ICPC SOUTH PACIFIC REGIONAL FINALS

AUGUST 31, 2024

Contest Problems

Sponsors

wisetech global

imc TRADING

Jane Street

This contest contains twelve problems. Good luck.

For problems that state "*Your answer should have an absolute or relative error of less than* 10^{-9}", your answer, x, will be compared to the correct answer, y. If $|x - y| < 10^{-9}$ or $\frac{|x-y|}{|y|} < 10^{-9}$, then your answer will be considered correct.

Definition 1

For problems that ask for a result modulo m:
If the correct answer to the problem is the integer b, then you should display the unique value a such that:

- $0 \leq a < m$
  and
- $(a - b)$ is a multiple of m.

Definition 2

A string $s_1 s_2 \cdots s_n$ is lexicographically smaller than $t_1 t_2 \cdots t_\ell$ if

- there exists $k \leq \min(n, \ell)$ such that $s_i = t_i$ for all $1 \leq i < k$ and $s_k < t_k$
  or
- $s_i = t_i$ for all $1 \leq i \leq \min(n, \ell)$ and $n < \ell$.

Definition 3

- Uppercase letters are the uppercase English letters $(A, B, \ldots, Z)$.
- Lowercase letters are the lowercase English letters $(a, b, \ldots, z)$.

Definition 4

Unless otherwise specified, the distance between two points (x_0, y_0) and (x_1, y_1) is defined as its Euclidean distance:

$$\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}.$$

This page is intentionally left (almost) blank.

Problem A
Alphabet Hunters
Time limit: 60 seconds

In the Kingdom of CodeLand, there lies a magical text guarded by 22 Alphabet Hunters—each representing a different lowercase letter of the alphabet (a–v). The Kingdom's scholars recently discovered ancient patterns—powerful strings that hold the key to great treasure. Each pattern has its own weight, which could be either a blessing or a curse.

Your task is to help the Kingdom's scholars select a powerful subset of Alphabet Hunters. When a subset is chosen, the magical text will transform: only the letters in this subset will remain, while the rest will disappear, splitting the text into separate substrings.

For each discovered pattern, count how many times it appears in all the resulting substrings, and multiply this count by its corresponding weight. The total score of a subset is the sum of the weighted counts of all patterns.

For example, consider the magical text aaabcabc and 5 patterns with their corresponding weights:

$(\texttt{a}, -1), (\texttt{aa}, 2), (\texttt{b}, -1), (\texttt{c}, -1), (\texttt{abc}, 4)$.

The eight relevant subsets give the following scores:

		a		aa		b		c		abc		
{a, b, c}	aaabcabc	4×-1	+	2×2	+	2×-1	+	2×-1	+	2×4	=	4
{a, b}	aaab_ab_	4×-1	+	2×2	+	2×-1	+	0×-1	+	0×4	=	−2
{a, c}	aaa_ca_c	4×-1	+	2×2	+	0×-1	+	2×-1	+	0×4	=	−2
{b, c}	___bc_bc	0×-1	+	0×2	+	2×-1	+	2×-1	+	0×4	=	−4
{a}	aaa__a__	4×-1	+	2×2	+	0×-1	+	0×-1	+	0×4	=	0
{b}	___b__b_	0×-1	+	0×2	+	2×-1	+	0×-1	+	0×4	=	−2
{c}	____c__c	0×-1	+	0×2	+	0×-1	+	2×-1	+	0×4	=	−2
{}	________	0×-1	+	0×2	+	0×-1	+	0×-1	+	0×4	=	0

Taking the subset {a, b, c} gives a score of 4, which is the maximum. Note that the subset {a, b, c, f, q} also gives a score of 4, but no subset produces a score larger than 4.

Time is of the essence, and your code must be swift and efficient to find the most powerful subset. The scholars need you to find the maximum score that can be attained by a subset of letters.

Input

The first line contains two integers n ($1 \le n \le 300\,000$) and m ($0 \le m \le 300\,000$), which are the length of the text and the number of patterns, respectively.

The second line contains a string of length n, consisting only of lowercase letters between a and v, inclusive.

Each of the next m lines contains a pattern p_i ($1 \le \text{length}(p_i) \le 300\,000$) and an integer w_i ($-10\,000\,000 \le w_i \le 10\,000\,000$), representing the i-th pattern and its associated weight.

The sum of the lengths of all patterns does not exceed $300\,000$. All patterns consist only of lowercase letters between a and v, inclusive.

Output

Display the maximum score that can be attained by a subset of letters.

