

EXAMENUL DE BACALAUREAT – 2007
Proba scrisă la INFORMATICĂ
PROBA E, limbajul C/C++

Varianta 77

- ♦ Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- ♦ Timpul efectiv de lucru este de 3 ore.
- ♦ În programele cerute la subiectele II și III, 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 (40 de puncte)

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. Într-o listă liniară dublu înălțuită cu cel puțin 2 elemente, fiecare element memorează în câmpul **urm** adresa elementului următor în listă și în câmpul **prec** adresa elementului precedent în listă, iar în variabila **p** adresa primului element din listă și în variabila **q** adresa ultimului element din listă. Dacă valoarea expresiei **p->urm->prec->urm==q->prec** este diferită de 0, câte elemente are lista?
 - a. 2
 - b. 4
 - c. 3
 - d. 5
2. Variabila **elev** este utilizată pentru a memora numele unui elev și media sa la bacalaureat. Declararea corectă a variabilei **elev** este:
 - a. `char elev[60];`
 - b. `struct {char nume[30]; float medie;} elev;`
 - c. `float elev[2];`
 - d. `struct elev {char nume[30]; float medie};`
3. Fie graful orientat cu 8 vârfuri și arcele **[1,2],[2,3],[3,1],[4,5],[5,6],[5,7],[6,7],[7,4],[8,7]**. Numărul de vârfuri cu proprietatea că gradul interior este egal cu gradul exterior este:
 - a. 2
 - b. 7
 - c. 0
 - d. 5
4. Care dintre următoarele secvențe determină în mod corect maximul dintre 3 numere **a, b** și **c** și atribuie această valoare variabilei **max**?
 - a. `if(a>b && a>c) max=a;`
`if(b>a && b>c) max=b;`
`else max=c;`
 - b. `if (a>b>c) max=a;`
`if (b>a>c) max=b;`
`if (c>a>b) max=c;`
 - c. `if (a>b) max=a;`
`else if (b>c) max=b;`
`else max=c;`
 - d. `max=a;`
`if (b>max) max=b;`
`if (c>max) max=c;`
5. Care dintre următoarele matrice este matricea de adiacență a unui un graf care are proprietatea că este arbore?

<ol style="list-style-type: none"> a. $\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ 	<ol style="list-style-type: none"> b. $\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$ 	<ol style="list-style-type: none"> c. $\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ 	<ol style="list-style-type: none"> d. $\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$
---	--	---	---
6. Ce se va afișa pe ecran la apelul **p(630,2,1)**?


```
void p(int x,int y,int k)
{if(x>1)
  if(x%y==0)
    {if(k==1)
      printf("%d ",y); | cout<<y<<" ";
      p(x/y,y,k+1);
    }
  else p(x,y+1,1);
}
```

 - a. 2
 - b. 2 3 3 5 7
 - c. 2 3 5 7
 - d. 2 1 3 2 5 1 7 1
7. Problema generării tuturor codurilor formate din 6 cifre distincte (cifre din mulțimea **{0,1,2,3,4,5,6,7,8,9}**) este similară cu generarea tuturor:
 - a. submultimilor cu 6 elemente ale mulțimii **{0,1,2,3,4,5,6,7,8,9}**
 - b. permutărilor unei mulțimi cu 6 elemente
 - c. aranjamentelor de 10 elemente luate câte 6
 - d. elementelor produsului cartezian **A⁶** unde **A** este o mulțime cu 10 elemente

8. Într-o listă liniară simplu înlănțuită fiecare element reține în câmpul **adr** adresa elementului următor din listă, iar în câmpul **inf** un număr întreg. Inserarea în listă a unui nou element indicat de pointerul **p** după un element din listă indicat de pointerul **q** se realizează cu secvența de instrucțiuni:
- p->adr=q; q->adr=p;**
 - q->adr=p; p->adr=q->adr;**
 - p->adr=q->adr; q->adr=p;**
 - p=q; p->adr=q->adr;**

SUBIECTUL II (20 de puncte)

Se consideră programul pseudocod alăturat:

S-a notat cu $x \div y$ restul împărțirii lui x la y și cu $[x]$ partea întreagă a numărului real x

- Ce se va afișa pentru $n=3533$ și $c=3$? (5p.)
- Scrieți o valoare pentru n și una pentru c astfel încât să se afișeze valoarea 0. (2p.)
- Scrieți programul C/C++ corespunzător algoritmului dat. (8p.)
- Scrieți un algoritm echivalent cu algoritmul dat, dar care să utilizeze alt tip de structură repetitivă (5p.)

```

citește n,c (n,c numere
naturale, n>0, 0≤c≤9 )
k←0
cât timp n%10=c execută
|   n←[n/10]
|   k←k+1
└─
scrie k

```

SUBIECTUL III (30 de puncte)

- Scrieți programul C/C++ care citește de la tastatură un număr natural n ($0 < n < 100$) și creează fișierul text **BAC.TXT** care conține pe prima linie, în ordine descrescătoare, toate numerele naturale de la n la 1, pe a doua linie în ordine descrescătoare, toate numerele naturale de la $n-1$ la 1 etc... pe linia $n-1$ numerele 2 1, iar pe ultima linie numărul 1. Pe fiecare linie numerele vor fi despărțite prin câte un spațiu.
Exemplu: dacă se citește $n=3$, atunci **BAC.TXT** va conține:
3 2 1
2 1
1 (10p.)
- Scrieți definiția completă a unui subprogram **elimin** care:
 - primește ca parametri două șiruri **s1** și **s2** de maximum 250 de litere mici ale alfabetului englez
 - returnează prin parametrul **s** șirul de litere distincte care apar fie în șirul **s1**, fie în șirul **s2**, fie în ambele șiruri.
De exemplu, dacă inițial în **s1** este memorat șirul **aebacdbaced**, iar în **s2** este memorat **aeeffgg** în final în **s** va fi memorat **aebcdfg**, nu neapărat în această ordine. (5p.)
 - Scrieți declarațiile de date și programul principal în care se citește de la tastatură un număr n ($1 < n < 20$) și apoi n șiruri de caractere formate din maximum 250 de litere mici ale alfabetului și afișează pe ecran un șir de caractere distincte format din toate literele care apar cel puțin în unul din șirurile citite, utilizând apeluri ale subprogramului **elimin**.
De exemplu, dacă $n=4$ și cele 4 șiruri sunt **xabc**, **yyaad**, **abcd**, **ccddzz**, se va afișa pe ecran **abcdxzy**, nu neapărat în această ordine. (5p.)
- Se citește de la tastatură un număr natural n și apoi n numere întregi de cel mult 6 cifre fiecare. Să se determine cea mai mare valoare negativă dintre cele citite, precum și numărul de apariții ale acestei valori printre numerele citite. Alegeți o metodă eficientă atât din punctul de vedere al timpului de executare cât și al gestionării memoriei.
 - Descrieți pe scurt metoda folosită explicând eficiența acesteia (3-4 rânduri). (2p.)
 - Scrieți programul C/C++ corespunzător metodei descrise la punctul a). Se va afișa valoarea negativă maximă și numărul de apariții sau mesajul **NU EXISTA** dacă nu există valori negative printre numerele citite.
De exemplu, pentru $n=8$ și numerele 10, 3, -4, -7, -4, -7, 11, -4 se vor afișa numerele -4 și 3, deoarece -4 este cea mai mare valoare negativă și apare de 3 ori. (8p.)