

1. Algoritmul alăturat este reprezentat în pseudocod. S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[z]$ partea întreagă a numărului real z .

- a) Scrieți valorile afișate dacă se citesc, în această ordine, numerele 2 și 3. (6p.)
- b) Dacă pentru variabila a se citește numărul 3, scrieți cel mai mic b de două cifre, astfel încât să se afișeze 7. (6p.)
- c) Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat repetitiva cu test inițial cu o structură repetitivă cu test final. (6p.)
- d) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

```

citeste a,b
(numere naturale nenule)
c ← 1

Cât timp b>0 execută
  Dacă b % 2 = 1 atunci
    c ← (c*a) % 10
  a ← (a*a) % 10
  b ← [ b / 2 ]
scrie c

```

2. Variabilele întregi $nr1$, $nr2$, $num1$ și $num2$ memorează numărătorii, respectiv numitorii a două fracții. Declarați variabilele precizate și scrieți o secvență de instrucțiuni C/C++ care calculează și afișează pe ecran pe linii diferite numărătorul și numitorul sumei celor două fracții. Se va afișa mesajul "eronat" dacă cel puțin unul dintre numitori are valoarea 0. (6p.)

3.

Precizați care este valoarea variabilei a , dacă după execuția următoarei secvențe de algoritm se afișează valoarea 2.

(6p.)

```

m=5;
a=.....;
while( m<=a)
{ a=a-m;
  m=m+1;
}
cout<<a;

```

1. Se citește un număr natural, n , și se cere să se scrie oglinditul lui n , format doar din cifrele impare.

Exemplu: dacă $n=1235405$, atunci se scrie 5531, iar dacă $n=40$, atunci se scrie 0.

Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate. (10p.)

2. Scrieți un program C/C++ care citește de la tastatură un număr natural n ($n \in [3, 50]$), și cele n elemente ale unui tablou unidimensional, numere naturale din intervalul $[0, 104]$. Programul modifică în memorie tabloul prin ordonarea descrescătoare a elementelor, după ultima cifra, aflate pe poziții impare în tablou (prima poziție este 1), celelalte elemente păstrându-și poziția inițială, apoi afișează pe ecran elementele tabloului obținut, separate prin câte un spațiu.

Exemplu: dacă $n=9$, iar tabloul citit este (15, 0, 42, 9, 3, 4, 11, 7, 228), atunci se obține tabloul

(228, 0, 15, 9, 3, 4, 42, 7, 11).

(10p.)

3. Fișierul text NUMERE.IN conține un șir de cel puțin trei și cel mult 100000000 de numere întregi, fiecare având cel mult 3 cifre. Numerele sunt separate prin câte un spațiu.

Se consideră că un șir format din cel puțin trei termeni este un șir produs dacă oricare termen (începând cu al treilea) este format din produsul ultimilor doi termeni consecutivi din șir. Se cere să se citească numerele

din fișierul **NUMERE.IN** și să se afișeze pe ecran numărul maxim de termeni consecutivi ai unui șir produs. Dacă nu există nici-un astfel de șir produs, se afișează pe ecran mesajul **NU EXISTA**. Se utilizează un algoritm eficient din punct de vedere al timpului de execuție și al memoriei folosite.

Exemplu: dacă fișierul **NUMERE.IN** are conținutul:

3 5 15 75 17 11 2 3 7 3 21 36 756 17

atunci se afișează pe ecran valoarea **5** (corespunzătoare șirului **7 3 21 36 756**).

- a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(2p.)**
- b) Scrieți programul C/C++ corespunzător algoritmului descris. **(8p.)**