

**Examenul de bacalaureat național 2019**

**Proba E. d)**

**Informatică**

**Limbajul C/C++**

**Varianta 2**

**Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică**  
**Filieră vocațională, profil militar, specializare matematică-informatică**

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizati în rezolvări 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). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

**SUBIECTUL I**

**(20 de puncte)**

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. O expresie C/C++ care are valoarea 0 este:

a.  $'m' < 'n'$

b.  $'m' == 'M'$

c.  $'M' == 'm' + 'N' - 'n'$

d.  $'N' == 'M' + 1$

2. Subprogramul **f** este definit alăturat.

Indicați ce se afișează în urma apelului de mai jos.

**f(75, 30);**

```
void f(int i, int j)
{ cout<<i<<" " | printf("%d ",i);
  if(i!=j)
    { if(i<j){ i=i+j; j=i-j; i=i-j; }
      f(i-j, j);
    }
}
```

a. 75 30 45

b. 75 45 15

c. 75 45 15 15

d. 75 30 45 15

3. Utilizând metoda backtracking, se generează toate torturile formate din trei straturi de tipuri distincte de cremă din mulțimea **{caramel, ciocolată, frișcă, nuci, vanilie}**. Ultimul strat este de frișcă sau de vanilie, iar aceste tipuri de cremă nu pot apărea pe primele două straturi. Două torturi cu straturi din aceleași tipuri de cremă sunt diferite dacă acestea sunt dispuse în altă ordine. Primele patru soluții generate sunt, în această ordine: **(caramel, ciocolată, frișcă)**, **(caramel, ciocolată, vanilie)**, **(caramel, nuci, frișcă)**, **(caramel, nuci, vanilie)**. A cincea soluție este:

a. **(ciocolată, caramel, frișcă)**

b. **(ciocolată, nuci, frișcă)**

c. **(nuci, caramel, vanilie)**

d. **(nuci, ciocolată, vanilie)**

4. Numărul de noduri ale unui arbore cu 4 muchii este:

a. 2

b. 3

c. 4

d. 5

5. Valorile care pot reprezenta gradele nodurilor unui graf neorientat, cu 6 noduri, sunt:

a. 1, 0, 0, 2, 2, 2

b. 2, 2, 3, 4, 0, 3

c. 2, 2, 5, 5, 0, 1

d. 6, 5, 4, 3, 2, 1

### **SUBIECTUL al II-lea**

**(40 de puncte)**

**1. Algoritmul alăturat este reprezentat în pseudocod.**

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

- Scrieți valoarea afișată dacă se citește numărul **27102**. (6p.)
- Scrieți trei numere distincte din intervalul **[10, 9999]** care pot fi citite, astfel încât în urma executării algoritmului, pentru fiecare dintre acestea, valoarea afișată să fie **11**. (6p.)
- Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind prima structură **cât timp...execută** cu o structură de tip **pentru...execută**. (6p.)

2. Fiind date două siruri de caractere **a** și **b**, îl numim pe **a** **sufix** al lui **b** dacă **a** este egal cu **b** sau dacă **b** se poate obține din **a** prin alipirea la stânga a unor noi caractere.

Variabilele **a** și **b** pot memora câte un sir cu cel mult **20** de caractere. Scrieți o secvență de instrucțiuni în urma executării căreia variabila **a** să memoreze un sufix al lui **b** format din trei caractere, sau sirul **nedeterminat**, dacă nu există un astfel de sufix.

**Exemplu:** dacă **b** memorează sirul **centaur**, atunci **a** memorează sirul **aur**, iar dacă **b** memorează sirul **au**, atunci **a** memorează sirul **nedeterminat**. (6p.)

3. În declarările alăturate, variabila **p** memorează coordonatele unui punct în sistemul de coordonate **xoy**, iar variabila **c** memorează datele caracteristice ale unui cerc: lungimea razei și coordonatele centrului său. Scrieți o expresie care are valoarea **1** dacă și numai dacă punctul corespunzător variabilei **p** coincide cu centrul cercului corespunzător variabilei **c**. (6p.)

```

citește n
    (număr natural)
m←0; x←1
cât timp x≤9 execută
    cp←n
    cât timp cp≠0 execută
        dacă cp%10=x atunci
            m←m*10+x
        ■
    cp←[cp/10]
    ■
    x←x+1
■
scrie m

```

```

struct punct
{ int x,y;
}p;
struct cerc
{ struct punct centru;
  float raza;
}c;

```

### **SUBIECTUL al III-lea**

**(30 de puncte)**

1. Subprogramul **MaxImp** are doi parametri, **a** și **b**, prin care primește câte un număr natural ( $2 \leq a < b \leq 400$ ). Subprogramul returnează cel mai mare număr natural din intervalul **[a,b]** pentru care produsul divizorilor săi impari pozitivi este strict mai mare decât el însuși sau 0, dacă nu există niciun astfel de număr. Scrieți definiția completă a subprogramului.

**Exemplu:** dacă **a=14** și **b=19**, atunci subprogramul returnează **18** ( $1 \cdot 3 \cdot 9 = 27 > 18$ ). (10p.)

2. Numim **pătrat** de dimensiune **m** al unui tablou bidimensional tabloul obținut din acesta păstrând doar elementele aflate pe primele **m** linii și pe primele **m** coloane ale sale.

Scrieți un program C/C++ care citește de la tastatură un număr natural, **n** ( $n \in [2, 20]$ ), apoi elementele unui tablou bidimensional cu **n** linii și **n** coloane, numere naturale din intervalul  $[0, 10^4]$ . Programul determină un pătrat de dimensiune maximă al tabloului citit, cu toate elementele egale, și afișează pe ecran valoarea acestei dimensiuni.

**Exemplu:** pentru **n=5** și tabloul alăturat, se afișează pe ecran 3. (10p.)

2	2	2	2	2
2	2	2	2	8
2	2	2	2	2
2	1	2	8	7
3	5	2	1	2

3. Fișierul **bac.txt** conține un sir de cel puțin două și cel mult  $10^6$  numere naturale din intervalul  $[0, 10^3]$ , separate prin câte un spațiu. Sirul are cel puțin un termen par și cel puțin un termen impar. Se cere să se afișeze pe ecran termenii sirului, separați prin câte un spațiu, astfel încât toti cei impari să apară înaintea tuturor celor pari, și atât subșirul format din cei impari, cât și subșirul format din cei pari, să fie în ordine crescătoare, ca în exemplu. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.

**Exemplu:** dacă fișierul conține numerele **12 2 3 1 2 5**

se afișează pe ecran: **1 3 5 2 2 12**

a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)

b) Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)