Codeforces Round 995 (Div. 3)

A. Preparing for the Olympiad

2 seconds, 256 megabytes

Monocarp and Stereocarp are preparing for the Olympiad. There are n days left until the Olympiad. On the i-th day, if Monocarp plans to practice, he will solve a_i problems. Similarly, if Stereocarp plans to practice on the same day, he will solve b_i problems.

Monocarp can train on any day he wants. However, Stereocarp watches Monocarp and follows a different schedule: if Monocarp trained on day i and $i < n$, then Stereocarp will train on day $(i + 1)$.

Monocarp wants to organize his training process in a way that the difference between the number of problems he solves and the number of problems Stereocarp solves is as large as possible. Formally, Monocarp wants to maximize the value of $(m - s)$, where m is the number of problems he solves, and s is the number of problems Stereocarp solves. Help Monocarp determine the maximum possible difference in the number of solved problems between them.

Input

The first line contains a single integer t $(1 \le t \le 10^3)$ — the number of test cases.

The first line of each test case contains a single integer n $(1 \le n \le 100)$.

The second line contains n integers $a_1, a_2, \dots, a_n$ $(1 \le a_i \le 100)$.

The third line contains n integers $b_1, b_2, \dots, b_n$ $(1 \le b_i \le 100)$.

Output

For each test case, print a single integer — the maximum possible difference between the number of problems Monocarp solves and the number of problems Stereocarp solves.

input
```
4
2
3 2
2 1
1
5
8
3
1 1 1
2 2 2
6
8 2 5 6 2 6
8 2 7 4 3 4
```
output
```
4
5
1
16
```

Let's analyze the example from the statement:

- In the first test case, it is optimal for Monocarp to train both days; then Stereocarp will train on day 2.
- In the second test case, it is optimal for Monocarp to train on the only day, and Stereocarp will not train at all.
- In the third test case, it is optimal for Monocarp to train on the last day (and only on that day).
- In the fourth test case, it is optimal for Monocarp to train on days $1, 3, 4, 6$; then Stereocarp will train on days $2, 4, 5$.

B. Journey

1 second, 256 megabytes

Monocarp decided to embark on a long hiking journey.

He decided that on the first day he would walk a kilometers, on the second day he would walk b kilometers, on the third day he would walk c kilometers, on the fourth day, just like on the first, he would walk a kilometers, on the fifth day, just like on the second, he would walk b kilometers, on the sixth day, just like on the third, he would walk c kilometers, and so on.

Monocarp will complete his journey on the day when he has walked at least n kilometers in total. Your task is to determine the day on which Monocarp will complete his journey.

Input

The first line contains one integer t $(1 \le t \le 10^4)$ — the number of test cases.

Each test case consists of one line containing four integers n, a, b, c ($1 \le n \le 10^9$; $1 \le a, b, c \le 10^6$).

Output

For each test case, output one integer — the day on which Monocarp will have walked at least n kilometers in total and will complete his journey.

input
```
4
12 1 5 3
6 6 7 4
16 3 4 1
1000000000 1 1 1
```
output
```
5
1
6
1000000000
```

In the first example, over the first four days, Monocarp will cover $1 + 5 + 3 + 1 = 10$ kilometers. On the fifth day, he will cover another 5 kilometers, meaning that in total over five days he will have covered $10 + 5 = 15$ kilometers. Since $n = 12$, Monocarp will complete his journey on the fifth day.

In the second example, Monocarp will cover 6 kilometers on the first day. Since $n = 6$, Monocarp will complete his journey on the very first day.

In the third example, Monocarp will cover $3 + 4 + 1 + 3 + 4 + 1 = 16$ kilometers over the first six days. Since $n = 16$, Monocarp will complete his journey on the sixth day.

C. Preparing for the Exam

1.5 seconds, 256 megabytes

Monocarp is preparing for his first exam at the university. There are n different questions which can be asked during the exam, numbered from 1 to n. There are m different lists of questions; each list consists of exactly $n - 1$ different questions. Each list i is characterized by one integer a_i, which is the index of the only question which is **not present** in the i-th list. For example, if $n = 4$ and $a_i = 3$, the i-th list contains questions $[1, 2, 4]$.