Sample Input 1

```
5 3
ababc
ab 1
abc -3
c 4
```

Sample Output 1

```
4
```

Sample Input 2

```
8 5
aaabcabc
a -1
aa 2
b -1
c -1
abc 4
```

Sample Output 2

```
4
```

Sample Input 3

```
1 3
a
a 100
a 10
a -1
```

Sample Output 3

```
109
```

Problem B
Buying Newfriendships
Time limit: 60 seconds

You may be familiar with the concept of "friendship." Friendship is a social relationship between two people comprising mutual trust, support, and care. A person in the friendship is commonly called a friend of the other, and historically speaking, these relationships were generally bidirectional.

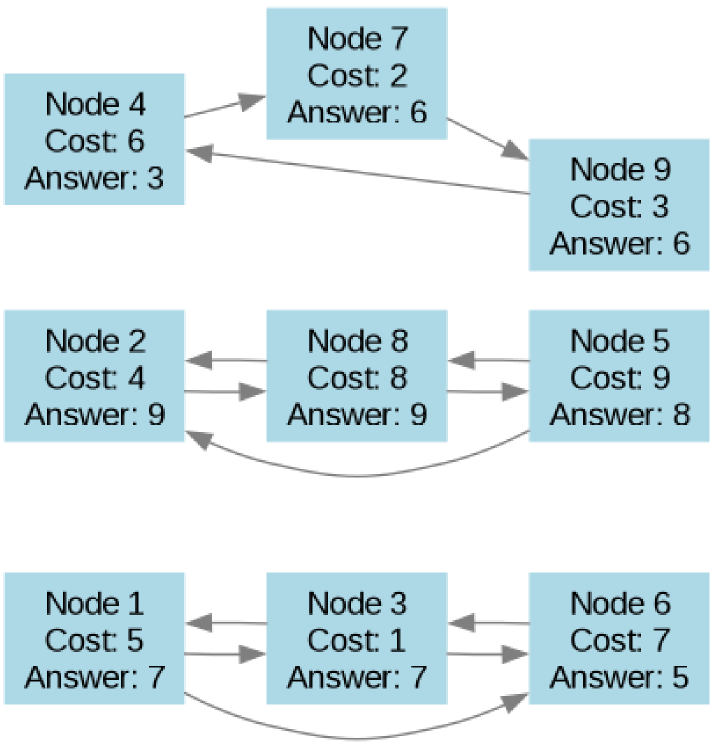
You may be less familiar with the concept of "newfriendship." Newfriendship is a parasocial relationship between two people and is notably different from friendship in that it is not necessarily bidirectional. That is, someone may not be the newfriend of their newfriend.

An interesting property of the parasocial newfriendship is that a person will always become newfriends with their newfriends' newfriends. For example, if ja$on is newfriends with Iggy, and Iggy is newfriends with Eliot, then ja$on will become newfriends with Eliot. This will repeat, so after ja$on becomes newfriends with Eliot, he will become newfriends with any of Eliot's newfriends, and so on. Of course, Eliot will not become newfriends with ja$on and probably doesn't even know he exists.

This is complicated by the fact that newfriendships have a recurring cost. Each person i has a recurring cost c_i needed to be newfriends with them. It's possible for a person to be newfriends with themselves, but in this case, they don't need to pay the recurring cost.

Max has taken an interest in analysing newfriendships and wants to know the largest recurring cost each person will end up paying out of each of their newfriends once they have become newfriends with everyone they can.

The following graph describes sample input #4 and its answer:



Given everyone's recurring cost and all existing newfriendships, what is the maximum recurring cost that each person will pay for some newfriendship after all newfriendships are created?

Input

The first line contains two integers N $(1 \leq N \leq 200\,000)$ and M $(0 \leq M \leq 200\,000)$, denoting that there are N people and M newfriendships.

Following are N lines, each containing a single integer c_i ($0 \leq c_i \leq 10^9$), denoting the recurring cost of being newfriends with person i.

Following are M lines, each containing two integers a and b ($1 \leq a, b \leq N$ and $a \neq b$), denoting that person a is newfriends with person b. The same newfriendship will not be listed more than once.

Output

For each person, in order, display the largest recurring cost out of all of their newfriends. If a person has no newfriends, display 0.

