

**Olimpiada Județeană de Informatică**  
**29 Februarie 2004**  
**Clasa a VII-a**

**Problema 1. "Șiruri"**

**100 puncte**

Se consideră un vector unidimensional  $x$  cu  $n$  componente numere naturale distincte, cel mult egale cu 32 000.

**Cerință**

Scrieți un program care să construiască vectorul  $y$  cu elemente din mulțimea  $\{1, 2, \dots, n\}$  astfel încât oricare ar fi numerele naturale  $i, j$  cu proprietatea că  $1 \leq i \leq n$ ,  $1 \leq j \leq n$  și  $x[i] < x[j]$  să avem  $y[i] < y[j]$ .

**Date de intrare**

Fișierul **siruri.in** va conține:

- pe prima linie numărul  $n$
- linia a doua componentele vectorului  $x$  separate printr-un spațiu.

**Date de iesire**

Fișierul **siruri.out** va conține pe prima linie componentele vectorului  $y$  separate printr-un spațiu.

**Restricții**

$$1 \leq n \leq 100$$

Componentele vectorului  $x$  sunt numere naturale cel mult egale cu **32 000**.

**Observație**

Datele de intrare sunt corecte (nu necesită validare).

**Exemplu**

<b>Siruri.in</b>	<b>Siruri.out</b>
6 12 3 7 16 10 1	5 2 3 6 4 1

**Timp maxim de executie/test: 0.1 secunde.**

**Problema 2. "NU te supăra, FRATE!"**

**100 puncte**

Jocul "Nu te supăra, frate!" se joacă de către doi jucatori. Pe o pistă circulară cu  $n$  căsuțe numerotate de la 1 la  $n$ , în ordinea dată de sensul acelor de ceasornic, se află înscrise valorile 0, 1 și 10. Cei doi jucători vor avea fiecare câte un pion și vor porni pe rând, din căsuța 1. Începe jucătorul 1. Fiecare jucător va muta pionul său conform cu valoarea obținută prin aruncarea zarului, cumulând sau pierzând puncte în funcție de căsuța în care ajunge. Fiecare jucător citește valoarea zarului când îi vine rândul.

Jocul are următoarele reguli:

1. Câștigătorul poate fi:

- jucătorul care ajunge primul din nou în căsuța cu numărul 1, indiferent de punctaj (exceptând cazul în care are punctaj 0)
  - în cazul în care se termină șirul aruncărilor cu zarul, înseamnă că jucătorii s-au plictisit și câștigă cel care a acumulat mai multe puncte, iar dacă au punctaje egale câștigă cel care se află în căsuța cu număr de ordine mai mare.
2. După aruncarea zarului, jucătorul mută pionul cu atâtea căsuțe cât indică valoarea zarului, în ordinea acelor de ceasornic, începând numărătoarea cu căsuța următoare poziției pe care se află. Prima căsuță nu conține valoarea 0 (zero).
3. După mutare pot apare următoarele situații:
- ajunge într-o căsuță cu valoarea 0 (zero) – jucătorul este penalizat, pierde toate punctele acumulate și reia jocul din pozitia 1
  - ajunge într-o căsuță cu valoarea 10 – primește un bonus de 10 puncte
  - ajunge într-o căsuță cu valoarea 1 – primește 1 punct
  - ajunge într-o căsuță în care se află celălalt pion (cu excepția căsuței 1, când câștigă) – jucătorul care ajunge ultimul este penalizat, pierde toate punctele și reia jocul de la căsuța 1.

### Cerință

Determinați jucătorul câștigător, pozițiile fiecărui jucător pe cerc și punctajul fiecărui jucător.

### Date de intrare

În fișierul JOC.IN se dau:

- **Pe prima linie numărul n al casutelor din cerc.**
- **Linia a doua conține o succesiune de n valori (0, 1 sau 10), separate printr-un spațiu, reprezentând valoarea fiecărei căsuțe.**
- **Linia a treia conține numărul de aruncări cu zarul**
- **Linia a patra conține o succesiune de valori întregi cuprinse între 1 și 6, separate printr-un spațiu, reprezentând aruncarea cu zarul.**

### Date de ieșire

**Fișierul JOC.OUT va conține 3 linii cu următoarele informații:**

**- pe prima linie: jucătorul câștigător**

- pe a doua linie: poziția și punctajul jucătorului numărul 1

- pe a treia linie: poziția și punctajul jucătorului numărul 2.

### Restricții

$7 \leq n \leq 100$

### Observație

Datele de intrare sunt corecte (nu necesită validare).

### Exemple

JOC.IN	JOC.OUT
10 1 1 1 1 1 10 0 1 1 0 8 3 6 2 4 1 3 5 3	2 6 10 1 3
JOC.IN	JOC.OUT
7 1 1 0 10 1 10 1 16 6 4 2 4 6 2 1 3 5 6 3 3 2 1 4 5	1 1 3 1 0
JOC.IN	JOC.OUT
12 1 10 1 1 0 10 1 1 1 10 0 1 4 5 6 3 3	2 9 11 10 11

**Timp maxim de executie/test: 0.1 secunde.**