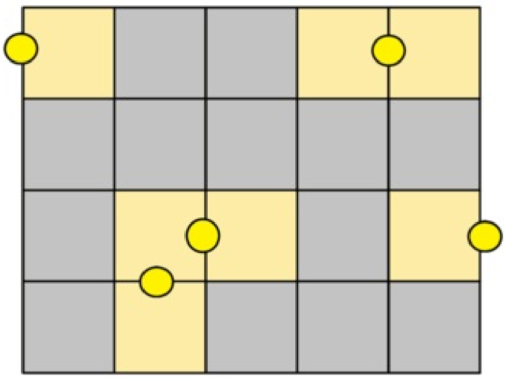
A. Park Lighting

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

Due to the coronavirus pandemic, city authorities obligated citizens to keep a social distance. The mayor of the city Semyon wants to light up Gluharniki park so that people could see each other even at night to keep the social distance.

The park is a rectangular table with n rows and m columns, where the cells of the table are squares, and the boundaries between the cells are streets. External borders are also streets. Every street has length 1. For example, park with $n = m = 2$ has 12 streets.

You were assigned to develop a plan for lighting the park. You can put lanterns in the middle of the streets. The lamp lights two squares near it (or only one square if it stands on the border of the park).



The park sizes are: $n = 4$, $m = 5$. The lighted squares are marked yellow. Please note that all streets have length 1. Lanterns are placed in the middle of the streets. In the picture **not all** the squares are lit.

Semyon wants to spend the least possible amount of money on lighting but also wants people throughout the park to keep a social distance. So he asks you to find the minimum number of lanterns that are required to light all the squares.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases in the input. Then t test cases follow.

Each test case is a line containing two integers n, m ($1 \le n, m \le 10^4$) — park sizes.

Output

Print t answers to the test cases. Each answer must be a single integer — the minimum number of lanterns that are required to light all the squares.

Example

input

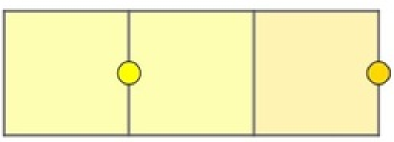
```
5
1 1
1 3
2 2
3 3
5 3
```

output

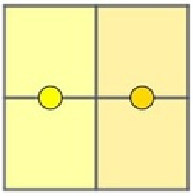
```
1
2
2
5
8
```

Note

Possible optimal arrangement of the lanterns for the 2-nd test case of input data example:



Possible optimal arrangement of the lanterns for the 3-rd test case of input data example:



B. Maria Breaks the Self-isolation

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

Maria is the most active old lady in her house. She was tired of sitting at home. She decided to organize a ceremony against the coronavirus.

She has n friends who are also grannies (Maria is not included in this number). The i-th granny is ready to attend the ceremony, provided that at the time of her appearance in the courtyard there will be at least a_i other grannies there. Note that grannies can come into the courtyard at the same time. Formally, the granny i agrees to come if the number of other grannies who came earlier or at the same time with her is greater than or equal to a_i.

Grannies gather in the courtyard like that.

- Initially, only Maria is in the courtyard (that is, the initial number of grannies in the courtyard is 1). All the remaining n grannies are still sitting at home.
- On each step Maria selects a subset of grannies, none of whom have yet to enter the courtyard. She promises each of them that at the time of her appearance there will be at least a_i other grannies (including Maria) in the courtyard. Maria can call several grannies at once. In this case, the selected grannies will go out into the courtyard **at the same moment of time**.
- She cannot deceive grannies, that is, the situation when the i-th granny in the moment of appearing in the courtyard, finds that now there are strictly less than a_i other grannies (except herself, but including Maria), is prohibited. Please note that if several grannies appeared in the yard at the same time, then each of them sees others at the time of appearance.

Your task is to find what maximum number of grannies (including herself) Maria can collect in the courtyard for the ceremony. After all, the more people in one place during quarantine, the more effective the ceremony!

