

**Examenul de bacalaureat național 2017**  
**Proba E. d)**  
**Informatică**  
**Limbajul C/C++**

Simulare

*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. Expresia C/C++ alăturată are valoarea: (4p.) | 17/3/2%17
- a. 0                                      b. 2                                      c. 10                                      d. 17

2. Se consideră algoritmul alăturat, reprezentat în pseudocod.

S-a notat cu **a%b** restul împărțirii numărului natural **a** la numărul natural nenul **b**.

- a) Scrieți ce se afișează dacă se citesc, în această ordine, numerele 15, 3 și 4. (6p.)
- b) Scrieți două seturi distincte de date de intrare astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea să se afișeze valoarea 0. (4p.)
- c) Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat structura **cât timp...execută** cu o structură repetitivă de tip **pentru...execută**. (6p.)
- d) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

```
citește n, a, b
(numere naturale nenule, a ≤ n, b ≤ n)
ok ← 0
x ← 1
cât timp x ≤ n execută
    dacă x%a = 0 și x%b ≠ 0 sau
        x%a ≠ 0 și x%b = 0 atunci
        scrie x, ' '
    ok ← 1
    x ← x + 1
dacă ok = 0 atunci
    scrie 0
```

**SUBIECTUL al II-lea**

**(30 de puncte)**

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

1. Variabila **s** din secvența următoare permite memorarea unui șir de cel mult 20 de caractere.

```
strcpy(s, "tezauriza");  
cout<<strstr(s, "za"); | printf("%s", strstr(s, "za"));
```

În urma executării secvenței, se afișează pe ecran:

**(4p.)**

- a. 3                      b. 2                      c. za                      d. zauriza
2. Matricea de adiacență a unui graf neorientat cu 7 noduri are 10 elemente nenule. Numărul maxim de componente conexe ale grafului este:
- a. 2                      b. 3                      c. 4                      d. 5

**(4p.)**

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

3. Se consideră arborele cu 8 noduri, numerotate de la 1 la 8, cu rădăcina 5 și muchiile [1, 5], [2, 7], [3, 7], [3, 8], [4, 5], [5, 7], [6, 7]. Enumerați nodurile care sunt descendenți direcți („fii”) ai nodului 7.

**(6p.)**

4. Variabila **m** memorează simultan, pentru fiecare dintre cele 20 de mașini oferite spre închiriere, următoarele date: anul fabricației mașinii (număr natural) și tipul de carburant al acesteia (șir de maximum 50 de caractere).

Știind că expresiile C/C++ de mai jos au ca valori anul fabricației celei de a treia mașini, respectiv tipul de carburant al acesteia, scrieți definiția unei structuri cu eticheta **masina**, care permite memorarea datelor despre o mașină, și declarați corespunzător variabila **m**.

**m[2].an**

**m[2].carburant**

**(6p.)**

5. Scrieți un program C/C++ care citește de la tastatură un număr natural **n** ( $n \in [2, 20]$ ) și construiește în memorie un tablou bidimensional cu **n** linii și **n** coloane în care:

- ultima coloană conține numerele naturale din intervalul [1, **n**], în ordine strict descrescătoare;

- toate elementele primei linii au valoarea **n**;

- oricare alt element este obținut prin însumarea celor două elemente vecine cu el, unul aflat pe coloana din dreapta, pe aceeași linie cu el, iar celălalt pe aceeași coloană cu el, dar pe linia anterioară, ca în exemplu.

Programul afișează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

**Exemplu:** pentru **n=4** pe ecran se afișează:

```
4   4   4   4  
15  11  7   3  
35  20  9   2  
65  30 10   1
```

**(10p.)**

**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 ce se afișează în urma apelului de mai jos.  
**f(6);**

**(4p.)**

```
void f(int x)
{ cout<<x; | printf("%d",x);
  if(x>3)
  { f(x-2);
    cout<<x; | printf("%d",x);
  }
}
```

- a. 642                      b. 6446                      c. 64246                      d. 642246

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

2. Se utilizează metoda backtracking pentru a obține toate buchetele formate din câte trei tipuri de flori de primăvară din mulțimea {**brândușă, iasomie, lalea, liliac, margaretă**}, astfel încât iasomia și liliacul nu vor fi plasate în același buchet. Știind că în cadrul unui buchet nu contează ordinea de așezare a florilor, primele patru soluții obținute sunt, în această ordine: (**brândușă, iasomie, lalea**), (**brândușă, iasomie, margaretă**), (**brândușă, lalea, liliac**), (**brândușă, lalea, margaretă**). Scrieți cea de a cincea și cea de a șasea soluție, în ordinea obținerii lor. **(6p.)**

3. Subprogramul **ranguri** are un singur parametru, **n**, prin care primește un număr natural ( $n \in [0, 10^9]$ ). Subprogramul returnează numărul de cifre ale lui **n** care sunt egale cu pozițiile pe care le ocupă în scrierea acestuia. Pozițiile sunt numerotate de la dreapta la stânga, iar cifra unităților ocupă poziția 0. Scrieți definiția completă a subprogramului.

**Exemplu:** dacă **n=6594270**, subprogramul returnează numărul 4.

**(10p.)**

4. Fișierul **bac.in** conține pe prima linie două numere naturale din intervalul  $[2, 10^4]$ , **m** și **n**, iar pe fiecare dintre următoarele două linii câte un șir de **m**, respectiv **n** numere naturale din intervalul  $[0, 10^9]$ , ordonate **strict crescător**. Numerele aflate pe aceeași linie a fișierului sunt separate prin câte un spațiu.

Se cere să se afișeze pe ecran, în ordine **strict descrescătoare**, numerele pare care apar în cel puțin unul dintre cele două șiruri. Numerele afișate sunt separate prin câte un spațiu, iar dacă nu există niciun astfel de număr, se afișează pe ecran mesajul **nu exista**.

Pentru determinarea numerelor cerute se va utiliza un algoritm eficient din punctul de vedere al timpului de executare.

**Exemplu:** dacă fișierul conține numerele

5 6

1 4 8 9 10

2 4 10 12 15 18

se afișează pe ecran

18 12 10 8 4 2

a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia.

**(2p.)**

b) Scrieți programul C/C++ corespunzător algoritmului descris.

**(8p.)**