During the exam, Monocarp will receive one of these m lists of questions. Then, the professor will make Monocarp answer all questions from the list. So, Monocarp will pass only if he knows all questions from the list.

Monocarp knows the answers for k questions $q_1, q_2, \dots, q_k$. For each list, determine if Monocarp will pass the exam if he receives that list.

Input

The first line contains one integer t $(1 \le t \le 10^4)$ — the number of test cases.

Each test case consists of three lines:

- the first line contains three integers n, m and k ($2 \le n \le 3 \cdot 10^5$; $1 \le m, k \le n$);
- the second line contains m **distinct** integers $a_1, a_2, \dots, a_m$ ($1 \le a_i \le n$; $a_i < a_{i+1}$);
- the third line contains k **distinct** integers $q_1, q_2, \dots, q_k$ ($1 \le q_i \le n$; $q_i < q_{i+1}$).

Additional constraints on the input:

- the sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, print a string of m characters. The i-th character should be `1` if Monocarp passes the exam if he receives the i-th question list, `0` if Monocarp won't pass.

input
```
4
4 4 3
1 2 3 4
1 3 4
5 4 3
1 2 3 4
1 3 4
4 4 4
1 2 3 4
1 2 3 4
2 2 1
1 2
2
```
output
```
0100
0000
1111
10
```

In the first test case, Monocarp knows the questions $[1, 3, 4]$. Let's consider all the question lists:

- the first list consists of questions $[2, 3, 4]$. Monocarp doesn't know the 2-nd question, so he won't pass;
- the second list consists of questions $[1, 3, 4]$. Monocarp knows all these questions, so he will pass;
- the third list consists of questions $[1, 2, 4]$. Monocarp doesn't know the 2-nd question, so he won't pass;
- the fourth list consists of questions $[1, 2, 3]$. Monocarp doesn't know the 2-nd question, so he won't pass.

D. Counting Pairs

2 seconds, 256 megabytes

You are given a sequence a, consisting of n integers, where the i-th element of the sequence is equal to a_i. You are also given two integers x and y ($x \le y$).

A pair of integers (i, j) is considered *interesting* if the following conditions are met:

- $1 \le i < j \le n$;
- if you simultaneously remove the elements at positions i and j from the sequence a, the sum of the remaining elements is at least x and at most y.

Your task is to determine the number of *interesting* pairs of integers for the given sequence a.

Input

The first line contains one integer t ($1 \le t \le 10^4$) — the number of test cases.

Each test case consists of two lines:

- The first line contains three integers n, x, y ($3 \le n \le 2 \cdot 10^5$, $1 \le x \le y \le 2 \cdot 10^{14}$);
- The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^9$).

Additional constraint on the input: the sum of n across all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output one integer — the number of *interesting* pairs of integers for the given sequence a.

input
```
7
4 8 10
4 6 3 6
6 22 27
4 9 6 3 4 5
3 8 10
3 2 1
3 1 1
2 3 4
3 3 6
3 2 1
4 4 12
3 3 2 1
6 8 8
1 1 2 2 2 3
```
output
```
4
7
0
0
1
5
6
```

In the first example, there are 4 *interesting* pairs of integers:

1. $(1, 2)$;
2. $(1, 4)$;
3. $(2, 3)$;
4. $(3, 4)$.

E. Best Price

2 seconds, 256 megabytes

A batch of Christmas trees has arrived at the largest store in Berland. n customers have already come to the store, wanting to buy them.

Before the sales begin, the store needs to determine the price for one tree (the price is the same for all customers). To do this, the store has some information about each customer.

For the i-th customer, two integers a_i and b_i are known, which define their behavior:

- if the price of the product is at most a_i, the customer will buy a tree and leave a positive review;
- otherwise, if the price of the product is at most b_i, the customer will buy a tree but leave a negative review;
- otherwise, the customer will not buy a tree at all.

Your task is to calculate the maximum possible earnings for the store, given that it can receive no more than k negative reviews.

Input

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

The first line of each test case contains two integers n and k ($1 \le n \le 2 \cdot 10^5$; $0 \le k \le n$).

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 2 \cdot 10^9$).

The third line contains n integers $b_1, b_2, \dots, b_n$ ($1 \le b_i \le 2 \cdot 10^9$; $a_i < b_i$).

