

## PROBLEMA 1

100 puncte

## LOTERIE

Patronul unei firme a hotărât să organizeze o loterie a facturilor, la care să participe  $N$  facturi pe care le-a înregistrat într-o lună, atât de la furnizori cât și de la beneficiari. El și-a creat un sistem prin care extrage din cele  $N$  facturi, în mod aleator, exact  $M$ . Facturile care participă la loterie se înregistrează prin numere întregi distincte, pozitive, dacă sunt către beneficiari, și negative, dacă sunt de la furnizori.

**Cerință**

Ajută patronul să afle câte valori distincte se pot forma prin cumularea valorilor de înregistrare ale celor  $M$  facturi extrase la întâmplare.

**Date de intrare**

Fișierul **loterie.in** conține pe prima linie două numere naturale,  $N$  și  $M$ , reprezentând numărul de facturi, respectiv numărul de facturi extrase, și pe a doua linie  $N$  numere întregi, separate printr-un spațiu, reprezentând valorile cu care sunt înregistrate cele  $N$  facturi.

**Date de ieșire**

Fișierul **loterie.out** conține numărul cerut.

**Restricții și precizări**

- $1 \leq M < N \leq 50$ ;
- valoarea absolută a unei facturi este mai mică de 1000 de lei.

## Exemplu

loterie.in	loterie.out	Explicație
4 2 10 5 15 25	6	Sumele obținute sunt : 15 20 25 30 35 40

**Timp maxim de execuție:** 0.5 secunde/test.

**Memorie totală disponibilă** 16 MB, din care 1 MB pentru stivă

**Dimensiunea maximă a sursei:** 10 KB.

## PROBLEMA 2

100 puncte

## MODEL

Pentru realizarea unui model decorativ un artist are la dispoziție  $N$  pătrate colorate. Oricare două pătrate au culori diferite și dimensiuni diferite. Artistul așează în model toate pătratele unele peste altele păstrând același centru și vizibilitatea fiecărui pătrat.

## Cerință

Cunoscând pentru fiecare pătrat lungimea laturii (număr natural par) și culoarea dată de numărul de ordine al pătratului, precizați modelul realizat de artist.

## Date de intrare

Fișierul **model.in** conține pe prima linie numărul  $n$  iar pe următoarele  $n$  linii numerele  $l_i$ ,  $1 \leq i \leq n$ , numere naturale pare, latura pătratului cu numărul de ordine  $i$  (culoarea  $i$ )

## Date de ieșire

Fișierul **model.out** conține pe prima linie numărul natural  $m$  (dimensiunea modelului) iar pe următoarele  $m$  linii culorile modelului. Elementele din zona vizibilă a unui pătrat vor avea toate valoarea culorii pătratului.

## Restricții și precizări

$1 \leq n \leq 500$ ,  $2 \leq l_i \leq 1000$ ,  $1 \leq i \leq n$ ,  $l_i$  numere pare diferite

## Exemplu

model.in	model.out	Explicație
3	12	Sunt 3 pătrate cu laturile 6, 12, 2 și culorile 1, 2, 3
6	2 2 2 2 2 2 2 2 2 2 2 2	
12	2 2 2 2 2 2 2 2 2 2 2 2	
2	2 2 2 1 1 1 1 1 1 2 2 2	
	2 2 2 1 1 1 1 1 1 2 2 2	
	2 2 2 1 1 3 3 1 1 2 2 2	
	2 2 2 1 1 3 3 1 1 2 2 2	
	2 2 2 1 1 1 1 1 1 2 2 2	
	2 2 2 1 1 1 1 1 1 2 2 2	
	2 2 2 2 2 2 2 2 2 2 2 2	
	2 2 2 2 2 2 2 2 2 2 2 2	
	2 2 2 2 2 2 2 2 2 2 2 2	
	2 2 2 2 2 2 2 2 2 2 2 2	
	2 2 2 2 2 2 2 2 2 2 2 2	