Consider an example: if $n = 6$ and $a = [1, 5, 4, 5, 1, 9]$, then:

- at the first step Maria can call grannies with numbers 1 and 5, each of them will see two grannies at the moment of going out into the yard (note that $a_1 = 1 \le 2$ and $a_5 = 1 \le 2$);
- at the second step, Maria can call grannies with numbers 2, 3 and 4, each of them will see five grannies at the moment of going out into the yard (note that $a_2 = 5 \le 5$, $a_3 = 4 \le 5$ and $a_4 = 5 \le 5$);
- the 6-th granny cannot be called into the yard — therefore, the answer is 6 (Maria herself and another 5 grannies).

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases in the input. Then test cases follow.

The first line of a test case contains a single integer n ($1 \le n \le 10^5$) — the number of grannies (Maria is not included in this number).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 2 \cdot 10^5$).

It is guaranteed that the sum of the values n over all test cases of the input does not exceed 10^5.

Output

For each test case, print a single integer k ($1 \le k \le n + 1$) — the maximum possible number of grannies in the courtyard.

Example

input

```
4
5
1 1 2 2 1
6
2 3 4 5 6 7
6
1 5 4 5 1 9
5
1 2 3 5 6
```

output

```
6
1
6
4
```

Note

In the first test case in the example, on the first step Maria can call all the grannies. Then each of them will see five grannies when they come out. Therefore, Maria and five other grannies will be in the yard.

In the second test case in the example, no one can be in the yard, so Maria will remain there alone.

The third test case in the example is described in the details above.

In the fourth test case in the example, on the first step Maria can call grannies with numbers 1, 2 and 3. If on the second step Maria calls 4 or 5 (one of them), then when a granny appears in the yard, she will see only four grannies (but it is forbidden). It means that Maria can't call the 4-th granny or the 5-th granny separately (one of them). If she calls both: 4 and 5, then when they appear, they will see $4 + 1 = 5$ grannies. Despite the fact that it is enough for the 4-th granny, the 5-th granny is not satisfied. So, Maria cannot call both the 4-th granny and the 5-th granny at the same time. That is, Maria and three grannies from the first step will be in the yard in total.

C. Celex Update

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

During the quarantine, Sicromoft has more free time to create the new functions in "`Celex-2021`". The developers made a new function `GAZ-GIZ`, which infinitely fills an infinite table to the right and down from the upper left corner as follows:

1	2	4	7	11	...
3	5	8	12	...	
6	9	13	...		
10	14	...			
15	...				
...					

The cell with coordinates (x, y) is at the intersection of x-th row and y-th column. Upper left cell $(1, 1)$ contains an integer 1. The developers of the `SUM` function don't sleep either. Because of the boredom, they teamed up with the developers of the `RAND` function, so they added the ability to calculate the sum on an arbitrary path from one cell to another, moving down or right. Formally, from the cell (x, y) in one step you can move to the cell $(x+1, y)$ or $(x, y+1)$.

After another Dinwows update, Levian started to study "`Celex-2021`" (because he wants to be an accountant!). After filling in the table with the `GAZ-GIZ` function, he asked you to calculate the quantity of possible different amounts on the path from a given cell (x_1, y_1) to another given cell (x_2, y_2), if you can only move one cell down or right.

Formally, consider all the paths from the cell (x_1, y_1) to cell (x_2, y_2) such that each next cell in the path is located either to the down or to the right of the previous one. Calculate the number of different sums of elements for all such paths.

Input

The first line contains one integer t ($1 \le t \le 57179$) — the number of test cases.

Each of the following t lines contains four natural numbers x_1, y_1, x_2, y_2 ($1 \le x_1 \le x_2 \le 10^9$, $1 \le y_1 \le y_2 \le 10^9$) — coordinates of the start and the end cells.

