

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

Simulare

Filiera teoretică, profilul real, specializarea științe ale naturii

- 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. 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
```

(30 de puncte)

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

1. Variabilele **x** și **y** sunt de tip întreg și memorează câte un număr natural nenul. Indicați expresia C/C++ care are valoarea **1** dacă și numai dacă valorile variabilelor **x** și **y** sunt consecutive. (4p.)

- a. $\text{abs}(x+y)$ b. $\text{abs}(x-y)$ c. $\text{abs}(x/y)$ d. $\text{abs}(x\%y)$

2. Variabilele **n** și **x** sunt de tip întreg. Pentru ca în urma executării secvenței alăturate variabila **n** să memoreze numărul tuturor valorilor naturale pătrate perfecte din intervalul **[0, x]**, zona punctată poate fi înlocuită cu: (4p.)
- ```
n=0;
while(.....)
 n=n+1;
```

- a.  $n \leq x * x$
- b.  $n * n \leq x$
- c.  $n \leq (x-1) * (x-1)$
- d.  $(n-1) * (n-1) \leq x$

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

3. Variabilele întregi **baza1** și **inaltime1** memorează baza, respectiv, înălțimea unui triunghi, iar variabilele întregi **baza2** și **inaltime2** memorează baza, respectiv, înălțimea unui alt triunghi.

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. 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.

Se citește un număr natural,  $n$  ( $n \in [1, 10^9]$ ), și se cere să se scrie cel mai mic număr strict mai mare decât  $n$ .

**Exemplu:** dacă  $n=95$  atunci se scrie **98** (atât **98**, cât și **72=2+7+14+49**, sunt numere pare).

- a)** Scrieți, în pseudocod, algoritmul de rezolvare pentru problema enunțată. **(10p.)**  
**b)** Precizați rolul tuturor variabilelor care au intervenit în algoritmul realizat la punctul **a)** și indicați datele de intrare, respectiv datele de ieșire ale problemei enunțate. **(6p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

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

1. În secvența de instrucțiuni de mai jos toate variabilele sunt de tip întreg.

```
for(i=0; i<5; i++)
{ for(j=0; j<5; j++)
 if(.....) cout<<"! "; | printf("! ");
 else cout<<"? "; | printf("? ");
 cout<<endl; | printf("\n");
}
```

```
! ? ! ? !
? ! ? ! ?
! ? ! ? !
? ! ? ! ?
! ? ! ? !
```

Indicați o expresie care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, să se afișeze pe ecran valorile din figura de mai sus, în această ordine. **(4p.)**

- a.  $(i+j)\%2==0$       b.  $i\%2+j\%2==0$       c.  $i\%2+j==0$       d.  $i+j\%2==0$

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

2. Se consideră un tablou unidimensional în care elementele sunt, în această ordine, (2020, 2016, 2012, 2008, 2004, 2000, 1996). Pentru a verifica dacă în tablou există elementul cu valoarea **x**, se aplică metoda căutării binare.

Scrieți toate valorile pe care le poate avea **x**, astfel încât căutarea să se încheie după ce **x** a fost comparat cu exact două elemente ale tabloului. **(6p.)**

3. Două tablouri unidimensionale se numesc **anagrame** dacă elementele lor nenule sunt egale, dispuse eventual în altă ordine.

Scrieți un program C/C++ care citește de la tastatură numere naturale din intervalul  $[0, 200]$ : pe prima linie două numere nenule, **n** și **m**, pe linia a doua cele **n** elemente ale unui tablou unidimensional, iar pe linia a treia cele **m** elemente ale unui alt tablou unidimensional. Numerele aflate pe aceeași linie sunt separate prin câte un spațiu. Programul afișează pe ecran mesajul **DA**, dacă cele două tablouri sunt anagrame, sau mesajul **NU**, în caz contrar.

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

```
6 8
5 1 6 0 7 7
7 5 0 0 6 7 0 1
```

pe ecran se afișează mesajul **DA**

iar dacă se citesc, în această ordine, numerele:

```
4 4
5 1 6 0
6 5 6 1
```

pe ecran se afișează mesajul **NU**. **(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 **distincte** 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, toți termenii șirului aflat în fișier care aparțin intervalului  $[100, 999]$  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 123 502 4321 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.)**