

2. Variabila `c` memorează date despre o ciocolată: denumirea, gramajul și data expirării. Știind că expresiile `C/C++` de mai jos au ca valori numere naturale reprezentând gramajul, respectiv data (ziua, luna și anul) expirării produsului, scrieți definiția unei structuri cu eticheta `ciocolata`, care permite memorarea datelor precizate pentru o ciocolată, și declarați corespunzător variabila `c`.

`c.gramaj` `c.datae.zi` `c.datae.luna` `c.datae.an` (6p.)

3. În secvența alăturată, variabila `i` este de tip întreg, iar variabilele `s` și `t` pot memora câte un șir cu cel mult 20 de caractere. Scrieți ce se afișează pe ecran în urma executării secvenței. (6p.)
- ```
strcpy(s, "sanataTEA");
cout<<strlen(s); | printf("%d", strlen(s));
i=0;
while(i<strlen(s))
 if(s[i]=='a')
 { strcpy(t, s+i+1); strcpy(s+i, t); }
 else i=i+1;
cout<<s; | printf("%s", s);
```

### SUBIECTUL al III-lea

(30 de puncte)

1. Un joc online cu  $n$  jetoane poate fi jucat de un grup de  $k$  ( $k \geq 2$ ) jucători, numai dacă toate cele  $n$  jetoane pot fi distribuite în mod egal celor  $k$  jucători. Subprogramul `joc` are un singur parametru,  $n$ , prin care primește un număr natural ( $n \in [2, 10^4]$ ), reprezentând numărul de jetoane ale unui joc de tipul precizat. Subprogramul returnează numărul valorilor distincte pe care le poate avea  $k$  pentru acest joc.

Scrieți definiția completă a subprogramului.

**Exemplu:** dacă  $n=12$ , atunci subprogramul returnează numărul 5 (cele 12 jetoane se pot distribui în mod egal pentru o grupă de 2 jucători, de 3 jucători, de 4 jucători, de 6 jucători sau de 12 jucători).

(10p.)

2. Scrieți un program `C/C++` care citește de la tastatură numere naturale din intervalul  $[2, 10^2]$ :  $m$ ,  $n$  și elementele unui tablou bidimensional cu  $m$  linii și  $n$  coloane. Programul afișează pe ecran, pentru fiecare linie a sa, cea mai mare dintre valorile strict mai mici decât 21 memorate în aceasta, sau mesajul `nu exista`, dacă nu există nicio astfel de valoare pe linia respectivă. Numerele, respectiv mesajele, sunt afișate pe linii separate ale ecranului, în ordinea liniilor corespunzătoare din tablou.

**Exemplu:** pentru  $m=3$ ,  $n=5$  și tabloul alăturat se afișează pe ecran

```
16
nu exista
9
```

|    |    |    |    |    |
|----|----|----|----|----|
| 6  | 16 | 21 | 4  | 90 |
| 92 | 26 | 36 | 95 | 80 |
| 5  | 2  | 9  | 7  | 3  |

(10p.)

3. Fișierele `bac1.txt` și `bac2.txt` conțin numere naturale din intervalul  $[1, 10^5]$ : fișierul `bac1.txt` conține pe prima linie un număr  $n_1$ , iar pe a doua linie un șir de  $n_1$  numere, iar fișierul `bac2.txt` conține pe prima linie un număr  $n_2$ , iar pe a doua linie un șir de  $n_2$  numere. Numerele aflate pe aceeași linie a unui fișier sunt ordonate crescător și sunt separate prin câte un spațiu.

Se cere să se afișeze pe ecran, în ordine crescătoare, separate prin câte un spațiu, numerele divizibile cu 5 care se găsesc doar în unul dintre șirurile aflate în cele două fișiere. Dacă nu există niciun astfel de număr, se afișează pe ecran mesajul `nu exista`. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** dacă fișierul `bac1.txt` conține numerele  $7$  1 2 3 4 7 20 60 iar fișierul `bac2.txt` conține numerele  $9$  3 5 7 8 9 10 12 20 24

pe ecran se afișează, în această ordine, numerele 5 10 60.

a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)

b. Scrieți programul `C/C++` corespunzător algoritmului proiectat. (8p.)