

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

Varianta 40

- ♦ 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. Să se precizeze câte tablouri bidimensionale cu 3 linii și 3 coloane, distincte, simetrice față de diagonala principală, au 0 pe diagonala principală și în rest numai valori din mulțimea $\{0, 1\}$? Două tablouri bidimensionale sunt distincte dacă diferă prin cel puțin un element.
 - a. 7
 - b. 10
 - c. 64
 - d. 8
2. Se consideră o listă liniară simplu înlănțuită ale cărei noduri rețin în câmpul **info** informația fiecărui nod, iar în câmpul **urm** adresa nodului următor. Lista are cel puțin patru elemente. Știind că **p** reține adresa celui de-al doilea nod din listă, ce instrucțiune trebuie executată pentru a afișa informația memorată în al patrulea nod?
 - a. `printf("%d", p->urm->info->urm); | cout<<p->urm->info->urm;`
 - b. `printf("%d", p->urm->info); | cout<<p->urm->info;`
 - c. `printf("%d", p->urm->urm->info); | cout<<p->urm->urm->info;`
 - d. `printf("%d", p->urm->urm->urm->info); | cout<<p->urm->urm->urm->info;`
3. În secvența alăturată **x % y** semnifică restul împărțirii întregi a lui **x** la **y**, iar **x/y** câtul împărțirii întregi a lui **x** la **y**.
 Pentru **n>2**, natural, secvența alăturată afișează 1 dacă și numai dacă:

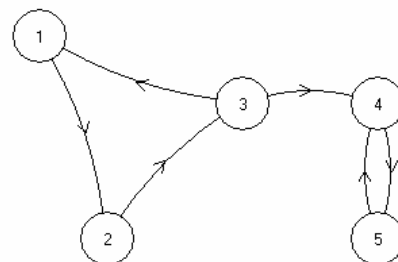
<ol style="list-style-type: none"> a. numărul n nu este divizibil cu n/2 c. numărul n este par 	<pre>for (i=2 ; i<=n/2; i++) if (n%i==0) ok=0; else ok=1; printf("%d",ok); cout<<ok;</pre> <ol style="list-style-type: none"> b. numărul n nu este prim d. numărul n este prim
---	---
4. Fie subprogramul **f** definit alăturat. În urma executării secvenței
`int x=1;f(x);`
 se vor afișa valorile:

<ol style="list-style-type: none"> a. 432 b. 234 	<pre>void f(int & b) {b=b+1;if (b<4) f(b); printf("%d",b); cout<<b; }</pre> <ol style="list-style-type: none"> c. 111 d. 444
--	--
5. Numărul maxim de muchii dintr-un graf neorientat cu 6 noduri și 4 componente conexe este:
 - a. 4
 - b. 1
 - c. 3
 - d. 2
6. Care dintre următoarele secvențe de instrucțiuni este corectă, știind că se folosește funcția cu antetul `void f(int a, int &b)`?
 - a. `a=1; b=1; f(a,b);`
 - b. `b=1; f(b,1);`
 - c. `a=1; f(a,1);`
 - d. `a=1;b=1; printf("%d",f(a,b)); | cout<<f(a,b);`

7.

Matricea drumurilor unui graf orientat este o matrice de dimensiune $n \times n$, definită astfel:

$a[i][j]=1$ dacă există cel puțin un drum de la nodul i la nodul j și, respectiv $a[i][j]=0$ dacă nu există niciun drum de la i la j . Care este matricea drumurilor pentru grafurile alăturate?



a. $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$

b. $\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

c. $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$

d. $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

8. Generarea tuturor șirurilor formate din trei elemente, fiecare element putând fi oricare număr din mulțimea $\{1, 2, 3\}$, se realizează cu ajutorul unui algoritm echivalent cu algoritmul de generare a:

- a. permutărilor b. combinațiilor c. produsului cartezian d. aranjamentelor

SUBIECTUL II (20 de puncte)

Se consideră programul pseudocod alăturat, în care $\text{mult}(n, a)$ semnifică puterea maximă la care apare numărul natural a în descompunerea în factori primi a lui n , iar $\text{max}(x, y)$ semnifică maximum numerelor naturale x și y . În plus $a \% b$ înseamnă restul împărțirii numerelor naturale a și b .

citește m, n ($m, n \in \mathbb{N}$, prime între ele)

$z \leftarrow \text{max}(\text{mult}(n, 2), \text{mult}(n, 5))$

$r \leftarrow 10 * m$

pentru $i=1, z$ execută

$r \leftarrow r \% n * 10$

dacă $r \neq 0$ atunci

 scrie „D”

altfel

 scrie „N”

1. Care este valoarea afișată dacă se citesc valorile $m=10$ și $n=3$? (5p.)
2. Dacă $m=9$, precizați o valoare pentru variabila n astfel încât să se afișeze „N”. (3p.)
3. Dacă $m=10$, atunci câte numere din intervalul $[1, 10]$ pot fi introduse ca valoare pentru n astfel încât să se afișeze „D”? (2p.)
4. Scrieți programul C/C++ corespunzător algoritmului dat, fără a scrie definiția subprogramelor. (10p.)

SUBIECTUL III (30 de puncte)

1. În fișierul text **MAT.TXT** se află $n \times m$ ($1 < n < 11, 1 < m < 21$) valori numerice întregi de cel mult nouă cifre dispuse pe mai multe linii și separate prin spații. Valorile variabilelor n și m se citesc de la tastatură. Să se scrie programul C/C++ care construiește în memorie și afișează pe ecran tabloul cu n linii și m coloane ce se formează cu elementele din fișier astfel încât ordinea tuturor elementelor din fișier să coincidă cu parcurgerea tabloului linie cu linie. (10p.)
2. a) Două subprograme **s1** și **s2** primesc ca parametri câte un șir, x , de cel mult 100 numere întregi și câte două valori p și q ($0 \leq p \leq q < 101$). Subprogramul **s1** returnează maximum elementelor șirului x situate între pozițiile p și q , inclusiv p și inclusiv q , iar subprogramul **s2** returnează prin parametrul **min** minimum elementelor șirului x situate între pozițiile p și q , inclusiv p și inclusiv q . Scrieți definițiile complete ale subprogramelor anterioare. (6p.)
b) Scrieți programul C/C++ care citește de la tastatură două tablouri unidimensionale **a** și **b** de cel mult 100 numere întregi și afișează pe ecran mesajul „DA” în cazul în care fiecare element din **a** este strict mai mic decât toate elementele din **b**, respectiv „NU”, în caz contrar. Se vor folosi apeluri utile ale subprogramelor precizate anterior. (4p.)
3. Scrieți programul C/C++ care să afișeze toate numerele impare de două cifre. Afișarea se va face pe ecran, câte două numere pe fiecare linie, numerele de pe aceeași linie fiind separate printr-un spațiu. (10p.)