

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

Varianta 35

- ♦ 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. Graful neorientat cu 8 noduri numerotate de la 1 la 8, este reprezentat cu ajutorul matricei de adiacență alăturată. Numărul minim de muchii ce trebuie adăugate pentru ca nodul 2 să fie legat prin lanțuri elementare de lungime 3 de toate nodurile grafului, este:

0	1	1	1	0	0	0	0
1	0	1	0	1	1	0	0
1	1	0	0	1	0	0	0
1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0
0	1	0	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

 - a. 4
 - b. 5
 - c. 2
 - d. 3
2. Într-o listă circulară simplu înlănțuită, **p** este adresa unui nod din listă și câmpul **next** memorează pentru fiecare nod, adresa nodului următor din listă. Pentru a număra elementele listei vom scrie secvența (variabila **q** este de același tip cu variabila **p**):
 - a. **q=p; k=1;while(q->next!=p) {k++;q=q->next;}**
 - b. **q=p; k=1;do {q=q->next;k++;} while (q==p);**
 - c. **q=p; k=1;while(q!=p) {k++;q=q->next;}**
 - d. **k=0;do {p=p->next;k++;} while (p!=NULL);**
3. Se dă un graf neorientat cu 75 de noduri numerotate de la 1 la 75, și muchiile [21,40], [30,38], [21,30], [60,75]. Atunci numărul de componente conexe ale grafului este:
 - a. 69
 - b. 71
 - c. 2
 - d. 73
4. Generarea tuturor cuvintelor de 4 litere, fiecare literă putând fi orice element din mulțimea {a,c,e,m,o,s}, se realizează cu ajutorul unui algoritm echivalent cu algoritmul de generare a:
 - a. produsului cartezian
 - b. combinațiilor
 - c. partițiilor unei mulțimi
 - d. permutărilor
5. Stabiliți care dintre următoarele expresii este adevărată dacă și numai dacă numărul întreg **x**, nu aparține intervalului închis [20,100].
 - a. **(x<=20) || (x>=100)**
 - b. **(x<20) || !(x>100)**
 - c. **(x<=19) || (x>=101)**
 - d. **(x<=19)&&(x>=101)**
6. Se consideră o stivă alocată dinamic care are cel puțin 10 elemente. Variabila **vf** memorează adresa de început a stivei și orice element al stivei memorează în câmpul **info** un număr întreg, iar în câmpul **next** adresa nodului următor.


```
while (vf && vf->info%2==0)
{
    aux=vf;
    vf=aux->next;
    delete aux;    /free (aux);
}
```

 Dacă în urma executării secvenței alăturate de program, variabila **vf** are valoarea **NULL**, atunci:
 - a. primul element memorat în stivă este par, celelalte fiind numere impare
 - b. în stivă nu s-a memorat nici un număr impar
 - c. ultimul element memorat în stivă este par celelalte elemente fiind numere impare
 - d. în stivă nu s-a memorat nici un număr par
7. În urma executării programului alăturat, se va afișa:

<pre>void f(int &a,int b) {a=a-1;b=a+1; cout<<a; /printf("%d",a); cout<<b; /printf("%d",b); }</pre>	<pre>void main() {int a=3,b=7; f(b,b); cout<<a; /printf("%d",a); cout<<b; /printf("%d",b); f(b,a); }</pre>
<ol style="list-style-type: none"> a. 673656 b. 676767 	<ol style="list-style-type: none"> c. 673767 d. 282819

8. Se consideră un graf orientat cu 6 vârfuri și arcele: $(1,4)$, $(1,5)$, $(2,3)$, $(2,4)$, $(3,4)$, $(4,3)$, $(4,6)$, $(5,4)$, $(6,4)$. Gradul interior al vârfului 4 este:
- a. 7 b. 3 c. 2 d. 5

SUBIECTUL II (20 de puncte)

Se consideră programul pseudocod alăturat:

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

1. Ce se va afișa pentru $a=10$ și $b=20$? (3p.)
2. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
3. Scrieți un algoritm echivalent cu algoritmul dat, în care să se utilizeze doar structuri repetitive cu test final. (5p.)
4. Dați un enunț de problemă care poate fi rezolvată cu acest algoritm. (2p.)

```

citește a,b
(numere naturale,  $a \leq b$ )
 $k \leftarrow 0$ 
pentru  $i \leftarrow a, b$  execută
     $nr \leftarrow 0; aux \leftarrow i$ 
    cât timp  $aux \neq 0$ 
         $nr \leftarrow nr * 10 + aux \% 10$ 
         $aux \leftarrow [aux / 10]$ 
    dacă  $nr < i$  atunci
         $k \leftarrow k + 1$ 
scrie k

```

SUBIECTUL III (30 de puncte)

1. Șirul lui Fibonacci este definit astfel: $t_0=0$, $t_1=1$, $t_n=t_{n-1}+t_{n-2}$ pentru $n \geq 2$.
Scrieți programul C/C++ care citește de la tastatură două numere naturale a, b ($1 \leq a \leq b \leq 32000$) și verifică dacă aceste numere sunt termeni consecutivi în șirul lui Fibonacci. Se va afișa pe ecran un mesaj dacă a și b îndeplinesc sau nu, condițiile din enunț.
Exemplu: pentru $a=34$, $b=55$ se va afișa pe ecran mesajul:
Numerele 34 și 55 sunt termeni consecutivi în șirul lui Fibonacci (10p.)
2. Scrieți definiția completă a unui subprogram **create** care creează în memorie un vector v cu elemente numere întregi, obținute prin parcurgerea în ordine a coloanelor unei matrice a începând de la prima până la ultima, pe fiecare coloană elementele fiind parcurse de jos în sus. Subprogramul va avea ca parametri: numerele întregi n și m ce reprezintă numărul de linii, respectiv de coloane ale matricei date, matricea a și vectorul v ce se va construi.
Exemplu: pentru $n=3$, $m=5$ și matricea :

1	4	-5	0	-7
4	7	-7	-6	3
-9	2	1	-1	-2

 se va construi vectorul $v : (-9, 4, 1, 2, 7, 4, 1, -7, -5, -1, -6, 0, -2, 3, -7)$ (10p.)
3. Subprogramul **ordonare** primește prin intermediul parametrilor $c1$, $c2$ și $c3$, trei cuvinte de maximum 30 de caractere fiecare și afișează aceste trei cuvinte în ordine lexicografică, despărțite prin câte un spațiu.
a) Scrieți definiția completă a subprogramului **ordonare**. (4p.)
b) Fișierul **cuvinte.in** conține pe fiecare linie câte trei cuvinte despărțite prin câte un spațiu. Scrieți programul C/C++ care citește din fișier, de pe fiecare rând al său câte trei cuvinte și folosind apeluri ale subprogramului **ordonare** le afișează în ordine lexicografică pe câte un rând de ecran.
Exemplu: (6p.)

Fișierul cuvinte.in conține:	Pe ecran se va afișa :
ariadna are examen	are ariadna examen
azi e soare	azi e soare
maine va ploua	maine ploua va
examenul e usor	e examenul usor
succes elevilor participanti	elevilor participanti succes