nodului precedent și respectiv adresa nodului următor din listă. Știind că lista este corect construită și că două noduri **p** și **q** ale acesteia se învecinează, atunci:
- a. `p->adp==q->adu` b. `p->adu==q->adu` c. `p->adp==q` d. `p->adp==q->adp`
7. Se consideră o listă simplu înlănțuită ale cărei noduri rețin în câmpul **urm** adresa nodului următor al listei sau **NULL** dacă nu există un element următor. Pentru inserarea unui nod aflat la adresa **p** imediat după un nod al listei aflat la adresa **q**, se utilizează unele dintre următoarele atribuiri: 1) `p->urm=q`; 2) `q->urm=p`; 3) `p=q->urm`; 4) `q=p->urm`; 5) `p->urm=q->urm`; 6) `q->urm=p->urm`; .
Stabiliți care dintre acestea se utilizează și în ce ordine:
- a. 3 6 b. 2 4 c. 5 2 d. 2 3
8. Se generează toate submulțimile formate din două elemente ale mulțimii {5, 6, 7, 8} în ordinea: 5 6, 5 7, 5 8, 6 7, 6 8 și 7 8. Dacă se utilizează exact aceeași metodă pentru a genera submulțimile de trei elemente ale mulțimii {2, 3, 4, 5, 6}, atunci penultima submulțime generată este:
- a. 3 4 5 b. 3 5 6 c. 4 5 6 d. 2 5 6

II. Se consideră programul pseudocod alăturat:

S-a notat cu $x \% y$ restul împărțirii numerelor întregi x și y și cu $[x]$ partea întreagă a numărului real x .

1. Care este litera afișată pentru $n=1602$ și $s=9$? (5p)
2. Pentru $n=53707$, scrieți o valoare de două cifre pentru variabila s astfel încât rezultatul afișat să fie **B**. (3p)
3. Pentru $n=573$, scrieți câte valori formate dintr-o singură cifră există pentru variabila s astfel încât rezultatul afișat să fie **A**. (2p)
4. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

```
citește n,s (numere naturale)
cât timp n>0 execută
    s←s-n%10
    n←[n/10]
sfârșit cât timp
dacă s=0 atunci
    scrie "A"
altfel
    scrie "B"
sfârșit dacă
```

III.

1. Din fișierul **DATE.TXT** se citesc n și m (n, m numere naturale, $0 < m < n < 5000$) de pe prima linie, apoi n numere naturale $a_1, a_2, \dots, a_n$ de pe linia a doua și apoi m numere naturale $b_1, b_2, \dots, b_m$ de pe linia a treia a fișierului. Să se verifice dacă șirul b citit de pe a treia linie a fișierului se poate obține din șirul a (aflat pe a doua linie a fișierului) prin eliminarea unora dintre componentele șirului a .
a) Se cere să se aleagă un algoritm eficient și să se realizeze o descriere de 3-4 rânduri a algoritmului ales justificându-se eficiența acestuia. (2p.)
b) Să se scrie programul C/C++ ce realizează prelucrarea descrisă și afișează pe ecran un mesaj corespunzător. (8p.)
De exemplu, pentru fișierul **DATE.TXT** cu conținutul:
5 3
1 6 3 1 3
6 1 3
se afișează mesajul "b se poate obtine din a"
2. Pentru o valoare n (număr natural de cel mult o cifră) citită de la tastatură se cere să se scrie un program care construiește în memorie o matrice de n linii și n coloane formată numai din elemente egale cu 1 și 2, elementele aflate pe cele 4 margini ale tabloului fiind egale cu 1, cele din interior fiind egale cu 2. Elementele matricei se scriu pe ecran, pe linii, ca în exemplul următor:
Pentru $n=4$, se afișează
1 1 1 1
1 2 2 1
1 2 2 1
1 1 1 1 (10p.)
3. Realizați următoarele cerințe utilizând limbajul C sau C++:
a) Scrieți definiția unui subprogram **mindiv** care determină cel mai mic dintre divizorii mai mari decât 1 ai unui număr natural transmis prin intermediul parametrului a ($a > 1$) și returnează acest divizor prin intermediul parametrului b . (4p.)
b) Scrieți programul care citește două numere naturale a și b ($a < b$) și determină cel mai mare număr prim din intervalul închis $[a, b]$ cu ajutorul subprogramului definit la a). Dacă nu există un astfel de număr, se va afișa mesajul **Nu exista**. (6p.)