Codeforces Round #674 (Div. 3)

A. Floor Number

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya goes to visit his classmate Petya. Vasya knows that Petya's apartment number is n.

There is only one entrance in Petya's house and the distribution of apartments is the following: the first floor contains 2 apartments, every other floor contains x apartments each. Apartments are numbered starting from one, from the first floor. I.e. apartments on the first floor have numbers 1 and 2, apartments on the second floor have numbers from 3 to $(x + 2)$, apartments on the third floor have numbers from $(x + 3)$ to $(2 \cdot x + 2)$, and so on.

Your task is to find the number of floor on which Petya lives. Assume that the house is always high enough to fit at least n apartments.

You have to answer t independent test cases.

Input

The first line of the input contains one integer t ($1 \le t \le 1000$) — the number of test cases. Then t test cases follow.

The only line of the test case contains two integers n and x ($1 \le n, x \le 1000$) — the number of Petya's apartment and the number of apartments on each floor of the house except the first one (there are two apartments on the first floor).

Output

For each test case, print the answer: the number of floor on which Petya lives.

Example

input
4 7 3 1 5 22 5 987 13
output
3 1 5 77

Note

Consider the first test case of the example: the first floor contains apartments with numbers 1 and 2, the second one contains apartments with numbers 3, 4 and 5, the third one contains apartments with numbers 6, 7 and 8. Therefore, Petya lives on the third floor.

In the second test case of the example, Petya lives in the apartment 1 which is on the first floor.

B. Symmetric Matrix

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Masha has n types of tiles of size 2×2. Each cell of the tile contains one integer. Masha has an **infinite number** of tiles of each type.

Masha decides to construct the square of size $m \times m$ consisting of the given tiles. This square also has to be a *symmetric with respect to the main diagonal matrix*, and each cell of this square has to be covered with exactly one tile cell, and also sides of tiles should be parallel to the sides of the square. All placed tiles cannot intersect with each other. Also, each tile should lie inside the square. See the picture in Notes section for better understanding.

Symmetric with respect to the main diagonal matrix is such a square s that for each pair (i, j) the condition $s[i][j] = s[j][i]$ holds. I.e. it is true that the element written in the i-row and j-th column equals to the element written in the j-th row and i-th column.

Your task is to determine if Masha can construct a square of size $m \times m$ which is a symmetric matrix and consists of tiles she has. Masha can use any number of tiles of each type she has to construct the square. Note that she **can not** rotate tiles, she can only place them in the orientation they have in the input.

You have to answer t independent test cases.

Input

The first line of the input contains one integer t ($1 \le t \le 100$) — the number of test cases. Then t test cases follow.

The first line of the test case contains two integers n and m ($1 \le n \le 100$, $1 \le m \le 100$) — the number of types of tiles and the size of the square Masha wants to construct.

The next $2n$ lines of the test case contain descriptions of tiles types. Types of tiles are written one after another, each type is written on two lines.

The first line of the description contains two positive (greater than zero) integers not exceeding 100 — the number written in the top left corner of the tile and the number written in the top right corner of the tile of the current type. The second line of the description contains two positive (greater than zero) integers not exceeding 100 — the number written in the bottom left corner of the tile and the number written in the bottom right corner of the tile of the current type.

It is forbidden to rotate tiles, it is only allowed to place them in the orientation they have in the input.

Output

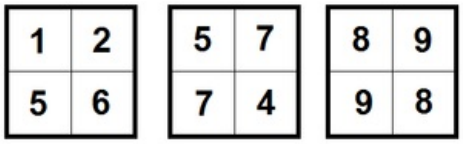
For each test case print the answer: "YES" (without quotes) if Masha can construct the square of size $m \times m$ which is a symmetric matrix. Otherwise, print "NO" (withtout quotes).

