

**Examenul de bacalaureat național 2016**  
**Proba E. d)**  
**Informatică**  
**Limbajul C/C++**

Varianta 4

*Filiera teoretică, profilul real, specializările: matematică-informatică  
matematică-informatică intensiv informatică*  
*Filiera vocațională, profilul militar, specializarea matematică-informatică*

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- În rezolvările cerute, 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).
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.

**SUBIECTUL I** **(30 de puncte)**

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Variabilele  $x$ ,  $y$  și  $z$  sunt de tip întreg și memorează numere naturale din intervalul  $[1, 10^3]$ . Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă valoarea variabilei  $x$  este strict mai mare decât valoarea oricăreia dintre variabilele  $y$  și  $z$ . **(4p.)**

- a.  $x*y > y*z \ \&\& \ x*z > y*z$  b.  $x*z > x*y \ \&\& \ y*z > y*x$   
c.  $y*z > x*z \ \&\& \ y*x > z*x$  d.  $y*z > y*x \ \&\& \ y*z > z*x$

2. Se consideră algoritmul alăturat, reprezentat în pseudocod.

S-a notat cu  $a \% b$  restul împărțirii numărului natural  $a$  la numărul natural nenul  $b$  și cu  $[c]$  partea întreagă a numărului real  $c$ .

- a) Scrieți valoarea afișată în urma executării algoritmului dacă se citește numărul 81112337. **(6p.)**
- b) Scrieți numărul de valori din intervalul  $[10000, 99999]$  care să înceapă cu cifra 1, care pot fi citite pentru variabila  $n$ , astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, să se afișeze valoarea 4. **(4p.)**

```
citește n  
(număr natural nenul)  
k ← 1  
m ← 1  
cât timp n > 9 execută  
┌ dacă n % 10 = [n / 10] % 10 atunci  
│   k ← k + 1  
│   ┌ dacă k > m atunci  
│   │   m ← k  
│   └─┘  
│ altfel  
│   k ← 1  
└─┘  
n ← [n / 10]  
scrie m
```

- c) Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind structura **cât timp...execută** cu o structură repetitivă de alt tip. **(6p.)**
- d) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**

**SUBIECTUL al II-lea**

**(30 de puncte)**

**Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

- Variabilele  $i$  și  $j$  sunt de tip întreg, iar variabila  $m$  memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 0 la 4, cu elemente numere întregi.  
O expresie C/C++ a cărei valoare este egală cu produsul dintre primul element de pe linia  $i$  și ultimul element de pe coloana  $j$  din acest tablou este: **(4p.)**
  - $m[0,i]*m[j,4]$
  - $m(0,i)*m(j,4)$
  - $m(i)(0)*m(4)(j)$
  - $m[i][0]*m[4][j]$
- Un graf orientat are 12 arce, 3 componente tare conexe, iar fiecare vârf al său are gradul interior un număr nenul. Numărul maxim de noduri pe care le poate avea graful este: **(4p.)**
  - 12
  - 11
  - 9
  - 8

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

- Un arbore cu 8 noduri, numerotate de la 1 la 8, este reprezentat prin următorul vector de „tați” (3, 0, 2, 5, 2, 5, 1, 5). Determinați cel mai lung lanț elementar care are o extremitate în rădăcină și enumerați nodurile sale, în ordinea apariției în acesta. **(6p.)**

- Variabilele  $i$  și  $s$  sunt întregi, iar variabila  $p$  memorează, pentru fiecare dintre cele 20 de zone de parcare ale unui oraș, numărul de locuri închiriate pe parcursul orei curente, precum și prețul practicat de zona respectivă pentru închirierea unui loc pentru o oră.  

```
struct parcare  
{ int nrLocuriInchiriate;  
  int pretOra;  
} p[20];
```

Fără a utiliza alte variabile decât cele menționate, scrieți o secvență de instrucțiuni în urma executării căreia variabila  $s$  să memoreze suma totală obținută în urma închirierii locurilor de parcare din oraș pe parcursul orei curente. **(6p.)**

- Scrieți un program C/C++ care citește de la tastatură, în această ordine, un număr natural  $n$ , apoi  $n$  cuvinte, separate prin Enter, urmate de un număr natural  $k$ . Numerele citite sunt din intervalul  $[1, 20]$ , iar fiecare cuvânt este format din cel mult 20 de caractere, numai litere mici ale alfabetului englez. Cel puțin unul dintre cuvinte are  $k$  litere. Programul construiește în memorie, apoi afișează pe ecran, un șir care să cuprindă cuvintele citite, separate prin câte un spațiu, astfel încât toate cele care au  $k$  litere să ocupe primele poziții, iar celelalte să se regăsească în continuarea acestora, într-o ordine oarecare.

**Exemplu:** dacă se citesc, în această ordine, datele alăturate, unul dintre șirurile obținute poate fi:  
`am un de cires mai`

**(10p.)**

```
5  
am  
un  
cires  
de  
mai  
2
```

**SUBIECTUL al III-lea**

**(30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Subprogramul `f` este definit alăturat. Indicați cea mai mare valoare din intervalul  $[20,25]$  pe care o poate avea variabila întreagă `x`, astfel încât, în urma apelului de mai jos, să se afișeze numerele 2 4 16.  
`f(x,1);` **(4p.)**

```
void f(int n, int p)
{
    if(n>0)
    {
        if(n%2==1)
            cout<<p<<" "; | printf(" ",p);
        f(n/2, p*2);
    }
}
```

- a. 21                      b. 22                      c. 23                      d. 24

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Utilizând metoda backtracking se generează, în ordine lexicografică, toate șirurile de câte 6 cifre din mulțimea  $\{0,1\}$  cu proprietatea că au cel mult două cifre cu valori egale pe poziții consecutive. Primele 5 soluții generate sunt, în această ordine: 001001, 001010, 001011, 001100, 001101. Scrieți a 7-a și a 8-a soluție, în ordinea generării acestora. **(6p.)**
3. Subprogramul `minDivPrim` are un singur parametru, `n`, prin care primește un număr natural ( $n \in [2, 10^9]$ ). Subprogramul returnează cel mai mic număr natural care are aceiași divizori primi ca `n`.  
Scrieți definiția completă a subprogramului.  
**Exemplu:** dacă `n=75`, subprogramul returnează numărul 15, iar dacă `n=7`, subprogramul returnează numărul 7. **(10p.)**
4. Se consideră șirul 1, -1, 2 ... definit astfel:  $f_1=1$ ,  $f_2=-1$ , iar  $f_n=1-2 \cdot f_{n-1}-f_{n-2}$ , dacă  $n \geq 3$  (unde  $n$  este un număr natural).  
Se citește de la tastatură un număr natural, `n` ( $n \in [1, 10^9]$ ), și se cere să se scrie în fișierul text `bac.out`, separați prin câte un spațiu, primii `n` termeni ai șirului, în ordine inversă apariției lor în acesta.  
Pentru determinarea și afișarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al spațiului de memorie și al timpului de executare.  
**Exemplu:** dacă `n=3`, fișierul conține valorile  
2 -1 1  
a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**  
b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**