

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

Varianta 10

- ♦ 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 arborele  $G=(V,E)$  în care mulțimea vârfurilor este  $V=\{1,2,3,4,5,6,7,8,9,10\}$ , iar mulțimea muchiilor este  $E=\{[1,3],[1,4],[2,1],[2,5],[3,7],[4,8],[4,9],[5,6],[9,10]\}$ . Considerând vârful 1 rădăcina arborelui, vectorul de țapi corespunzător arborelui  $G$  este:
  - a.  $T=(0,1,1,3,1,5,3,4,9,4)$
  - b.  $T=(0,1,1,1,3,5,3,4,4,4)$
  - c.  $T=(0,1,1,1,5,2,4,3,4,9)$
  - d.  $T=(0,1,1,1,2,5,3,4,4,9)$
2. Într-o listă simplu înlanțuită alocată dinamic fiecare element reține în câmpul **nr** un număr întreg și în câmpul **urm** adresa următorului element din listă. Știind că variabila **p** conține adresa primului element din listă și variabila **t** este de același tip cu variabila **p**, stabiliți care dintre următoarele secvențe eliberează întreaga zonă de memorie ocupată de elementele listei:
  - a. 

```
while(p!=NULL)
{ t=p; p=p->urm;
  delete p; / free(p);
}
```
  - b. 

```
while(p!=NULL)
{ t=p; p=p->urm;
  delete t; / free(t);
}
```
  - c. 

```
while(p!=NULL)
{ t=p; t=t->urm;
  delete t; / free(t);
}
```
  - d. 

```
delete p; / free(p);
```
3. Fie  $G$  un graf neorientat conex cu 100 de vârfuri și 2007 muchii. Numărul maxim de muchii ce pot elimina din  $G$  astfel încât acesta să rămână conex este:
  - a. 1907
  - b. 1007
  - c. 1237
  - d. 1908
4. Considerăm  $n$  copii și  $p$  tricouri pe care sunt imprimate numerele de la 1 la  $p$  ( $n, p \in \mathbb{N}$ ,  $1 \leq p \leq n$ ). Algoritmul care să genereze și să afișeze toate modurile în care pot fi împărțite cele  $p$  tricouri celor  $n$  copii este echivalent cu algoritmul folosit pentru generarea:
  - a. aranjamentelor
  - b. permutărilor
  - c. produsului cartezian
  - d. combinațiilor
5. Subprogramul **nrap(s,c)** returnează numărul aparițiilor caracterului  $c$  în șirul  $s$ . Dacă  $a$  și  $b$  sunt două șiruri de caractere formate doar din litere mici ale alfabetului englez, cu ce expresie trebuie înlocuite punctele de suspensie din secvența următoare astfel încât, după executarea ei, variabila de tip întreg  $k$  să aibă valoarea 1 dacă mulțimea formată din literele distincte ale șirului  $a$  este egală cu mulțimea formată din literele distincte ale șirului  $b$  sau valoarea 0 în caz contrar?
 

```
k=1;
for(c='a';c<='z';c++)
  if(.....) k=0;
```

  - a.  $(nrap(a,c)>0 \ \&\& \ nrap(b,c)==0) \ || \ (nrap(b,c)>0 \ \&\& \ nrap(a,c)==0)$
  - b.  $nrap(a,c)>0 \ \&\& \ nrap(b,c)==0$
  - c.  $nrap(a,c)==1 \ \&\& \ nrap(b,c)==0$
  - d.  $nrap(a,c)>0 \ \&\& \ nrap(b,c)==0 \ \&\& \ nrap(b,c)>0 \ \&\& \ nrap(a,c)==0$
6. Fie subprogramul **f** definit alăturat și  $a$  o variabilă de tip întreg. Dacă în urma apelului  $f(f(a))$  valoarea returnată de funcție a fost 55, atunci valoarea variabilei  $a$  a fost:
 

```
int f(int n)
{ if (n==0) return 0;
  else return n+f(n-1);
}
```

