

SUBIECTUL al II-lea

(40 de puncte)

1. **Algoritmul alăturat este reprezentat în pseudocod.**
S-a notat cu $a \% b$ restul împărțirii numărului natural a la numărul natural nenul b și cu $[c]$ partea întreagă a numărului real c .
- a. Scrieți numărul afișat în urma executării algoritmului dacă se citesc, în această ordine, numerele 812302105 și 4. **(6p.)**
- b. Dacă pentru k se citește numărul 1, scrieți trei numere din intervalul $[10^3, 10^4]$ care pot fi citite pentru n , astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze un număr format din două cifre identice. **(6p.)**
- c. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat structura **repetă... până când** cu o structură repetitivă de alt tip. **(6p.)**
2. Pentru un punct se memorează coordonatele (abscisa și ordonata) în sistemul de coordonate xOy . Variabilele A_x și A_y , de tip întreg, memorează abscisa, respectiv ordonata punctului A , iar variabilele B_x și B_y , de tip întreg, memorează abscisa, respectiv ordonata punctului B . Declarați variabilele și scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran mesajul **intersecteaza axa**, dacă segmentul cu extremitățile în punctele A și B intersectează axa Ox a sistemului de coordonate, sau mesajul **nu intersecteaza axa**, în caz contrar. **(6p.)**
3. Variabila i este de tip întreg, iar variabila c este de tip **char**. Scrieți ce se afișează în urma executării secvenței de program alăturate. **(6p.)**
- ```

for(i=1; i<=7; i++)
{ if(i<=3) c='a'+(1+i/2)*(i%2);
 else c='0'+(1-i%2)*2;
 cout<<c; | printf("%c", c);
}

```

**SUBIECTUL al III-lea (30 de puncte)**

1. Se citește un număr natural  $n$  ( $n \geq 2$ ) și se cere să se scrie numărul divizorilor care, în descompunerea în factori primi a lui  $n$ , apar la o putere impară.  
Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate.  
**Exemplu:** dacă  $n=9000$ , se scrie 2 ( $9000=2^3 \cdot 3^2 \cdot 5^3$ ). **(10p.)**
2. Scrieți un program C/C++ care citește de la tastatură un număr natural  $n$  ( $n \in [2, 10^2]$ ) și construiește în memorie un tablou unidimensional cu  $n$  elemente, cu proprietatea că parcurgându-l de la stânga la dreapta se obține șirul primelor  $n$  pătrate perfecte impare, ordonat strict descrescător, ca în exemplu. Elementele tabloului obținut se afișează pe ecran, separate prin câte un spațiu.  
**Exemplu:** pentru  $n=6$  se obține tabloul (121, 81, 49, 25, 9, 1). **(10p.)**
3. Fișierul **bac.in** conține numere naturale: pe prima linie două numere din intervalul  $[1, 10^6]$ ,  $m$  și  $n$ , pe a doua linie un șir de  $m$  numere pare din intervalul  $[1, 10^9]$ , iar pe a treia linie un șir de  $n$  numere impare din intervalul  $[1, 10^9]$ . Numerele aflate pe aceeași linie a fișierului sunt separate prin câte un spațiu, și ambele șiruri sunt ordonate strict crescător.  
Se cere să se afișeze pe ecran, în ordine strict crescătoare, un șir format dintr-un număr maxim de termeni care aparțin cel puțin unuia dintre cele două șiruri, astfel încât oricare două elemente aflate pe poziții consecutive să fie de paritate diferită. Numerele afișate sunt separate prin câte un spațiu.  
Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
- ```

5 3
2 4 8 10 14
3 5 11
    
```
- Exemplu:** dacă fișierul are conținutul alăturat, se afișează pe ecran
2 3 4 5 8 11 14 sau 2 3 4 5 10 11 14
- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
- b. Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**