A. Fast Food Restaurant

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

Tired of boring office work, Denis decided to open a fast food restaurant.

On the first day he made a portions of dumplings, b portions of cranberry juice and c pancakes with condensed milk.

The peculiarity of Denis's restaurant is the procedure of ordering food. For each visitor Denis himself chooses a set of dishes that this visitor will receive. When doing so, Denis is guided by the following rules:

- every visitor should receive at least one dish (dumplings, cranberry juice, pancakes with condensed milk are all considered to be dishes);
- each visitor should receive no more than one portion of dumplings, no more than one portion of cranberry juice and no more than one pancake with condensed milk;
- all visitors should receive different sets of dishes.

What is the maximum number of visitors Denis can feed?

Input

The first line contains an integer t $(1 \le t \le 500)$ — the number of test cases to solve.

Each of the remaining t lines contains integers a, b and c $(0 \le a, b, c \le 10)$ — the number of portions of dumplings, the number of portions of cranberry juice and the number of condensed milk pancakes Denis made.

Output

For each test case print a single integer — the maximum number of visitors Denis can feed.

Example

input

```
7
1 2 1
0 0 0
9 1 7
2 2 3
2 3 2
3 2 2
4 4 4
```

output

```
3
0
4
5
5
5
7
```

Note

In the first test case of the example, Denis can feed the first visitor with dumplings, give the second a portion of cranberry juice, and give the third visitor a portion of cranberry juice and a pancake with a condensed milk.

In the second test case of the example, the restaurant Denis is not very promising: he can serve no customers.

In the third test case of the example, Denise can serve four visitors. The first guest will receive a full lunch of dumplings, a portion of cranberry juice and a pancake with condensed milk. The second visitor will get only dumplings. The third guest will receive a pancake with condensed milk, and the fourth guest will receive a pancake and a portion of dumplings. Please note that Denis hasn't used all of the prepared products, but is unable to serve more visitors.

B. Different Rules

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

Nikolay has only recently started in competitive programming, but already qualified to the finals of one prestigious olympiad. There going to be n participants, one of whom is Nikolay. Like any good olympiad, it consists of two rounds. Tired of the traditional rules, in which the participant who solved the largest number of problems wins, the organizers came up with different rules.

Suppose in the first round participant A took x-th place and in the second round — y-th place. Then the total score of the participant A is sum $x + y$. The overall place of the participant A is the number of participants (including A) having their total score less than or equal to the total score of A. Note, that some participants may end up having a common overall place. It is also important to note, that in both the first and the second round there were no two participants tying at a common place. In other words, for every i from 1 to n **exactly one** participant took i-th place in first round and **exactly one** participant took i-th place in second round.

Right after the end of the Olympiad, Nikolay was informed that he got x-th place in first round and y-th place in the second round. Nikolay doesn't know the results of other participants, yet he wonders what is the minimum and maximum place he can take, if we consider the most favorable and unfavorable outcome for him. Please help Nikolay to find the answer to this question.

Input

The first line contains an integer t $(1 \le t \le 100)$ — the number of test cases to solve.

Each of the following t lines contains integers n, x, y ($1 \le n \le 10^9$, $1 \le x, y \le n$) — the number of participants in the olympiad, the place that Nikolay took in the first round and the place that Nikolay took in the second round.

Output

Print two integers — the minimum and maximum possible overall place Nikolay could take.

Examples

input
```
1
5 1 3
```
output
```
1 3
```

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

Note

Explanation for the first example:

Suppose there were 5 participants A-E. Let's denote Nikolay as A. The the most favorable results for Nikolay could look as follows:

Participant	Round 1	Round 2	Total score	Plac
A	1	3	4	1
B	2	4	6	3
C	3	5	8	5
D	4	1	5	2
E	5	2	7	4

However, the results of the Olympiad could also look like this:

Participant	Round 1	Round 2	Total score	Plac
A	1	3	4	3
B	2	2	4	3
C	3	1	4	3
D	4	4	8	4
E	5	5	10	5

In the first case Nikolay would have taken first place, and in the second — third place.