Example

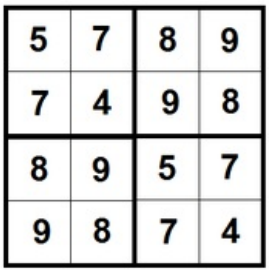
input
6 3 4 1 2 5 6 5 7 7 4 8 9 9 8 2 5 1 1 1 1 2 2 2 2 1 100 10 10 10 10 1 2 4 5 8 4 2 2 1 1 1 1 1 2 3 4 1 2 1 1 1 1
output
YES NO YES NO YES YES

Note

The first test case of the input has three types of tiles, they are shown on the picture below.



Masha can construct, for example, the following square of size 4×4 which is a symmetric matrix:



C. Increase and Copy

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Initially, you have the array a consisting of one element 1 ($a = [1]$).

In one move, you can do one of the following things:

- Increase some (**single**) element of a by 1 (choose some i from 1 to the current length of a and increase a_i by one);
- Append the copy of some (**single**) element of a to the end of the array (choose some i from 1 to the current length of a and append a_i to the end of the array).

For example, consider the sequence of five moves:

1. You take the first element a_1, append its copy to the end of the array and get $a = [1, 1]$.
2. You take the first element a_1, increase it by 1 and get $a = [2, 1]$.
3. You take the second element a_2, append its copy to the end of the array and get $a = [2, 1, 1]$.
4. You take the first element a_1, append its copy to the end of the array and get $a = [2, 1, 1, 2]$.
5. You take the fourth element a_4, increase it by 1 and get $a = [2, 1, 1, 3]$.

Your task is to find the **minimum** number of moves required to obtain the array with the sum at least n.

You have to answer t independent test cases.

Input
The first line of the input contains one integer t ($1 \le t \le 1000$) — the number of test cases. Then t test cases follow.

The only line of the test case contains one integer n ($1 \le n \le 10^9$) — the lower bound on the sum of the array.

Output
For each test case, print the answer: the **minimum** number of moves required to obtain the array with the sum at least n.

Example

input
5 1 5 42 1337 1000000000
output
0 3 11 72 63244

D. Non-zero Segments

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Kolya got an integer array $a_1, a_2, \ldots, a_n$. The array can contain both positive and negative integers, but Kolya doesn't like 0, so the array doesn't contain any zeros.

Kolya doesn't like that the sum of some subsegments of his array can be 0. The subsegment is some consecutive segment of elements of the array.

You have to help Kolya and change his array in such a way that it doesn't contain any subsegments with the sum 0. To reach this goal, you can insert any integers between any pair of adjacent elements of the array (integers can be really any: positive, negative, 0, any by absolute value, even such a huge that they can't be represented in most standard programming languages).

Your task is to find the minimum number of integers you have to insert into Kolya's array in such a way that the resulting array doesn't contain any subsegments with the sum 0.

Input
The first line of the input contains one integer n ($2 \le n \le 200\,000$) — the number of elements in Kolya's array.

The second line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9, a_i \neq 0$) — the description of Kolya's array.

Output
Print the minimum number of integers you have to insert into Kolya's array in such a way that the resulting array doesn't contain any subsegments with the sum 0.

Examples

input
4 1 -5 3 2
output
1

input
5 4 -2 3 -9 2
output
0

input
9 -1 1 -1 1 -1 1 1 -1 -1
output
6

input
8 16 -5 -11 -15 10 5 4 -4
output
3

Note

Consider the first example. There is only one subsegment with the sum 0. It starts in the second element and ends in the fourth element. It's enough to insert one element so the array doesn't contain any subsegments with the sum equal to zero. For example, it is possible to insert the integer 1 between second and third elements of the array.

There are no subsegments having sum 0 in the second example so you don't need to do anything.

E. Rock, Paper, Scissors

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice and Bob have decided to play the game "Rock, Paper, Scissors".

The game consists of several rounds, each round is independent of each other. In each round, both players show one of the following things at the same time: rock, paper or scissors. If both players showed the same things then the round outcome is a draw. Otherwise, the following rules applied:

- if one player showed rock and the other one showed scissors, then the player who showed rock is considered the winner and the other one is considered the loser;
- if one player showed scissors and the other one showed paper, then the player who showed scissors is considered the winner and the other one is considered the loser;
- if one player showed paper and the other one showed rock, then the player who showed paper is considered the winner and the other one is considered the loser.