Sample Input 1

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

Sample Output 1

```
2
1
0
```

Sample Input 2

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

Sample Output 2

```
4
3
3
3
0
```

Sample Input 3

```
1 0
1
```

Sample Output 3

```
0
```

Sample Input 4	Sample Output 4
9 13 5 4 1 6 9 7 2 8 3 5 2 7 9 4 7 8 2 5 8 3 6 3 1 6 3 1 6 2 8 9 4 1 3 8 5	7 9 7 3 8 5 6 9 6

This page is intentionally left (almost) blank.

Problem C
Cursor Navigation
Time limit: 60 seconds

You are using an unusual text editor called "Screamacs" to edit a parenthetical sequence p, which consists of only open and close parentheses.

A valid parenthetical sequence is a string of even length comprising only '(' and ')' characters. Each '(' is unambiguously paired to a ')', and so the number of '(' and ')' characters is equal. For example, "((()))()" is valid, but "(()))" is not, and neither is "())(". Interestingly, "())(" is not valid, but it is a palindrome.

The Screamacs editor provides some convenient tools for navigating such a sequence. When your cursor is on a particular character in the sequence, you have three operations for moving.

1. You can move the cursor **left** to the preceding character (if you are not at the first character),
2. You can move the cursor **right** to the following character (if you are not at the last character), and
3. You can **jump** to the matching parenthetical pair.

For example, consider p = `(()())`. If you are at position 2, `(()())`, then you can move left to get to position 1, `(()())`, then jump to position 6, `(()())`.

You are curious about how many ways there are for you to get from your start position s in p to a target position t. A "way" is a sequence of operations. Two sequences are considered different if they are different lengths or have a different operation for at least one position. Since you get bored easily, you will never take a sequence of operations that visits a position more than once, including s and t.

How many ways are there from s to t?

Input

The first line of input contains a single even integer, n ($2 \leq n \leq 200\,000$), which is the length of the parenthetical sequence.

The second line of input contains two integers, s and t ($1 \leq s, t \leq n$), which are the start and target positions, respectively.

The third line of input contains the valid parenthetical sequence p. The sequence p is comprised only of the characters '(' and ')' and is of length n.

Output

Display the number of ways modulo 1 000 000 007.

Sample Input 1	Sample Output 1
2 1 2 ()	2

Sample Input 2	Sample Output 2
4 3 2 (())	3

Sample Input 3	Sample Output 3
2 2 2 ()	1

Sample Input 4	Sample Output 4
6 1 5 (()())	5

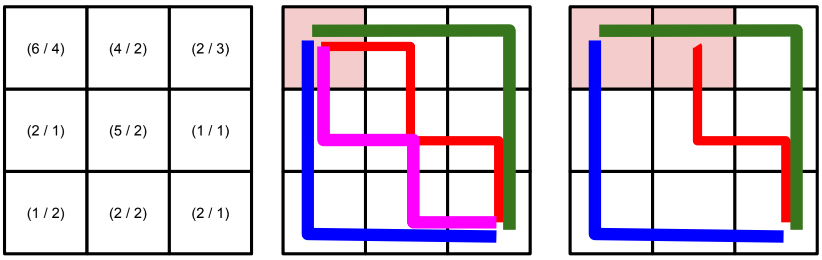
Problem D
Duplicated Decontamination
Time limit: 60 seconds

Mo is a robot tasked with cleaning a house laid out in an $n \times n$ grid of rooms. Starting in the north-west room, Mo's goal is to clean every room in the house. However, Mo can only move south or east, which means that cleaning all the rooms is impossible (except when $n = 1$).

Fortunately, some rooms are equipped with cloning stations that allow Mo to create exact duplicates of itself. These duplicates operate independently and can also use cloning stations to produce more copies of Mo. Like Mo, these duplicates are restricted to moving only south and east.

Each cloning station has a usage cost and a maximum number of times it can be used. Your task is to determine whether it is feasible for Mo (and all its clones) to clean every room in the house, and if it is, what the minimum total cost is to clean the house.

For example, consider the following house. In example A, Mo makes 3 copies of itself (costing $6 \times 3 = 18$), then the 4 robots clean the house following the paths shown. Example B also cleans all rooms, but for a lower price. Mo copies itself in the north-west room (costing 6), then the clone makes another copy in the north-middle room (costing an additional 4), giving a total of 10 cost. An even cheaper option is to do the second clone in the middle-west room, for a total cost of 8.



Legend
(Cost / Usage Limit)

Example A:
Make 3 copies in top-left corner

Example B:
Make 1 copy in top-left corner and 1 copy in the "(4 / 2)" cell.

