

**Olimpiada Județeană de Informatică**  
**28 Februarie 2004**  
**Clasa a VI-a**

**Problema 1. "Număr de control"**

**100 puncte**

Gigel a primit spre păstrare un set de  $n$  cutii de greutate nu neapărat distincte. El a cântărit cutiile și pentru fiecare greutate distinctă a notat pe o foaie, în ordine crescătoare a greutăților, numărul de cutii cu greutatea respectivă.

Deoarece fratele său mai mic avea prostul obicei să se joace cu numerele scrise de el pe foaie, Gigel s-a gândit să calculeze un „număr de control” după următorul algoritm: începând de la primul număr a grupat numerele de apariții ale greutăților câte trei (dacă îi rămân numere negrupate la sfârșit, le ignoră). Dacă într-un grup sunt numai numere pare sau numai impare notează grupul cu cifra 1, altfel îl notează cu cifra 0. Din șirul astfel obținut, se formează un număr care are ca valoare cifra zecilor egală cu numărul de valori 1 și cifra unităților egală cu numărul de valori 0, obținându-se astfel „numărul de control”.

**Cerință**

Citind greutatele cutiilor, să se determine „numărul de control” și să se verifice dacă este număr prim.

**Date de intrare**

Se citește de la tastatură numărul  $n$  urmat de greutatele cutiilor.

**Date de ieșire**

Se va scrie pe ecran „numărul de control”, urmat de valoarea 0 sau 1 pe linia următoare. Pe următoarea linie se va afișa 1 dacă numărul este prim, respectiv 0 în caz contrar.

**Atenție: Se vor afișa doar două valori, fără mesaje, pe două linii diferite!!!**

**Restricții**

$$1 \leq n \leq 100$$

Fiecare greutate este un număr natural, mai mic sau egal cu 200

**Observație**

Datele de intrare sunt corecte (nu necesită validare).

**Exemplu**

Date de intrare

$$n=21$$

1 3 2 6 2 6 2 8 9 8 8 9 10 8 11 18 11 12 14 15 17

Date de ieșire

31

1

Explicație

După ordonare se obține șirul: 1 2 2 2 3 6 6 8 8 8 8 9 9 10 11 11 12 14 15 17 18

Se obține apoi:  $\underbrace{1\ 3\ 1}_{1}$   $\underbrace{2\ 4\ 2}_{1}$   $\underbrace{1\ 2\ 1}_{0}$   $\underbrace{1\ 1\ 1\ 1}_{1}$

1 1 0 1

Din valorile 1 1 0 1 se obține numărul de control 31, care este număr prim.

**Timp maxim de executare: 1 secundă/test**

## Problema 2. "Vânătoare"

100 puncte

Vânătorul șef al regelui Arthur a primit însărcinare să vâneze primele rațe ce se întorc din țările calde. Regele fiind un tip cu idei fixe, i-a cerut vânătorului să vâneze rațele albe cu săgeți albe, iar rațele negre cu săgeți negre.

Rațele vin în rânduri (stoluri) din ce în ce mai mari: mai întâi una, apoi două, trei, cinci, opt, treisprezece, ș.a.m.d. Se observă că numărul de rațe dintr-un rând este egal cu numărul de rațe de pe cele două rânduri anterioare. Rațele fiind niște creaturi ordonate zboară în rânduri, în care nu vei putea găsi două rațe de aceeași culoare alăturate, fiecare rând începând cu o rață albă.

Vânătorul știe că dacă a început să doboare o rață, trebuie să le doboare pe toate de pe rândul acesteia, deoarece supraviețuitoarele vor alerta celelalte rațe și ele nu se vor mai întoarce niciodată, iar vânătorul nostru își va pierde slujba.

### Cerință

Știind că vânătorul a primit **ka** săgeți albe și **kb** săgeți negre, trebuie să determinați câte rânduri de rațe a doborât și câte săgeți de fiecare tip i-au rămas, știind că el vrea să-și păstreze slujba.

### Date de intrare

Se citesc de la tastatură numerele **ka** și **kb** (în această ordine).

**Atenție: Valorile se introduc obligatoriu în această ordine!**

### Date de ieșire

Se va afișa pe ecran:

- numărul de rânduri doborâte
- numărul de săgeți albe rămase
- numărul de săgeți negre rămase.

### Restricții

$0 \leq ka, kb \leq 2.000.000.000$

### Observație

Datele de intrare sunt corecte (nu necesită validare).

### Exemplu

Pentru **ka**=9 și **kb**=10

Se va afișa:

4

2

6

### Explicație

Pentru exemplu avem rândurile (A- rață albă, N- rață neagră) de rațe:

A

A N

A N A

A N A N A

**Timp maxim de executare: 1 secundă/test**