Alice and Bob decided to play exactly n rounds of the game described above. Alice decided to show rock a_1 times, show scissors a_2 times and show paper a_3 times. Bob decided to show rock b_1 times, show scissors b_2 times and show paper b_3 times. Though, both Alice and Bob **did not choose** the sequence in which they show things. It is guaranteed that $a_1 + a_2 + a_3 = n$ and $b_1 + b_2 + b_3 = n$.

Your task is to find two numbers:

1. the minimum number of round Alice can win;
2. the maximum number of rounds Alice can win.

Input

The first line of the input contains one integer n ($1 \le n \le 10^9$) — the number of rounds.

The second line of the input contains three integers a_1, a_2, a_3 ($0 \le a_i \le n$) — the number of times Alice will show rock, scissors and paper, respectively. It is guaranteed that $a_1 + a_2 + a_3 = n$.

The third line of the input contains three integers b_1, b_2, b_3 ($0 \le b_j \le n$) — the number of times Bob will show rock, scissors and paper, respectively. It is guaranteed that $b_1 + b_2 + b_3 = n$.

Output
Print two integers: the minimum and the maximum number of rounds Alice can win.

Examples

input
2 0 1 1 1 1 0
output
0 1

input
15 5 5 5 5 5 5
output
0 15

input
3 0 0 3 3 0 0
output
3 3

input
686 479 178 29 11 145 530
output
22 334

input
319 10 53 256 182 103 34
output
119 226

Note
In the first example, Alice will not win any rounds if she shows scissors and then paper and Bob shows rock and then scissors. In the best outcome, Alice will win one round if she shows paper and then scissors, and Bob shows rock and then scissors.

In the second example, Alice will not win any rounds if Bob shows the same things as Alice each round.

In the third example, Alice always shows paper and Bob always shows rock so Alice will win all three rounds anyway.

F. Number of Subsequences

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a string s consisting of lowercase Latin letters "a", "b" and "c" and question marks "?".

Let the number of question marks in the string s be k. Let's replace each question mark with one of the letters "a", "b" and "c". Here we can obtain all 3^k possible strings consisting only of letters "a", "b" and "c". For example, if $s =$"ac?b?c" then we can obtain the following strings: $[$"acabac", "acabbc", "acabcc", "acbbac", "acbbbc", "acbbcc", "accbac", "accbbc", "accbcc"$]$.

Your task is to count the total number of subsequences "abc" in all resulting strings. Since the answer can be very large, print it modulo $10^9 + 7$.

A subsequence of the string t is such a sequence that can be derived from the string t after removing some (possibly, zero) number of letters without changing the order of remaining letters. For example, the string "baacbc" contains two subsequences "abc" — a subsequence consisting of letters at positions $(2, 5, 6)$ and a subsequence consisting of letters at positions $(3, 5, 6)$.

Input
The first line of the input contains one integer n $(3 \le n \le 200\,000)$ — the length of s.

The second line of the input contains the string s of length n consisting of lowercase Latin letters "a", "b" and "c" and question

marks"?".

Output

Print the total number of subsequences "abc" in all strings you can obtain if you replace all question marks with letters "a", "b" and "c", modulo $10^9 + 7$.

Examples

input
6 ac?b?c
output
24

input
7 ???????
output
2835

input
9 cccbbbaaa
output
0

input
5 a???c
output
46

Note

In the first example, we can obtain 9 strings:

- "acabac" — there are 2 subsequences "abc",
- "acabbc" — there are 4 subsequences "abc",
- "acabcc" — there are 4 subsequences "abc",
- "acbbac" — there are 2 subsequences "abc",
- "acbbbc" — there are 3 subsequences "abc",
- "acbbcc" — there are 4 subsequences "abc",
- "accbac" — there is 1 subsequence "abc",
- "accbbc" — there are 2 subsequences "abc",
- "accbcc" — there are 2 subsequences "abc".

So, there are $2 + 4 + 4 + 2 + 3 + 4 + 1 + 2 + 2 = 24$ subsequences "abc" in total.