In the example, if the usage limit in the north-west room was 2 instead of 4, then Example A would be invalid, since the cloning station was used 3 times.

Given the locations of all cloning stations, what is the minimum cost to clean every room in the house?

Input

The first line of input contains two integers n ($1 \leq n \leq 200\,000$), which is the size of the house, and S ($1 \leq S \leq 200\,000$), which is the number of cloning stations.

The next S lines describe the cloning stations. Each of these lines contains four integers r ($1 \leq r \leq n$), which is the row of this cloning station, c ($1 \leq c \leq n$), which is the column of this cloning station, P ($1 \leq P \leq 10^9$), which is the cost for each use of this cloning station, and L ($1 \leq L \leq 200\,000$), which is the maximum number of times this cloning station may be used.

The north-west corner is at $(1, 1)$, and the south-east corner is at (n, n). There is at most one cloning station in each room.

Output

If it is not possible to clean every room, display `Impossible`. Otherwise, display `Possible`, followed by a single integer M, which is the minimum cost to clean every room.

Sample Input 1	Sample Output 1
3 9 1 1 6 4 1 2 4 2 1 3 2 3 2 1 2 1 2 2 5 2 2 3 1 1 3 1 1 2 3 2 2 2 3 3 2 1	Possible 8

Sample Input 2	Sample Output 2
3 2 1 2 1 10 3 1 4 1	Impossible

Sample Input 3	Sample Output 3
10 1 1 1 1000000000 20	Possible 9000000000

Sample Input 4	Sample Output 4
1 1 1 1 1 1	Possible 0

Sample Input 5	Sample Output 5
3 2 1 1 4 1 1 2 10 1	Possible 14

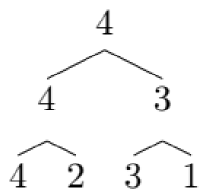
Problem E
Early Bird
Time limit: 60 seconds

The Society for the Production of Pristine Code (SPPC) has organized a tournament in which 2^n participants compete against each other over the course of n rounds. In each round, competitors are paired up, and only one member of each pair advances to the next round until a single winner emerges.

To obtain a complete ranking, i.e. who comes in 2nd place, 3rd place, etc., the judges decide to use two factors: who won in matchups and the order of registration, to encourage early signup. Competitor A is said to have *thrashed* competitor B if either A won against B directly or if A won against some competitor directly that thrashed B. The order of registration was rewarded by the following rule:

Early Bird Rule: If A signed up earlier than B and B did not thrash A, then A must have a better rank than B.

However, right after the tournament, the judges noticed a big problem: their new rule could force them to assign competitor B a better rank than competitor A even though A thrashed B. Consider the following tournament tree, where competitors are numbered in their order of signup (1 signed up first and 4 signed up last).



Since competitor 1 signed up before 2 and was not thrashed by 2, 1 must have a better rank than 2. Similarly, 2 must have a better rank than 3. Since 1 has a better rank than 2 and 2 has a better rank than 3, it follows that 1 must have a better rank than 3. This transitive application of the Early Bird Rule forces 1 to have a better rank than 3, even though 3 thrashed 1.

And even worse, the Early Bird Rule says nothing about where to rank competitor 4! Since they signed up last and thrashed all other contestants, the Early Bird Rule does not say anything about how the judges should rank them. So [4, 1, 2, 3], [1, 4, 2, 3], [1, 2, 4, 3], and [1, 2, 3, 4] are all valid rankings just considering the Early Bird Rule.

In an effort to fix this, the judges quickly add the following rule:

Common Sense Rule: If A thrashed B and the Early Bird Rule does not force B to have a better rank than A (possibly transitively), then A must have a better rank than B.

Note that if the Early Bird Rule forces competitor A to have a better rank than competitor B, then the Common Sense Rule will not change that.

	1	2	3	4
1	–	**1 ranked better** By Early Bird Rule	**1 ranked better** By Early Bird Rule (1 > 2 > 3)	**4 ranked better** By Common Sense Rule
2	**1 ranked better** By Early Bird Rule	–	**2 ranked better** By Early Bird Rule	**4 ranked better** By Common Sense Rule
3	**1 ranked better** By Early Bird Rule (1 > 2 > 3)	**2 ranked better** By Early Bird Rule	–	**4 ranked better** By Common Sense Rule
4	**4 ranked better** By Common Sense Rule	**4 ranked better** By Common Sense Rule	**4 ranked better** By Common Sense Rule	–

It can be proven that there is exactly one valid ranking that satisfies both rules. Help the judges determine the final ranking using the Early Bird Rule and the Common Sense Rule.

