

Problem 1 - Metaverse calendar (metaverse)

Meeting your friends in the metaverse can be hard sometimes!

Some of them log in every day, others only once a month.

Given a window of days, your goal is to calculate how many times you're all logged in together.

For the problem, we assume that in the first day, all players are logged in.

Input data

In each file, the first line contains T the number of test cases.

Then, for each test case, there are the following two lines:

1. Two space-separated integer numbers N and M , the number of friends in your group and the window of days in which you have to simultaneously log in.
2. N space-separated integer numbers P_n , the array of frequency of the N players, for example, if $P_n=3$ means that friend n logs in each third day.

Output data

The output file must contain T lines. For each test case in the input file, the output file must contain a line with the characters:

Case #t: S

where t is the test case number (from 1 to T) and S is the number of days when all the N players are logged in at the same time.

Constraints

- Days are counted starting from 1
- On day 1 all friends are connected
- $1 \leq T \leq 20$
- $2 \leq N \leq 6$
- $1 \leq M \leq 1\,000\,000$

Scoring

- **input 1** : $T = 1$, $N \leq 2$ and $M \leq 100$
- **input 2** : $T = 5$, $N \leq 3$ and $M \leq 1\,000$
- **input 3** : $T = 10$, $N \leq 4$ and $M \leq 10\,000$
- **input 4** : $T = 15$, $N \leq 5$ and $M \leq 100\,000$
- **input 5** : $T = 20$, $N \leq 6$ and $M \leq 1\,000\,000$

Examples

input	output
1 3 17 2 3 4	Case #1: 2

Explanation

In this example: P_1 logs in every second day, P_2 logs in every third day, P_3 logs in every fourth day. The solution is two, because they are logged in simultaneously only two days out of 17: on day one and on day 13.

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
P_1																	
P_2																	
P_3																	