**Timp maxim de execuție:** 1 secundă/test.

**Memorie totală disponibilă** 5 MB, din care 2 MB pentru stivă

**Dimensiunea maximă a sursei:** 10 KB.

## PROBLEMA 3

100 puncte

## TABĂRA

În județul Prahova s-a decis organizarea Taberei de vară pentru elevii cu performanțe deosebite la învățătură. De aceea organizatorii au cerut fiecărei școli din județ o listă cu numărul de elevi care s-ar califica să participe la aceasta tabără. Pentru că locul de desfășurare al taberei trebuia pregătit anterior venirii elevilor, organizatorii au decis să construiască  $n$  căsuțe, corespunzătoare celor  $n$  școli din județ, de aceeași dimensiune, în care să instaleze apoi paturile necesare. Problema a apărut când au primit lista elevilor propuși și au constatat că nu îi pot caza în aceleași condiții. Pentru a nu se ivi discuții între participanți, organizatorii au hotărât ca în fiecare căsuță să instaleze același număr de paturi, iar, dacă o școală este acceptată să participe la program, atunci toți elevii de la acea școală să fie primiți. Evident, în aceste condiții, unele școli au fost excluse din program.

## Cerință

Scrieți un program care să citească numărul  $n$  de școli, precum și numărul elevilor propuși din fiecare școală și care să determine ce efective de elevi vor fi acceptate în tabără.

## Date de intrare

Fișierul de intrare **tabara.in** conține pe prima linie un număr natural  $n$  care reprezintă numărul de școli. Pe cea de a doua linie sunt scrise  $n$  numere naturale separate prin câte un spațiu care reprezintă efectivele de elevi de la fiecare școală.

## Date de ieșire

Fișierul de ieșire **tabara.out** va conține pe prima linie două numere naturale  $k$  și  $p$ , care reprezintă numărul de școli alese pentru a participa la tabără și numărul de paturi instalate în fiecare cabană. Pe cea de a doua linie vor fi scrise  $k$  numere naturale separate prin câte un spațiu, reprezentând numărul de elevi propuși de cele  $k$  școli alese.

Pe a treia linie va fi un număr natural  $q$ , reprezentând numărul elevilor care nu au fost acceptați.

## Restricții și precizări

- $0 < n \leq 10000$  și  $0 < nr_i \leq 1000$ , pentru orice  $i=1, 2, \dots, n$
- În fiecare căsuță trebuie să fie cel puțin un pat.
- Într-o cabană pot sta elevi de la mai multe școli.
- Dacă există mai multe soluții, veți afișa una dintre acestea.
- Pentru 50% din teste,  $n \leq 1000$ .

## Exemplu

tabara.in	tabara.out	Explicație
7	4 2	O soluție posibilă:
4 7 4 2 1 13 2	7 4 2 1	Se aleg 4 școli și se instalează câte 2 paturi în fiecare cabană.
	19	Au fost respinși 19 elevi

**Timp maxim de execuție:** 0.5 secunde/test.

**Memorie totală disponibilă** 10 MB, din care 2 MB pentru stivă

**Dimensiunea maximă a sursei:** 10 KB.

## PROBLEMA 4

100 puncte

## CARTONAȘE

Un grup format din  $p$  copii dispune de  $K$  cartonașe numerotate cu numere naturale consecutive de la 1 la  $K$ . Fiecare copil își alege un număr natural  $x_i$ , unde  $1 \leq x_i \leq 100$  și extrage toate cartonașele cu numere de forma  $n * x_i + 1$ , cu  $n$  număr natural nenul, notându-și numerele extrase, după care pune cartonașele la loc. Numerele  $x_1, x_2, \dots, x_p$  alese de copii trebuie să fie distincte între ele.

## Cerință

Se cere să se determine câte cartonașe distincte au fost extrase de către cei  $p$  copii.