Input

The first line of the input contains a single integer n ($1 \leq n \leq 16$), the number of rounds the tournament has.

The second line contains a permutation of $1, \ldots, 2^n$, describing the bottom row of the tournament tree, which has been ordered so that the winner of a round is always the competitor listed first.

Output

Display the final ranking, in order from best to worst.

Sample Input 1

```
2
4 2 3 1
```

Sample Output 1

```
4 1 2 3
```

Sample Input 2

```
3
6 1 7 5 4 8 3 2
```

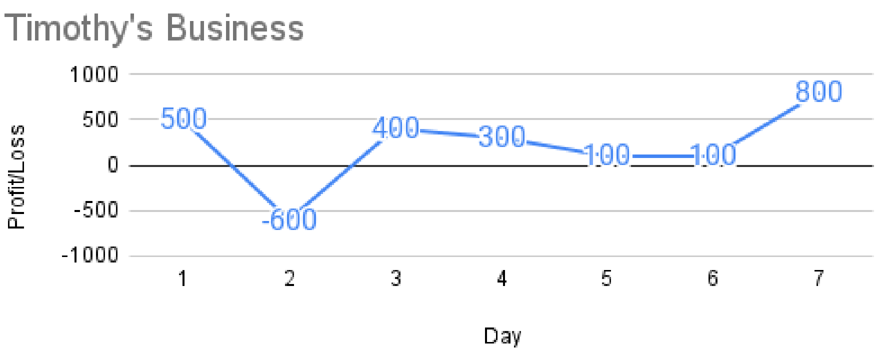
Sample Output 2

```
6 1 4 3 2 7 5 8
```

Problem F
Faking Business Growth
Time limit: 60 seconds

Timothy recently started a small printing business with his friend. So far, his business has been open for n days and he has recorded his net profit/loss for each day.

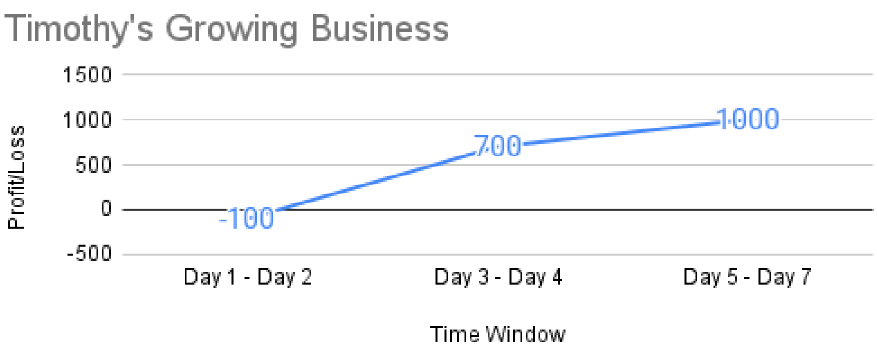
Timothy really wants to show off to his friends by creating a chart showing that his business has a positive trend in profitability. To do this, he initially made a chart with the date on the x-axis and the profit/loss for each day on the y-axis.



Sadly, his business is not taking off quite as he planned. But, all is not lost! Timothy realized that he could modify his chart by combining groups of consecutive dates into a single time window. The profit/loss of a time window is the sum of the profit/loss for all dates within that range. Doing this correctly will let him create a chart where the profit/loss for each time window is strictly increasing. Each day must be in exactly one of the time windows.

For example, suppose that Timothy's business has operated for 7 days and his profits/losses for the days are $[500, -600, 400, 300, 100, 100, 800]$. He could create the following time windows:

- Day 1 – 2: $500 + (-600) = -100$
- Day 3 – 4: $400 + 300 = 700$
- Day 5 – 7: $100 + 100 + 800 = 1000$



Now, his chart shows that his business's profit/loss is strictly increasing. To avoid raising suspicion, Timothy wants to partition his dates into as many time windows as possible to let his chart show a lot of data.

In the example above, Timothy could have instead used 4 time windows (Day 1 – 2, Day 3, Day 4 – 6, Day 7).

What is the largest number of time windows that Timothy can partition his profits/losses into such that his friends will be impressed by his 'growing' business?

Input

The first line of input contains an integer n ($2 \leq n \leq 365$), which is the number of days that Timothy's business has been open for.

The second line contains n integers $p_1, p_2, \ldots, p_n$ ($-10^9 \leq p_i \leq 10^9$), which are the profit/loss numbers for each day.