Output

For each test case, in a separate line, print the number of possible different sums on the way from the start cell to the end cell.

Example

input

```
4
1 1 2 2
1 2 2 4
179 1 179 100000
5 7 5 7
```

output

```
2
3
1
1
```

Note

In the first test case there are two possible sums: $1+2+5=8$ and $1+3+5=9$.

D. The Best Vacation

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

You've been in love with Coronavirus-chan for a long time, but you didn't know where she lived until now. And just now you found out that she lives in a faraway place called Naha.

You immediately decided to take a vacation and visit Coronavirus-chan. Your vacation lasts exactly x days and that's the exact number of days you will spend visiting your friend. You will spend exactly x consecutive (successive) days visiting Coronavirus-chan.

They use a very unusual calendar in Naha: there are n months in a year, i-th month lasts exactly d_i days. Days in the i-th month are numbered from 1 to d_i. There are no leap years in Naha.

The mood of Coronavirus-chan (and, accordingly, her desire to hug you) depends on the number of the day in a month. In particular, you get j hugs if you visit Coronavirus-chan on the j-th day of the month.

You know about this feature of your friend and want to plan your trip to get as many hugs as possible (and then maybe you can win the heart of Coronavirus-chan).

Please note that your trip should **not necessarily** begin and end in the same year.

Input

The first line of input contains two integers n and x ($1 \le n \le 2 \cdot 10^5$) — the number of months in the year and the number of days you can spend with your friend.

The second line contains n integers $d_1, d_2, \ldots, d_n$, d_i is the number of days in the i-th month ($1 \le d_i \le 10^6$).

It is guaranteed that $1 \le x \le d_1 + d_2 + \ldots + d_n$.

Output

Print one integer — the maximum number of hugs that you can get from Coronavirus-chan during the best vacation in your life.

Examples

input

```
3 2
1 3 1
```

output

```
5
```

input

```
3 6
3 3 3
```

output

```
12
```

input

```
5 6
4 2 3 1 3
```

output

```
15
```

Note

In the first test case, the numbers of the days in a year are (indices of days in a corresponding month) $\{1, 1, 2, 3, 1\}$. Coronavirus-chan will hug you the most if you come on the third day of the year: $2+3=5$ hugs.

In the second test case, the numbers of the days are $\{1, 2, 3, 1, 2, 3, 1, 2, 3\}$. You will get the most hugs if you arrive on the third day of the year: $3+1+2+3+1+2=12$ hugs.

In the third test case, the numbers of the days are $\{1, 2, 3, 4, 1, 2, 1, 2, 3, 1, 1, 2, 3\}$. You will get the most hugs if you come on the twelfth day of the year: your friend will hug you $2+3+1+2+3+4=15$ times.

E. Are You Fired?

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

Levian works as an accountant in a large company. Levian knows how much the company has earned in each of the n consecutive months — in the i-th month the company had income equal to a_i (positive income means profit, negative income means loss, zero income means no change). Because of the general self-isolation, the first $\lceil \frac{n}{2} \rceil$ months income might have been completely unstable, but then everything stabilized and for the last $\lfloor \frac{n}{2} \rfloor$ months **the income was the same**.

Levian decided to tell the directors $n-k+1$ numbers — the total income of the company for each k consecutive months. In other words, for each i between 1 and $n-k+1$ he will say the value $a_i + a_{i+1} + \ldots + a_{i+k-1}$. For example, if $a = [-1, 0, 1, 2, 2]$ and $k = 3$ he will say the numbers $0, 3, 5$.

Unfortunately, if at least one total income reported by Levian is not a profit (income ≤ 0), the directors will get angry and fire the failed accountant.

Save Levian's career: find any such k, that for each k months in a row the company had made a profit, or report that it is impossible.

Input

The first line contains a single integer n ($2 \le n \le 5 \cdot 10^5$) — the number of months for which Levian must account.