Additional constraint on the input: the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer — the maximum possible earnings for the store, given that it can receive no more than k negative reviews.

input
```
5
2 0
2 1
3 4
1 1
2
5
3 3
1 5 2
3 6 4
4 3
2 3 2 8
3 7 3 9
3 1
2 9 5
12 14 9
```
output
```
2
5
9
14
15
```

Consider the example from the statement:

- In the first test case, the price should be set to 1. Then both customers will buy one tree each and leave no negative reviews;
- In the second test case, the price should be set to 5. Then the only customer will buy a tree and leave a negative review;
- In the third test case, the price should be set to 3. Then all customers will buy one tree each, and the store will receive two negative reviews.
- In the fourth test case, the price should be set to 7. Then two customers will buy one tree each, and the store will receive one negative review.

F. Joker

2 seconds, 256 megabytes

Consider a deck of n cards. The positions in the deck are numbered from 1 to n from top to bottom. A joker is located at position m.

q operations are applied sequentially to the deck. During the i-th operation, you need to take the card at position a_i and move it either to the beginning or to the end of the deck. For example, if the deck is $[2, 1, 3, 5, 4]$, and $a_i = 2$, then after the operation the deck will be either $[1, 2, 3, 5, 4]$ (the card from the second position moved to the beginning) or $[2, 3, 5, 4, 1]$ (the card from the second position moved to the end).

Your task is to calculate the number of distinct positions where the joker can be after each operation.

Input

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

The first line of each test case contains three integers n, m, and q ($2 \le n \le 10^9$; $1 \le m \le n$; $1 \le q \le 2 \cdot 10^5$).

The second line contains q integers $a_1, a_2, \dots, a_q$ ($1 \le a_i \le n$).

Additional constraint on the input: the sum of q over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print q integers — the number of distinct positions where the joker can be after each operation.

input
```
5
6 5 3
1 2 3
2 1 4
2 1 1 2
5 3 1
3
3 2 4
2 1 1 1
18 15 4
13 15 1 16
```
output
```
2 3 5
2 2 2 2
2
2 3 3 3
2 4 6 8
```

G. Snakes

3 seconds, 512 megabytes

Suppose you play a game where the game field looks like a strip of 1×10^9 square cells, numbered from 1 to 10^9.

You have n snakes (numbered from 1 to n) you need to place into some cells. Initially, each snake occupies exactly one cell, and you can't place more than one snake into one cell. After that, the game starts.

The game lasts for q seconds. There are two types of events that may happen each second:

- snake s_i *enlarges*: if snake s_i occupied cells $[l, r]$, it enlarges to a segment $[l, r + 1]$;
- snake s_i *shrinks*: if snake s_i occupied cells $[l, r]$, it shrinks to a segment $[l + 1, r]$.

Each second, exactly one of the events happens.

If at any moment of time, any snake runs into some obstacle (either another snake or the end of the strip), you lose. Otherwise, you win with the score equal to the maximum cell occupied by any snake so far.

What is the minimum possible score you can achieve?

Input

The first line contains two integers n and q ($1 \le n \le 20$; $1 \le q \le 2 \cdot 10^5$) — the number of snakes and the number of events. Next q lines contain the description of events — one per line.

The i-th line contains

- either "s_i +" ($1 \le s_i \le n$) meaning that the s_i-th snake enlarges
- or "s_i -" ($1 \le s_i \le n$) meaning that the s_i-th snake shrinks.

Additional constraint on the input: the given sequence of events is valid, i. e. a snake of length 1 never shrinks.

Output

Print one integer — the minimum possible score.

input
```
3 6
1 +
1 -
3 +
3 -
2 +
2 -
```
output
```
4
```

input

```
5 13
5 +
3 +
5 -
2 +
4 +
3 +
5 +
5 -
2 +
3 -
3 +
3 -
2 +
```

output

```
11
```

In the first test, the optimal strategy is to place the second snake at cell 1, the third snake — at 2, and the first one — at 3. The maximum occupied cell is cell 4, and it's the minimum possible score.

In the second test, one of the optimal strategies is to place:

- snake 2 at position 1;
- snake 3 at position 4;
- snake 5 at position 6;
- snake 1 at position 9;
- snake 4 at position 10.