Output

Display the maximum number of time windows Timothy can partition his transactions into while still impressing his friends.

Sample Input 1	Sample Output 1
7 500 -600 400 300 100 100 800	4

Sample Input 2	Sample Output 2
10 10 9 8 7 6 5 4 3 2 1	3

Sample Input 3	Sample Output 3
6 100 -1000 50 -800 10 -400	3

Sample Input 4	Sample Output 4
2 -1 -1	1

Problem G
Greenest Blue
Time limit: 60 seconds

John has a colour spectrum from 0 to 10^9, where 0 is pure blue and 10^9 is pure green. All colours between pure blue and pure green are some combination of blue and green. If you have two colours, say x and y, where $x < y$, then x is 'more blue' than y and y is 'more green' than x.

John wishes to know what the *greenest blue* is for him. Ideally, there would be some colour-value where every colour-value greater than it is green and every colour-value less than or equal to it is blue. Every day, John looks at a colour-value and states whether he thinks it is blue or green. However, John isn't always consistent — for example, he may say that 1000 is blue one day, and that 800 is green the next day.

The greenest blue, B, is defined as the largest colour-value that John said is blue and all colour-values he said are green are strictly larger than B.

John will always consider 0 to be blue and 10^9 to be green. Given a list of colour-values and whether John thinks each is blue or green, what is the greenest blue?

Input

The first line of input contains a single integer n ($1 \leq n \leq 500\,000$), which is the number of colour-values.

The next n lines describe what colour John thinks each colour-value is. Each line contains an integer v ($0 < v < 10^9$), which is the colour-value, and a letter c ($c \in \{\texttt{B}, \texttt{G}\}$), which is whether John thinks v is blue or green.

Output

Display the colour-value of the greenest blue.

Sample Input 1	Sample Output 1
3 10 B 12 G 15 B	10

Sample Input 2	Sample Output 2
5 300 B 400 B 100 B 200 B 500 G	400

Sample Input 3	Sample Output 3
2 5 G 5 B	0

This page is intentionally left (almost) blank.

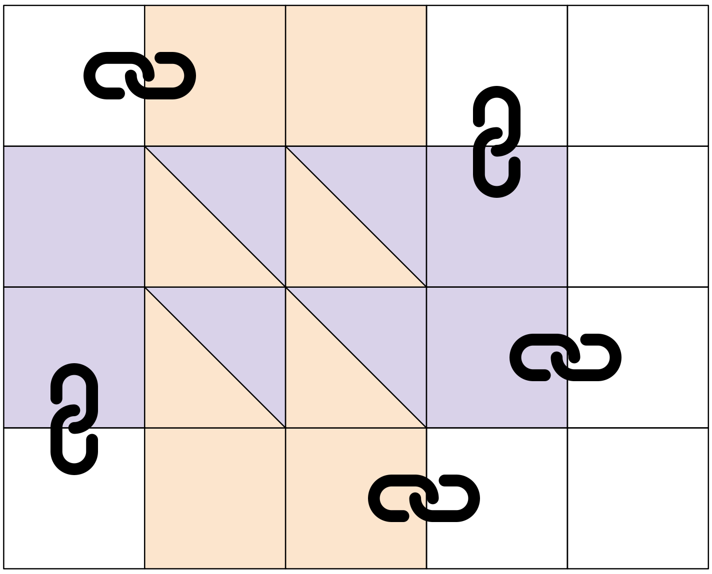
Problem H
Housing Permit
Time limit: 60 seconds

Eliot is planning his dream house. It's going to have several stories, including a basement with an atomic force microscope for experiments. Eliot is a very private person and will not allow any of his friends to visit his new house. He is the only person allowed past the threshold. However, he still wants to ensure that his house is as large as possible.

The city block that Eliot is looking to buy property on can be modeled as an $R \times C$ grid of square cells. Eliot can select any non-empty contiguous rectangle to build his house on. That means the height and width may be different. However, there are some restrictions. The council wants to encourage friendship, so there are restrictions on any planned houses whose owners do not allow friends to visit their house. Unfortunately, Eliot falls into this category.

Each restriction involves two adjacent cells. Two cells are adjacent if they share an edge (not just a corner). So a cell can have up to 4 adjacent neighbouring cells. Each restriction prevents both of the adjacent cells from being in a house together. For example, suppose that a restriction involves (r, c) and $(r+1, c)$, then either neither of these can be in Eliot's house or exactly one of them, but never both. A rectangle that is fully contained in the grid and satisfies all restrictions is called a *house rectangle*.