## Date de intrare

Fișierul de intrare **cart.in** va conține pe prima linie numerele naturale  $P$  și  $K$  separate prin câte un spațiu, iar pe a doua linie,  $P$  numere naturale  $x_1, x_2, \dots, x_p$  separate prin câte un spațiu.

## Date de ieșire

Fișierul de ieșire **cart.out** va conține o singură linie pe care va fi scris un singur număr natural reprezentând numărul de cartonașe distincte extrase.

## Restricții și precizări

$$1 \leq P \leq 15$$

$$1 \leq K \leq 2000000000$$

$$1 \leq x_i \leq 100, \text{ unde } 1 \leq i \leq P$$

40% din teste vor avea  $K \leq 64000$

## Exemple

cart.in	cart.out	Explicație
3 20 2 4 3	12	Au fost extrase cartonașele distincte cu numerele: 3, 4, 5, 7, 9, 10, 11, 13, 15, 16, 17, 19
4 41 5 4 3 2	30	Au fost extrase cartonașele distincte cu numerele: 3, 4, 5, 6, 7, 9, 10, 11, 13, 15, 16, 17, 19, 21, 22, 23, 25, 26, 27, 28, 29, 31, 33, 34, 35, 36, 37, 39, 40, 41

**Timp maxim de execuție:** 0.5 secunde/test.

**Memorie totală disponibilă** 16 MB, din care 1 MB pentru stivă

**Dimensiunea maximă a sursei:** 10 KB.

## PROBLEMA 5

100 puncte

## PUNCTE DE INFORMARE

Pentru o bună organizare a concertului Andre Rieu în România, s-a decis amplasarea unor puncte de informare în zona centrală a capitalei. Zona corespunzătoare a fost delimitată după principiile cartografierii la suprafața unui caroiaj de dimensiune  $N \times N$ , nodurile caroiajului fiind puncte având coordonatele din mulțimea  $\{0, 1, \dots, N-1\}$ . Primele centre de informare au fost deja plasate de către organizatorii români așa că la venirea echipei olandeze a rămas de plasat un singur punct de informare. Acesta trebuie să fie plasat astfel încât maximul dintre distanțele de la un nod oarecare al caroiajului până la cel mai apropiat punct de informare să fie minim.

## Cerință

Scrieți un program care să determine distanța maximă de la un nod al caroiajului la cel mai apropiat punct de informare **după plasarea ultimului punct de informare.**

## Date de intrare

Fișierul de intrare **puncte.in** conține pe prima linie două numere naturale separate prin spațiu  $N$  dimensiunea caroiajului și  $P$  numărul punctelor de informare plasate. Pe fiecare dintre următoarele  $P$  linii se află câte două numere naturale cuprinse între 0 și  $N-1$ , separate prin spațiu, reprezentând coordonatele unui punct de informare (abscisa, ordonata).

## Date de ieșire

Fișierul de ieșire **puncte.out** va conține pe prima linie un singur număr natural reprezentând distanța maximă de la un nod al caroiajului la cel mai apropiat punct de informare după plasarea ultimului punct de informare.

## Restricții și precizări

- $1 \leq N \leq 1000$
- $1 \leq P < N \cdot N$
- Se consideră că **distanța dintre două noduri ale caroiajului**  $(x_1, y_1)$  și  $(x_2, y_2)$  este distanța Manhattan  $|x_1 - x_2| + |y_1 - y_2|$

## Exemplu

puncte.in	puncte.out	Explicație
7 5 0 1 1 4 3 6 5 0 5 5	3	Noul punct de informare se va înființa în nodul de coordonate (3, 3). Altă soluție nu micșorează distanța maximă până la cel mai apropiat punct de informare conform cerinței.

**Timpi maxim de execuție:** 0.1 secunde/test.

**Memorie totală disponibilă** 32 MB, din care 1 MB pentru stivă

**Dimensiunea maximă a sursei:** 10 KB.