

## Problema 3-E2 – Cod binar

Prelucrarea digitală a informației presupune prelucrarea unor valori reprezentate binar (baza 2 - valori de 0 sau 1). Pe de altă parte, calculele matematice uzuale le realizăm în baza 10. Astfel, în multe situații, este necesară verificarea conversiei unor numere zecimale în binar.

### Cerință

Având la dispoziție o succesiune de  $n$  valori întregi din intervalul  $[0; 255]$ , să se realizeze un program ce permite mai întâi conversia acestor valori în baza 2, pe 8 biți. Pentru conversia unei valori  $x$  în baza 2, se va implementa următorul algoritm: se împarte  $x$  la 2 și apoi succesiv se împarte câtul obținut la 2 până când câtul devine 0. Valoarea binară finală este dată concatenând resturile împărțirilor, pornind de la ultimul rest, până la primul rest obținut. Acestea sunt completate cu 0 (la stânga) pentru a ajunge la numărul de biți doriți, și anume 8.

Să se afișeze pe ecran numărul întreg ce corespunde valorii binare cu cele mai multe cifre de 0 (comparativ cu celelalte numere). Dacă sunt mai multe numere întregi cu același număr maxim de cifre de 0, se vor afișa toate aceste valori în ordinea în care au fost citite de la tastatură.

### Date de intrare

Se vor citi de la tastatură (fluxul *stdin*) următoarele date:

- o valoare întreagă  $n$  pentru numărul de valori întregi, urmată de caracterul *newline* (tasta *Enter*);
- succesiunea de valori întregi, introduse câte o valoare pe linie urmată de caracterul *newline* (tasta *Enter*).

### Date de ieșire

Programul va afișa pe ecran la ieșire, valoarea întreagă care în reprezentare binară conține cele mai multe cifre de 0, urmată de caracterul *newline* (tasta *Enter*). În cazul în care sunt mai multe valori în această situație, acestea vor fi afișate câte o valoare pe linie urmată de caracterul *newline* (tasta *Enter*), în ordinea în care apar în șirul inițial introdus de la tastatură.

**ATENȚIE la respectarea cerinței problemei: afișarea rezultatului trebuie făcută EXACT în modul în care a fost indicat! Cu alte cuvinte, pe stream-ul standard de ieșire nu se va afișa nimic în plus față de cerința problemei; ca urmare a evaluării automate, orice caracter suplimentar afișat, sau o afișare diferită de cea indicată, duc la un rezultat eronat și prin urmare la obținerea calificativului „Respins”.**

### Restricții și precizări

1. Numărul de valori  $n$  este strict mai mare decât 1 și strict mai mic ca 20, iar valorile întregi din succesiunea de valori sunt în intervalul  $[0; 255]$ .
2. **Atenție:** În funcție de limbajul de programare ales, fișierul ce conține codul trebuie să aibă una din extensiile *.c*, *.cpp*, *.java*, sau *.m*. Editorul web **nu va adăuga automat** aceste extensii și lipsa lor duce la imposibilitatea de compilare a programului!
3. **Atenție:** Fișierul sursă trebuie numit de candidat sub forma: *<nume>.<ext>* unde *nume* este numele de familie al candidatului și *extensia* este cea aleasă conform punctului anterior. Atenție la restricțiile impuse de limbajul Java legate de numele clasei și numele fișierului!

## Exemplu

Intrare	Ieșire
6 7 21 240 211 45 30	7 21
<b>Explicație:</b> 7 - 111 (resturi) - 00000111 (8 biți) - 5 de 0 21 - 10101 (resturi) - 00010101 (8 biți) - 5 de 0 240 - 11110000 (resturi) - 11110000 (8 biți) - 4 de 0 211 - 11010011 (resturi) - 11010011 (8 biți) - 3 de 0 45 - 101101 (resturi) - 00101101 (8 biți) - 4 de 0 30 - 11110 (resturi) - 00011110 (8 biți) - 4 de 0 Cele mai multe cifre de 0 sunt 5 și se găsesc la valorile 7 și 21.	

**Timp de lucru: 120 de minute**