In the example below, there are 5 restrictions (3 horizontal and 2 vertical). The largest house rectangle is 8 units large and there are two possible locations for it.



Help Eliot determine the area of the largest house rectangle. Also, since Eliot likes to keep his options open, determine how many house rectangles have the largest possible area. Two house rectangles are considered different if at least one cell is in one but not the other.

Input

The first line of input contains two integers R ($1 \leq R \leq 200\,000$), which is the number of rows, and C ($1 \leq C \leq 200\,000$ and $R \times C \leq 200\,000$), which is the number of columns.

The next line contains a single integer H ($0 \leq H \leq R \times (C-1)$), which is the number of horizontal restrictions.

The next H lines describe the horizontal restrictions. Each of these lines contains two integers r ($1 \leq r \leq R$) and c ($1 \leq c \leq C$), which indicates there is a restriction on the cells (r, c) and $(r, c+1)$.

The next line contains a single integer V ($0 \leq V \leq (R-1) \times C$), which is the number of vertical restrictions.

The next V lines describe the vertical restrictions. Each of these lines contains two integers r ($1 \leq r \leq R-1$) and c ($1 \leq c \leq C$), which indicates there is a restriction on the cells (r, c) and $(r+1, c)$.

No two restrictions represent the same set of cells.

Output

Display the area of the largest house rectangle and the number of house rectangles that have the largest possible area.

Sample Input 1	Sample Output 1
4 5 3 1 1 3 4 4 3 2 1 4 3 1	8 2

Sample Input 2	Sample Output 2
2 2 2 1 1 2 1 2 1 1 1 2	1 4

Sample Input 3	Sample Output 3
5 5 4 3 1 3 4 3 2 3 3 4 1 3 4 3 2 3 3 3	5 8

Sample Input 4	Sample Output 4
3 5 0 0	15 1

Sample Input 5	Sample Output 5
3 3 1 1 1 1 1 1	6 2

Sample Input 6	Sample Output 6
2 4 1 2 1 2 1 3 1 4	4 1

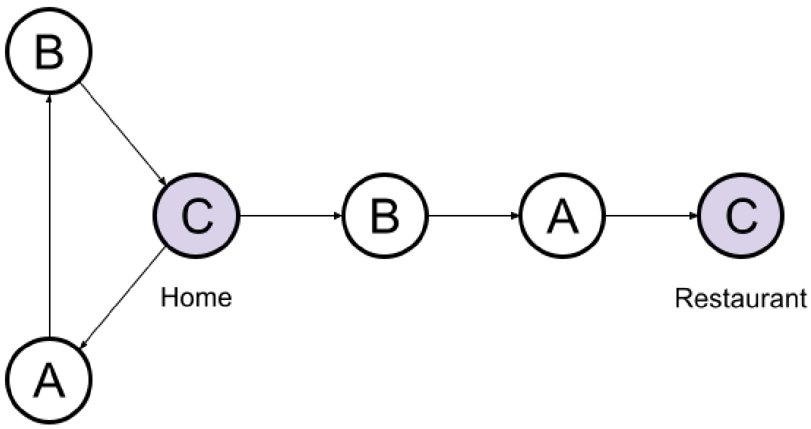
This page is intentionally left (almost) blank.

Problem I
Iggy's Palindromic Walk
Time limit: 60 seconds

The legendary iguanas Izzy and Iggy have been married for almost two years now! Izzy said that as an anniversary gift, she wants a really nice palindrome. A palindrome is a string that is the same forwards and backwards. In the preliminary contest, we helped Iggy create a short palindrome for Izzy, but Iggy wants to do more: along with this gift, he wants to take Izzy on a romantic walk on their way to a restaurant: a *palindromic walk*.

The town they are in consists of many locations that are connected by a collection of one-way roads. Each location has a name. Iggy would like to start at their home and walk along a series of roads that take them to the restaurant. Each time they get to a location (including when they start at home), they write down the first letter of the location's name. When they finish their walk at the restaurant, the letters they wrote down should be a palindrome.

For example, in the following city, they could start at their home and walk the left triangle first before walking to the restaurant, making a palindrome of `CABCBAC`.



On the walk, locations and roads may be used multiple times (including their home and the restaurant), so long as they start at their home and end at the restaurant.

Given the layout of the city, find a palindromic walk.