C1. Skyscrapers (easy version)

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

This is an easier version of the problem. In this version $n \le 1000$

The outskirts of the capital are being actively built up in Berland. The company "Kernel Panic" manages the construction of a residential complex of skyscrapers in New Berlskva. All skyscrapers are built along the highway. It is known that the company has already bought n plots along the highway and is preparing to build n skyscrapers, one skyscraper per plot.

Architects must consider several requirements when planning a skyscraper. Firstly, since the land on each plot has different properties, each skyscraper has a limit on the largest number of floors it can have. Secondly, according to the design code of the city, it is unacceptable for a skyscraper to simultaneously have higher skyscrapers both to the left and to the right of it.

Formally, let's number the plots from 1 to n. Then if the skyscraper on the i-th plot has a_i floors, it must hold that a_i is at most m_i ($1 \le a_i \le m_i$). Also there mustn't be integers j and k such that $j < i < k$ and $a_j > a_i < a_k$. Plots j and k are **not** required to be adjacent to i.

The company wants the total number of floors in the built skyscrapers to be as large as possible. Help it to choose the number of floors for each skyscraper in an optimal way, i.e. in such a way that all requirements are fulfilled, and among all such construction plans choose any plan with the maximum possible total number of floors.

Input

The first line contains a single integer n ($1 \leq n \leq 1000$) — the number of plots.

The second line contains the integers $m_1, m_2, \ldots, m_n$ ($1 \leq m_i \leq 10^9$) — the limit on the number of floors for every possible number of floors for a skyscraper on each plot.

Output

Print n integers a_i — the number of floors in the plan for each skyscraper, such that all requirements are met, and the total number of floors in all skyscrapers is the maximum possible.

If there are multiple answers possible, print any of them.

Examples

input

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

output

```
1 2 3 2 1
```

input

```
3
10 6 8
```

output

```
10 6 6
```

Note

In the first example, you can build all skyscrapers with the highest possible height.

In the second test example, you cannot give the maximum height to all skyscrapers as this violates the design code restriction. The answer $[10, 6, 6]$ is optimal. Note that the answer of $[6, 6, 8]$ also satisfies all restrictions, but is not optimal.

C2. Skyscrapers (hard version)

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

This is a harder version of the problem. In this version $n \leq 500\,000$

The outskirts of the capital are being actively built up in Berland. The company "Kernel Panic" manages the construction of a residential complex of skyscrapers in New Berlskva. All skyscrapers are built along the highway. It is known that the company has already bought n plots along the highway and is preparing to build n skyscrapers, one skyscraper per plot.

Architects must consider several requirements when planning a skyscraper. Firstly, since the land on each plot has different properties, each skyscraper has a limit on the largest number of floors it can have. Secondly, according to the design code of the city, it is unacceptable for a skyscraper to simultaneously have higher skyscrapers both to the left and to the right of it.

Formally, let's number the plots from 1 to n. Then if the skyscraper on the i-th plot has a_i floors, it must hold that a_i is at most m_i ($1 \leq a_i \leq m_i$). Also there mustn't be integers j and k such that $j < i < k$ and $a_j > a_i < a_k$. Plots j and k are **not** required to be adjacent to i.

The company wants the total number of floors in the built skyscrapers to be as large as possible. Help it to choose the number of floors for each skyscraper in an optimal way, i.e. in such a way that all requirements are fulfilled, and among all such construction plans choose any plan with the maximum possible total number of floors.

Input

The first line contains a single integer n ($1 \leq n \leq 500\,000$) — the number of plots.

The second line contains the integers $m_1, m_2, \ldots, m_n$ ($1 \leq m_i \leq 10^9$) — the limit on the number of floors for every possible number of floors for a skyscraper on each plot.

Output

Print n integers a_i — the number of floors in the plan for each skyscraper, such that all requirements are met, and the total number of floors in all skyscrapers is the maximum possible.

