

**EXAMENUL DE BACALAUREAT – 2007**  
**Proba scrisă la INFORMATICĂ**  
**PROBA E, limbajul Pascal**

Varianta 46

- ♦ 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. Fie graful orientat  $G$  cu  $n=5$  noduri, dat prin următoarele liste de adiacență:  
 1: (2, 3), 2: (3, 4), 3: (4, 5), 4: (1, 2), 5: (4).  
 Care dintre următoarele propoziții este falsă?
  - a. există cel puțin un nod în graful  $G$  care are gradul intern egal cu cel extern
  - b. există cel puțin un drum între oricare două noduri ale grafului  $G$
  - c. graful  $G$  nu are circuite
  - d. graful  $G$  are 9 arce
2. Fie  $a$ ,  $b$  și  $c$  oricare trei numere naturale nenule. Știind că  $a$  este divizor al lui  $b$ , iar  $c$  nu este multiplu al lui  $a$ , care dintre următoarele expresii are valoare **true**?
  - a.  $\text{not}((b \bmod a \neq 0) \text{ or } (c \bmod a = 0))$
  - b.  $(b \bmod a \neq 0) \text{ or } (c \bmod a = 0)$
  - c.  $(a \bmod b = 0) \text{ and } (a \bmod c \neq 0)$
  - d.  $(b \bmod a \neq 0) \text{ and } (c \bmod a = 0)$
3. Un arbore are nodurile numerotate de la 1 la 5. Care dintre următorii vectori nu poate fi vector de tați?
  - a. 2 0 1 1 2
  - b. 4 1 1 0 2
  - c. 3 4 0 2 3
  - d. 3 1 0 1 2
4. Fie  $n$  un număr natural format din 6 cifre. Pentru a memora în variabila  $a$  cifra miilor numărului  $n$ , se folosește atribuirea :
  - a.  $a:=n \text{ div } 1000 \bmod 10$
  - b.  $a:=n \text{ div } 100 \text{ div } 10$
  - c.  $a:=n \bmod 1000 \bmod 100$
  - d.  $a:=n \text{ div } 1000 \bmod 100$
5. Pentru subprogramul alăturat, apelul `func(4)` are ca rezultat:
 

```
function func(x: integer):integer;
begin
  if x<=0
  then func := -1
  else
    func := 1 div func(x-1)-2*func(x-2)
  end;
```

- a. -1
  - b. -5
  - c. -6
  - d. 0
6. Se consideră graful neorientat dat prin matricea de adiacență alăturată.  
 Care dintre următoarele afirmații este adevărată?
 

- a. nodurile 1, 2, 4 se află în aceeași componentă conexă
  - b. graful conține 3 componente conexe și cel puțin un nod izolat
  - c. graful conține 2 componente conexe și nu are cicluri
  - d. graful conține 3 componente conexe și nu are cicluri

0	1	1	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	0	0	0
7. Se generează prin metoda **backtracking** mulțimi distincte cu elemente numere naturale nenule și cu proprietatea că suma elementelor fiecărei mulțimi este egală cu 7 astfel:  
 $\{1, 2, 4\}$ ,  $\{1, 6\}$ ,  $\{2, 5\}$ ,  $\{3, 4\}$ ,  $\{7\}$ . Folosind aceeași metodă pentru a genera mulțimi distincte cu elemente numere naturale nenule și cu proprietatea că suma elementelor fiecărei mulțimi este egală cu 9, stabiliți în ce ordine sunt generate următoarele mulțimi:  
 a)  $\{2, 3, 4\}$ ; b)  $\{3, 6\}$ ; c)  $\{2, 7\}$ ; d)  $\{1, 8\}$ .
  - a. d a b c
  - b. d a c b
  - c. a c b d
  - d. a b c d

8. Se consideră o listă liniară simplu înlănțuită, cu cel puțin două noduri. Fiecare nod reține în câmpul **info** o valoare numerică iar în câmpul **adr** adresa următorului nod din listă. Dacă **p** este adresa unui nod din listă atunci **p** și **p<sup>^</sup>.adr** conțin aceeași informație în câmpul **info** dacă și numai dacă:
- $p^{^}.adr = p$
  - $p = p^{^}.info$
  - $p^{^}.info = p^{^}.adr^{^}.info$
  - $p.info = p^{^}.adr.info$

## SUBIECTUL II (20 de puncte)

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  $x$ .

- Care este valoarea afișată pentru  $n=50324$ ? (4p.)
- Pentru  $n = \overline{31a2b}$ , unde  $a$  este cifra sutelor iar  $b$  este cifra unităților, câte perechi ordonate  $(a,b)$  de cifre există pentru ca valoarea afișată să fie 1. (3p.)
- Scrieți programul Pascal corespunzător algoritmului dat. (8p.)
- Să se scrie un program pseudocod echivalent cu cel dat folosindu-se un alt tip de structură repetitivă. (5p.)

```

citește n (nr. natural)
s1 ← 0
s2 ← 0
nr ← 0
cât timp n <> 0 execută
    dacă n % 2 = 0 atunci
        s1 ← s1 + n % 10
    altfel
        s2 ← s2 + n % 10
    n ← [n/10]
dacă s1 = s2 atunci
    nr ← 1
scrie nr

```

## SUBIECTUL III (30 de puncte)

- Din fișierul **bac.in** se citește de pe prima linie un număr natural  $n$  ( $0 < n < 5000$ ), iar de pe linia a doua se citesc  $n$  numere naturale, fiecare având cel mult patru cifre. În fișier numerele sunt separate prin câte un spațiu.
  - Se cere să se aleagă un algoritm eficient pentru determinarea valorilor distincte și să se descrie în 3-4 rânduri algoritmul justificându-se eficiența sa; (4p.)
  - Scrieți programul Pascal ce realizează prelucrarea descrisă la punctul **a**. Valorile distincte determinate se scriu, cu spațiu între ele, în fișierul **bac.out**. (6p.)
- Realizați următoarele cerințe utilizând limbajul Pascal:
  - Scrieți definiția completă a unui subprogram recursiv **sub**, care primește prin intermediul parametrului  $n$  un număr natural cu cel mult 8 cifre și returnează prin intermediul celui de-al doilea parametru **max**, cea mai mare cifră din scrierea lui  $n$ ; (4p.)
  - Scrieți un program care citește de la tastatură un număr natural **nr** cu cel mult 8 cifre și care, folosind subprogramul **sub**, afișează pe ecran cea mai mare cifră a numărului **nr** și de câte ori apare aceasta în scrierea sa. (6p.)

Exemplu, pentru **nr=26361**, se afișează: 6 2.
- Scrieți programul Pascal care citește de la tastatură un șir de cel mult 100 de caractere, format doar din literele mici ale alfabetului englez și caractere spațiu. Programul afișează pe ecran șirul codificat astfel:
  - după fiecare vocală se adaugă consoana imediat următoare (după **a** se inserează **b**, după **i** se inserează **j** ș.a.m.d.),
  - la sfârșitul șirului se adaugă grupul de litere **stop**.

Exemplu: dacă șirul inițial este „azi este proba de informatica”, după codificare se afișează: „abzij efstef propbab def i jnfoprmbatijcabstop” (10p.)