The second line contains $\lceil \frac{n}{2} \rceil$ integers $a_1, a_2, \ldots, a_{\lceil \frac{n}{2} \rceil}$, where a_i ($-10^9 \le a_i \le 10^9$) — the income of the company in the i-th month.

Third line contains a single integer x ($-10^9 \le x \le 10^9$) — income in every month from $\lceil \frac{n}{2} \rceil + 1$ to n.

Output

In a single line, print the appropriate integer k or -1, if it does not exist.

If there are multiple possible answers, you can print any.

Examples

input

```
3
2 -1
2
```

output

```
2
```

input

```
5
2 2 -8
2
```

output

```
-1
```

input

```
6
-2 -2 6
-1
```

output

```
4
```

Note

In the first example, $k=2$ and $k=3$ satisfy: in the first case, Levian will report the numbers $1, 1$, and in the second case — one number 3.

In the second example, there is no such k.

In the third example, the only answer is $k=4$: he will report the numbers $1, 2, 3$.

F. Tasty Cookie

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Oh, no!

The coronavirus has caught you, and now you're sitting in a dark cellar, with tied legs (but not hands). You have a delicious cookie, a laptop in front of you, and your ideal development environment is open. The coronavirus convinces you to solve the following problem.

You are given two arrays A and B of size n. You can do operations of two types with array A:

- Reverse array A. That is the array $[A_1,\ A_2,\ \ldots,\ A_n]$ transformes into $[A_n,\ A_{n-1},\ \ldots,\ A_1]$.
- Replace A with an array of its prefix sums. That is, the array $[A_1,\ A_2,\ \ldots,\ A_n]$ goes to $[A_1,\ (A_1+A_2),\ \ldots,\ (A_1+A_2+\ldots+A_n)]$.

You need to understand if you can get an array B from the array A. If it is possible, you will have to restore the order of these operations by minimizing the number of operations of the second type. Fortunately, the coronavirus is good today, so he has allowed you not to restore actions if the minimum number of second type operations is more than $2 \cdot 10^5$. But coronavirus resents you, so if you restore the answer, the total number of operations should not exceed $5 \cdot 10^5$.

Solve this problem and get the cookie, or the coronavirus will extend the quarantine for five years and make the whole economy collapse!

Input

The first line contains a single integer n ($1 \le n \le 2 \cdot 10^5$).

The second line contains n integers $A_1, A_2, \ldots, A_n$ ($1 \le A_i \le 10^{12}$).

The third line contains n integers $B_1, B_2, \ldots, B_n$ ($1 \le B_i \le 10^{12}$).

Output

If you cannot get B from the A array, print "IMPOSSIBLE" (without quotes) on a single line.

If the minimum number of operations of the second type exceeds $2 \cdot 10^5$, print "BIG" (without quotes). In the second line print the number of operations of the second type, that needs to be applied to get array B from A.

Otherwise, in the first line print "SMALL" (without quotes). In the second line print the total number of operations of the first and second types $m \le 5 \cdot 10^5$ (it is guaranteed that in this case there is such a sequence of actions). In the third line print a line of length m, consisting of characters 'R"and 'P' (without quotes).

The i-th character should be 'R', if the i-th action is of the first type, and should be 'P', otherwise.

If there are several such sequences, you can print any of them.

You can print each character in the uppercase or in the lowercase.

Examples

input

```
2
5 7
5 7
```

output

```
SMALL
0
```

input

```
2
1 1
300000 1
```

output

```
BIG
299999
```

input

```
2
10 1
13 14
```

output

```
SMALL
6
RPPPRP
```

input

```
3
1 2 1
2 1 2
```

output
IMPOSSIBLE

Note

In the first example, the arrays A and B already equal, so the number of required operations $= 0$.

In the second example, we need to replace A with the prefix sums 299999 times and then reverse the array. Since $299999 > 2 \cdot 10^5$, there is no need to restore the answer.

In the fourth example, you cannot get B from the A.