

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

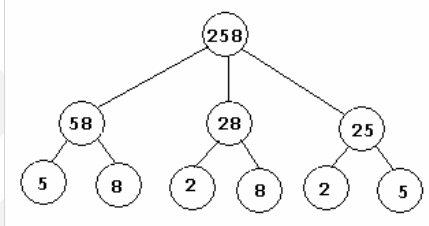
**Varianta 29**

- ♦ 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. Se consideră un arbore cu rădăcină în care orice nod care nu este rădăcină memorează un număr obținut prin ștergerea unei cifre din numărul păstrat în nodul tată (conform exemplului din figura alăturată).  
 Știind că rădăcina memorează valoarea 1234, că fiii oricărui nod sunt diferiți și că orice frunză conține o singură cifră, stabiliți câte frunze memorează cifra 1.

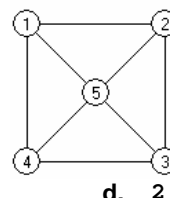


- a. 6                      b. 12                      c. 3                      d. 1
2. Variabila **P**, de tip structură, memorează în două câmpuri **x** și **y** două numere reale reprezentând coordonatele punctului **P** din plan. Condiția ca acest punct să coincidă cu unul dintre punctele (1,0) sau (0,1) este:
- a. **P.x==0 && P.y==1 && P.x==1 && P.y==0**  
 b. **P.x==0 && P.y==1 || P.x==1 && P.y==0**  
 c. **P.x==0 || P.y==1 && P.x==1 || P.y==0**  
 d. **P.x==0 || P.y==1 || P.x==1 || P.y==0**
3. Se consideră o listă simplu înlanțuită în care fiecare element memorează în câmpul **nr** un număr natural strict pozitiv și în câmpul **urm** adresa elementului următor din listă. Lista memorează, în ordine, pornind de la primul element, valorile 1,2,2,3,3,3,4,4,4,4,5,5,5,5,5. Știind că **prim** reține adresa primului element al listei iar **p** este o variabilă de același tip cu **prim**, stabiliți câte valori nule există în listă după executarea secvenței:

```
p=prim;
while(p->urm!=NULL)
    if(p->nr == p->urm->nr)
        {p->nr=0; p=p->urm->urm;}
    else p=p->urm;
```

- a. 11                      b. 5                      c. 6                      d. 7
4. Care este valoarea expresiei  $(2 \cdot 3 - 5 / 2 - (3 - 7 \cdot 2))$ ?
- a. -2                      b. 1.5                      c. -1.5                      d. 2

5. Se consideră graful neorientat din figura alăturată. Numărul maxim de muchii ce pot fi eliminate din graf astfel încât graful parțial rezultat să fie conex este:



- a. 4                      b. 5                      c. 3
6. Produsul cartezian  $\{1,2,3\} \times \{2,3\}$  este obținut cu ajutorul unui algoritm backtracking care generează perechile (1,2), (1,3), (2,2), (2,3), (3,2), (3,3).

Care este numărul perechilor obținute prin utilizarea aceluiași algoritm la generarea produsului cartezian  $\{1,2,3,4\} \times \{2,3,4\}$ ?

- a. 12                      b. 10                      c. 81                      d. 6

7. Pentru definiția subprogramului alăturat stabiliți ce se afișează la apelul `f(5,1)`.
- ```
void f(int i, int k)
{
    if(k<=4){
        printf("%d",i*k); | cout<<i*k;
        f(i-1,k+1);
    }
}
```
- a. 1248                      b. 5898                      c. 1234                      d. 5488
8. Pentru a verifica dacă toate elementele unui vector **A** sunt strict mai mici decât toate elementele unui alt vector **B** ordonat crescător, se compară pe rând fiecare element din **A** cu toate elementele din **B**. Această metodă de verificare este:
- a. incorectă                      b. corectă dar neeficientă  
c. corectă numai pentru componente întregi                      d. corectă și eficientă

### SUBIECTUL II (20 de puncte)

Se consideră programul pseudocod alăturat :

1. Ce se va afișa pentru  $x=1, y=10$ ? (5p.)
2. Câte perechi  $(x, y)$  există în intervalul  $[1;10]$  astfel încât să se afișeze valoarea 5? (3p.)
3. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
4. Scrieți programul pseudocod care să fie echivalent cu algoritmul dat și care să nu conțină nici o structură repetitivă sau recursivă. (2p.)

citește  $x, y$  (numere naturale,  $x < y$ )

```

k ← 0
cat timp x < y execută
    x ← x+1
    y ← y-1
    k ← k+1
    ■
dacă x=y atunci
    scrie 2*k+1
altfel
    scrie 2*k
    ■

```

### SUBIECTUL III (30 de puncte)

1. Scrieți programul C/C++ care, pentru o valoare  $n$  (număr natural,  $1 \leq n \leq 100$ ) citită de la tastatură, scrie în fișierul `bac.txt`, separate prin spațiu, primele  $n$  numere strict pozitive divizibile cu 3. De exemplu, pentru  $n=4$ , conținutul fișierului este : 3 6 9 12 (10p.)
2. Scrieți un program C/C++ care verifică dacă un număr natural  $n$  cu cel mult 9 cifre, citit de la tastatură, este un număr „fierăstrău”. Definim număr „fierăstrău” un număr cu cel puțin 3 cifre care este format numai cu ajutorul a două cifre distincte  $a$  și  $b$  și are una dintre formele aba, ababa, abababa, ababababa, cu  $a > b$ . Programul afișează **DA** în caz afirmativ și **NU** în caz contrar. Exemple: 9393939, 515 sunt numere fierăstrău iar 9354539 nu este număr fierăstrău. (10p.)
3. Subprogramul `maxim` primește prin intermediul parametrului **A** un vector cu cel mult 100 de numere reale și prin intermediul parametrului  $n$  numărul de elemente din vector. El returnează valoarea celui mai mare element din vector, șterge toate aparițiile acestui element din vector și modifică corespunzător valoarea lui  $n$ .
  - a) Scrieți numai antetul subprogramului `maxim`. (3p.)
  - b) Scrieți programul C/C++ care citește de la tastatură un număr natural  $n$  ( $1 \leq n \leq 100$ ) și apoi cele  $n$  componente reale ale unui vector **v** și afișează, în ordine strict descrescătoare, separate prin spațiu, numerele care apar în vector. Se vor folosi apeluri ale subprogramului `maxim` definit la punctul a). De exemplu, pentru vectorul 2,3,1,1,6,2,7,6,1 se afișează 7 6 3 2 1. (7p.)