If there are multiple answers possible, print any of them.

Examples

input

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

output

```
1 2 3 2 1
```

input

```
3
10 6 8
```

output

```
10 6 6
```

Note

In the first example, you can build all skyscrapers with the highest possible height.

In the second test example, you cannot give the maximum height to all skyscrapers as this violates the design code restriction. The answer $[10, 6, 6]$ is optimal. Note that the answer of $[6, 6, 8]$ also satisfies all restrictions, but is not optimal.

D. Happy New Year

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Being Santa Claus is very difficult. Sometimes you have to deal with difficult situations.

Today Santa Claus came to the holiday and there were m children lined up in front of him. Let's number them from 1 to m. Grandfather Frost knows n spells. The i-th spell gives a candy to every child whose place is in the $[L_i, R_i]$ range. Each spell can be used at most once. It is also known that if all spells are used, each child will receive at most k candies.

It is not good for children to eat a lot of sweets, so each child can eat no more than one candy, while the remaining candies will be equally divided between his (or her) Mom and Dad. So it turns out that if a child would be given an even amount of candies (possibly zero), then he (or she) will be unable to eat any candies and will go sad. However, the rest of the children (who received an odd number of candies) will be happy.

Help Santa Claus to know the maximum number of children he can make happy by casting some of his spells.

Input

The first line contains three integers of n, m, and k ($1 \leq n \leq 100\,000, 1 \leq m \leq 10^9, 1 \leq k \leq 8$) — the number of spells, the number of children and the upper limit on the number of candy a child can get if all spells are used, respectively.

This is followed by n lines, each containing integers L_i and R_i ($1 \leq L_i \leq R_i \leq m$) — the parameters of the i spell.

Output

Print a single integer — the maximum number of children that Santa can make happy.

Example

input

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

output

```
4
```

Note

In the first example, Santa should apply the first and third spell. In this case all children will be happy except the third.

E. Concatenation with intersection

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Vasya had three strings a, b and s, which consist of lowercase English letters. The lengths of strings a and b are equal to n, the length of the string s is equal to m.

Vasya decided to choose a substring of the string a, then choose a substring of the string b and concatenate them. Formally, he chooses a segment $[l_1, r_1]$ ($1 \leq l_1 \leq r_1 \leq n$) and a segment $[l_2, r_2]$ ($1 \leq l_2 \leq r_2 \leq n$), and after concatenation he obtains a string $a[l_1, r_1] + b[l_2, r_2] = a_{l_1} a_{l_1+1} \ldots a_{r_1} b_{l_2} b_{l_2+1} \ldots b_{r_2}$.

Now, Vasya is interested in counting number of ways to choose those segments adhering to the following conditions:

- segments $[l_1, r_1]$ and $[l_2, r_2]$ have non-empty intersection, i.e. there exists at least one integer x, such that $l_1 \leq x \leq r_1$ and $l_2 \leq x \leq r_2$;
- the string $a[l_1, r_1] + b[l_2, r_2]$ is equal to the string s.

Input

The first line contains integers n and m ($1 \leq n \leq 500\,000, 2 \leq m \leq 2 \cdot n$) — the length of strings a and b and the length of the string s.

The next three lines contain strings a, b and s, respectively. The length of the strings a and b is n, while the length of the string s is m.

All strings consist of lowercase English letters.

Output

Print one integer — the number of ways to choose a pair of segments, which satisfy Vasya's conditions.

Examples

input

```
6 5
aabbaa
baaaab
aaaaa
```

output

```
4
```

input

```
5 4
azaza
zazaz
azaz
```

output

```
11
```

input

```
9 12
abcabcabc
xyzxyzxyz
abcabcayzxyz
```

output

```
2
```

Note

Let's list all the pairs of segments that Vasya could choose in the first example:

1. $[2, 2]$ and $[2, 5]$;
2. $[1, 2]$ and $[2, 4]$;
3. $[5, 5]$ and $[2, 5]$;
4. $[5, 6]$ and $[3, 5]$;