

## **Задача I. ИГРА С БОНБОНИ**

След дълго и изморително пътуване по света, проф. Маню Красев най-накрая успял да се прибере в родния си град Долно Нанагорнище. Веднага щом се прибрал в къщи, двамата му малки сина се втурнали към него да получат подаръци. Професорът очаквал това и им бил взел от хубави – по-хубави подаръци. Понеже Маню Красев бил професор и искал да развива мисленето на децата си той им бил приготвил една игра с изключително вкусни швейцарски шоколадови бонбони, които са разделени на  $N$  купчинки с по  $M_1, M_2, \dots, M_N$  бонбони. В играта участват двамата му сина, които се редуват да правят ходове. Играчът, който е на ред да прави ход, избира  $L$  купчинки ( $1 \leq L \leq K$ ) и премахва (изяжда) някакъв положителен брой бонбони от всяка от избраните купчинки. Играта приключва, когато бонбоните свършат, т.е. не могат да се правят повече ходове. Печели този който е направил последния ход. Победителят ще получи последен модел лаптоп **UVN** с процесор, който завършва безкраен цикъл само за две секунди!!!

Напишете програма, която по зададени  $N$  ( $1 < N < 2005$ ),  $K$  ( $1 \leq K \leq N$ ) и началните количества  $M_1, M_2, \dots, M_N$  на бонбоните във всяка от купчинките ( $1 \leq M_N \leq 100\,000$ ) определя дали играчът, който е първи на ход винаги губи играта, ако другият играч играе възможно най-добре или може да спечели, без значение как играе другият играч.

Програмата трябва да чете входните данни от **стандартния вход**. На първия ред е записано едно число  $T$  – броят на тестовите примери. Следват  $T$  теста, всеки от които се състои от два реда. На първия от тях са записани две числа –  $N$  и  $K$ , а на втория  $N$  числа – броят бонбони  $M_1, M_2, \dots, M_N$  във всяка от купчинките.

За всеки тестов пример програмата трябва да изведе на отделен ред на **стандартния изход** по едно число. Числото трябва да е 1 или 2 според това, дали съответно първият или вторият играч може да спечели със сигурност играта. Последователността на отговорите трябва да отговаря на последователността на входните данни.

### **Пример**

<b>Вход:</b>	<b>Изход:</b>
4	1
3 1	2
7 4 2	1
3 1	1
6 4 2	
3 1	
4 4 2	
6 6	
1 2 3 4 5 6	