  - a. 10
  - b. 4
  - c. 3
  - d. 5

7. Fie  $G=(V,E)$  un graf orientat în care mulțimea nodurilor este  $V=\{1,2,\dots,10\}$ , iar mulțimea arcelor este  $E=\{(i,j)\in V\times V \mid i\neq j \text{ și } j \bmod i=0\}$  (prin  $a \bmod b$  am notat restul împărțirii lui  $a$  la  $b$ ). Stabiliți care dintre următoarele afirmații este adevărată:
- Pentru oricare pereche de noduri  $i$  și  $j$  ( $i\neq j$ ) există cel puțin un drum de la  $i$  la  $j$  și cel puțin un drum de la  $j$  la  $i$
  - pentru orice nod al grafului  $G$  suma dintre gradul interior și gradul exterior este nenulă
  - toate vârfurile grafului  $G$  au gradul interior egal cu gradul exterior
  - graful  $G$  conține circuite
8. Fie  $v$  un tablou unidimensional format din  $n$  componente întregi, numerotate de la 1 la  $n$ , iar  $p$  și  $k$  două variabile de tip întreg. În care dintre următoarele secvențe variabilei  $p$  i se atribuie prima valoare strict pozitivă din tabloul  $v$ , dacă există o astfel de valoare, sau 0 în caz contrar?
- $p=0;k=1;$   
 $\text{while}(v[k]>0) \ k++;$   
 $p=v[k];$
  - $p=0;k=n;$   
 $\text{while}(v[k]<=0) \ k--;$   
 $p=v[k];$
  - $p=0;$   
 $\text{for}(k=1;k\leq n;k++)$   
 $\text{if}((v[k]>0)\&\&(p==0)) \ p=v[k];$
  - $p=0;$   
 $\text{for}(k=1;k\leq n;k++)$   
 $\text{if}(v[k]>0) \ p=v[k];$

## SUBIECTUL II (20 de puncte)

Se consideră programul pseudocod alăturat:

- Ce se va afișa pentru  $n=6$ ? (4 p.)
- Care este cea mai mare valoare ce poate fi introdusă pentru  $n$  astfel încât algoritmul să afișeze doar tripletele 3,4,5 și 6,8,10? (3 p.)
- Scrieți programul C/C++ corespunzător algoritmului dat. (10 p.)
- Scrieți un program C/C++ echivalent cu algoritmul dat care să folosească doar două structuri repetitive. (3p.)

```

citește n {n∈N}
pentru i←1,n-2 execută
    pentru j←i+1,n-1 execută
        pentru k←j+1,n execută
            dacă k*k=i*i+j*j atunci
                scrie i,j,k

```

## SUBIECTUL III (30 de puncte)

- Scrieți un program care construiește în memorie o matrice pătratică de dimensiune  $n$  ( $1\leq n\leq 30$ ) care să conțină pe prima linie, în ordine crescătoare, numerele de la 1 la  $n$ , pe a doua linie, în ordine descrescătoare, numerele de la  $2*n$  la  $n+1$ , pe a treia linie, în ordine crescătoare, numerele de la  $2*n+1$  la  $3*n$ , etc. Valoarea lui  $n$  se citește de la tastatură, iar matricea se va afișa pe ecran, câte o linie a matricei pe câte o linie a ecranului, cu spații între elementele fiecărei linii, după cum se poate observa în exemplu.  
De exemplu, pentru  $n=3$  se construiește în memorie și se afișează matricea:
 

1	2	3
6	5	4
7	8	9

 (10p.)
- Scrieți un subprogram `vocal` care are ca parametru un șir de caractere  $s$  format din cel mult 50 de litere ale alfabetului englez și returnează numărul de vocale ( $a, e, i, o, u$ ) din șirul respectiv. (10p.)
- Fișierul text `BAC.IN` are următoarea structură: pe prima linie se găsește un număr natural  $n$  ( $2\leq n\leq 1.000.000$ ), iar pe a doua linie se găsesc  $n$  numere întregi formate din cel mult 9 cifre și despărțite prin spațiu. Știind că fișierul `BAC.IN` conține întotdeauna cel puțin două numere distincte printre cele  $n$  aflate pe cea de-a doua linie, scrieți un program care să creeze mai întâi fișierul text `BAC.OUT` și apoi să scrie în el, pe prima linie, separate printr-un spațiu și în ordine descrescătoare, cele mai mari două valori distincte din fișierul `BAC.IN`. Alegeți o metodă eficientă de rezolvare atât ca timp de executare, cât și ca gestionare a memoriei. (10p.)

Exemplu:

<b>BAC.IN</b> 10 5 11 7 3 8 -2 11 11 -7 5	<b>BAC.OUT</b> 11 8
---	------------------------