

3. Pentru un telefon achiziționat în anul curent se memorează următoarele date: o literă corespunzătoare tipului sistemului de operare (litera **A** pentru **Android** sau litera **W** pentru **Windows**), precum și luna achiziționării (număr natural din intervalul $[1, 5]$). Variabilele **tip1** și **luna1** memorează litera corespunzătoare tipului sistemului de operare, respectiv luna achiziționării unui telefon, iar variabilele **tip2** și **luna2** memorează litera corespunzătoare tipului sistemului de operare, respectiv luna achiziționării unui alt telefon.
- Declarați variabilele **tip1** și **tip2** și scrieți o secvență C/C++ în urma executării căreia să se afișeze pe ecran mesajul **Aceeasi luna**, în cazul în care telefoanele sunt achiziționate în aceeași lună, sau tipul sistemului de operare al ultimului telefon achiziționat, în caz contrar.
- Exemplu:** dacă **tip1** memorează litera **A**, **tip2** memorează litera **W**, **luna1=4** și **luna2=1**, se afișează pe ecran cuvântul **Android** (6p.)

SUBIECTUL al III-lea

(30 de puncte)

1. La un laborator sunt studiate aglomerările de fulgi de nea formate din câte **nouă** cristale de **patru** tipuri diferite date (notate cu **1**, **2**, **3** sau **4**), astfel încât din fiecare tip să existe cel puțin câte un cristal. O astfel de aglomerare de fulgi a fost reprezentată printr-un număr natural, în care fiecare cifră reprezintă tipul unui cristal. Se citește un număr natural, **n** ($n \in [0, 10^9]$), și se cere să se scrie valoarea **1**, dacă prin **n** este reprezentată o aglomerare de fulgi de nea dintre cele studiate, sau **0** în caz contrar. Scrieți în pseudocod algoritmul corespunzător.
- Exemplu:** dacă **n=112243413** se scrie **1**, iar dacă **n=12314** sau **n=112253513** sau **n=112243457** sau **n=111122223**, se scrie **0**. (10p.)
2. Fiind date două numere naturale **a** și **b**, numim valoare **generată** de **a** și **b** un număr obținut din **a** prin alipirea la stânga sau la dreapta sa a cifrelor lui **b**, în ordinea în care apar în acesta.
- Exemplu:** dacă **a=123** și **b=45**, se pot obține două numere generate de acestea: **12345** și **45123**. Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul $[2, 50]$, **n**, și **k**, și construiește în memorie un tablou unidimensional cu **n** elemente, numerotate de la **1** la **n**, în care oricare al **i**-lea element ($i \in [1, n]$) este egal cu cea mai mică valoare generată de **i** și de câtul împărțirii lui **k** la **i**. Programul afișează pe ecran elementele tabloului obținut, separate prin câte un spațiu.
- Exemplu:** pentru **n=7** și **k=4**, se obține tabloul (**14, 22, 13, 14, 5, 6, 7**) pentru **n=7** și **k=10**, se obține tabloul (**101, 25, 33, 24, 25, 16, 17**). (10p.)
3. Fișierul **numere.in** conține un șir de cel mult 10^6 numere naturale din intervalul $[0, 99]$. Numerele din fișier sunt separate prin câte un spațiu. Se cere să se determine primul număr din șir care conține cea mai mare cifră ce apare în scrierea numerelor din fișier. Numărul determinat se afișează pe ecran. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
- Exemplu:** dacă fișierul conține numerele **2 12 34 5 38 30 87 70 11 8 82 25** se afișează pe ecran valoarea **38**.
- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)
b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)