

**Examenul de bacalaureat național 2016**  
**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 efectiv de lucru 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. Indicați expresia C/C++ care are valoarea 1 dacă și numai dacă numărul memorat în variabila întregă  $x$  aparține mulțimii  $\{2015, 2016, 2017\}$ . (4p.)
- a.  $x==2015 \ \&\& \ !(x<2016 \ || \ x<2017)$       b.  $x==2015 \ \&\& \ !(x<2016 \ \&\& \ x<2017)$   
c.  $x==2015 \ || \ !(x<2016 \ || \ x>2017)$       d.  $x==2015 \ || \ !(x<2016 \ \&\& \ x>2017)$

**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 valoarea afișată dacă se citesc, în această ordine, numerele 29678, 125, 1233345, 2, 78664, 0. (6p.)
- b) Scrieți un șir de date de intrare, format din numere naturale, astfel încât, în urma executării algoritmului, să se afișeze valoarea 2016. (4p.)
- c) Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind a doua structură **repetă ...până când** cu o structură repetitivă de alt tip. (6p.)
- d) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

```
m ← 0
repetă
    citește x
    (număr natural)
    y ← x
    repetă
        c ← y % 10
        y ← [y / 10]
    până când c ≤ y % 10
    dacă y ≠ 0 atunci
        m ← m * 10 + c
    ■
până când x = 0
scrie m
```

**SUBIECTUL al II-lea**

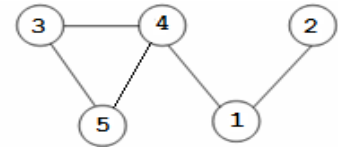
**(30 de puncte)**

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

1. Se consideră arborele cu 7 noduri, numerotate de la 1 la 7, reprezentat prin vectorul de „tați” (5, 4, 5, 3, 0, 3, 1). Mulțimea tuturor „frunzelor” arborelui este: **(4p.)**

- a. {2}                                      b. {1, 4}                                      c. {3, 5}                                      d. {2, 6, 7}

2. Se consideră graful neorientat cu cinci noduri, reprezentat alăturat. Numărul minim de muchii ce trebuie adăugate astfel încât, în graful obținut, între oricare două noduri distincte să existe cel puțin un lanț elementar de lungime 3, este: **(4p.)**



- a. 1    b. 2    c. 3    d. 4

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

3. Se consideră declararea alăturată, în care variabilele `t1` și `t2` memorează baza și înălțimea a două triunghiuri.

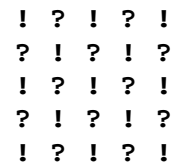
```
struct triunghi  
{ int baza, inaltime;  
  } t1, t2;
```

Scrieți o secvență de instrucțiuni C/C++ care afișează pe ecran mesajul **aceeasi arie**, dacă cele două triunghiuri au arie egală, sau mesajul **arii diferite**, în caz contrar. **(6p.)**

4. Variabilele `i` și `j` din secvența de instrucțiuni de mai jos sunt de tip întreg, iar variabila `a` memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 0 la 4, cu elemente de tip `char`. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila `a` să aibă elementele din figura de mai jos.

```
for(i=0; i<5; i++)  
  for(j=0; j<5; j++)  
    .....
```

**(6p.)**



5. Două fraze se numesc **anagrame** dacă literele lor sunt identice, dispuse eventual în altă ordine.

Scrieți un program C/C++ care citește de la tastatură două fraze, separate prin Enter, fiecare fiind formată din cel mult 100 de caractere. Cuvintele din fiecare frază conțin numai litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Programul afișează pe ecran mesajul **DA**, dacă cele două fraze sunt anagrame, sau mesajul **NU**, în caz contrar.

**Exemple:** dacă se citesc, în această ordine, frazele:

`quid est veritas`  
`est vir qui adest`

pe ecran se afișează mesajul **DA**  
iar dacă se citesc, în această ordine, frazele:

`lycophron ptolemaios`  
`chronophyl apo melitos`

pe ecran se afișează mesajul **NU**.

**(10p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

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

1. Se consideră subprogramul `f`, definit alăturat. Indicați numărul valorilor întregi afișate în urma apelului de mai jos.  
`f(2024,2);`

**(4p.)**

```
void f(int x, int nr)
{
    if(x*nr>0)
    {
        if(x%4==0)
            cout<<x<<' '; | printf("%d ",x);
        f(x-4,nr-1);
    }
    cout<<x<<' '; | printf("%d ",x);
}
```

a. 3

b. 4

c. 5

d. 6

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

2. Utilizând metoda backtracking se generează, în ordine lexicografică, toate șirurile de 4 litere distincte din mulțimea  $\{s, i, m, u, l, a, r, e\}$ , în care nu există două vocale sau două consoane alăturate. Primele 5 șiruri generate sunt, în această ordine: **alem, aler, ales, alim, alir**. Scrieți, în ordinea obținerii lor, următoarele două șiruri generate imediat după șirul **ruse**. **(6p.)**

3. Un număr natural nenul se numește **sPar** dacă atât el, cât și suma divizorilor săi proprii (divizori diferiți de 1 și de el însuși), sunt numere pare.

Subprogramul `sPar` are un singur parametru, `n`, prin care primește un număr natural ( $n \in [1, 10^9]$ ). Subprogramul returnează cel mai mic număr `sPar`, strict mai mare decât `n`.

Scrieți definiția completă a subprogramului.

**Exemplu:** dacă `n=95`, atunci subprogramul returnează **98** (atât **98**, cât și  $72=2+7+14+49$ , sunt numere pare). **(10p.)**

4. Un număr `x` se numește **sufix** al unui număr `y` dacă `y` se poate obține din `x` prin alipirea, la stânga sa, a cel puțin unei alte cifre.

Fișierul `bac.in` conține un șir de cel mult  $10^9$  numere naturale din intervalul  $[0, 10^9]$ . Numerele din șir sunt separate prin câte un spațiu.

Se cere să se afișeze pe ecran, în ordine strict crescătoare, toate numerele din intervalul  $[100, 999]$  care sunt termeni ai șirului aflat în fișier și sunt sufixe pentru cel puțin un alt termen al aceluiași șir. Numerele sunt afișate câte unul pe linie, iar dacă în șir nu există astfel de numere, se afișează pe ecran mesajul **Nu exista**. Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al timpului de executare.

**Exemplu:** dacă fișierul `bac.in` are conținutul

15502 49 54321 6149 76149 123 123 502 4321 321 321

atunci pe ecran se afișează numerele

321

502

a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**

b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**