Input

The input starts with a line containing two integers n ($2 \le n \le 100$), which is the number of locations, and m ($1 \le m \le n(n-1)$), which is the number of one-way roads. Iggy and Izzy's home is at location 1, and the restaurant is at location n.

The second line contains n uppercase letters, $c_1, c_2, \ldots, c_n$, which are the first letters in each of the n locations.

The next m lines describe the roads. Each line contains two integers s ($1 \le s \le n$) and t ($1 \le t \le n$ and $s \ne t$) indicating that there is a one-way road from location s to location t. No two roads are the same.

Output

If it is impossible to make a palindromic walk, display `Impossible`.

Otherwise, display `Possible`. Then display an integer k, which is the length of the walk, followed by k integers $a_1, a_2, \ldots, a_k$ ($1 \le a_i \le n$), which are the locations visited, in order. You must start at home ($a_1 = 1$) and end at the restaurant ($a_k = n$).

Your solution's length, k, must be at most 100 000. It can be shown that if a palindromic walk exists, then there exists some palindromic walk whose length is at most 100 000.

If there are multiple solutions, any will be accepted.

Sample Input 1

```
6 6
C A B A B C
1 2
2 3
3 1
1 5
5 4
4 6
```

Sample Output 1

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

Sample Input 2

```
2 1
X Y
1 2
```

Sample Output 2

```
Impossible
```

Sample Input 3

```
2 2
T T
1 2
2 1
```

Sample Output 3

```
Possible
4
1 2 1 2
```

Sample Input 4

```
6 7
A B C C B A
1 2
2 3
3 4
3 5
4 5
4 6
5 6
```

Sample Output 4

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

Problem J
Jeff's Macros
Time limit: 60 seconds

Jeff is teaching an algorithms course at his local university and has introduced his students to his new favourite editor, Screamacs. For their final assignment, students are required to complete a multiple-choice test, where each question offers four options (A, B, C, D), with exactly one correct answer each.

To enhance the learning experience, Jeff allows students to utilise Screamacs' innovative feature: magic macros! A magic macro is a sequence of letters (for example, `ABC`). When invoked, the macro searches through the text for its first letter and replaces that letter with the entire macro string. If its first letter appears multiple times, it will select one of them.

For instance, with the macro `ABC`, the string `BAC` can expand to `BABCC`, while `BAA` could yield either `BAABC` or `BABCA`.

To evaluate the assignment, which consists of n questions, Jeff will recursively apply the macros in all possible ways to generate strings of length n. For each resulting string of length n, he will tally the number of correct answers, which constitutes the score for that particular string. The highest score among all such macro expansions will be the student's grade. If no macro expansion results in a string of length n, the student's grade will be 0.

For example, suppose the assignment has 5 questions and the available macros are `AB` and `CD`. If the student's submission is `AC`, there are 8 different strings of length 5 that can be generated. For instance, `ABBCD` can be formed through the following macro expansion:

$$\texttt{AC} \to \texttt{ABC} \to \texttt{ABCD} \to \texttt{ABBCD}$$

If the correct answers are `BACDD`, then the best macro expansion from `AC` is `ABCDD`, which gives a score of 3 (the last three questions are answered correctly).

Given the student's submission, the correct answers, and the available macros, what is the student's grade?

Input

The first line of input contains two strings S, which is the student's submission, and A, which represents the answers to the assignment. Both S and A are comprised of between 1 and 8 uppercase letters. The length of S is at most the length of A.

The second line of input contains a single integer m ($1 \leq m \leq 8$), which is the number of available macros.

The next line contains m distinct strings, which are the macros. Each macro is comprised of between 2 and 8 uppercase letters.

Output

Display the student's grade.

Sample Input 1

```
AC BACDD
2
AB CD
```

Sample Output 1

```
3
```

Sample Input 2

```
A AAAAAAA
1
AA
```

Sample Output 2

```
7
```

Sample Input 3	Sample Output 3
A AA 1 ABC	0

Sample Input 4	Sample Output 4
AD DABCDCC 3 ABD BC CC	1

Sample Input 5	Sample Output 5
B BBCABC 2 BBCABC BACBACA	6

Problem K
King of Spin
Time limit: 60 seconds

The King of Spin presides over the Kingdom of Rotations: a 2D world with many towns. But everything changed when the Line Nation attacked. King Trevor the Translated rules the Line Nation, another 2D world. The towns in the Line Nation and the towns in the Kingdom of Rotation are located at exactly the same coordinates in their respective worlds. If there is a town at (x, y) in the Line Nation, then there is a town at (x, y) in the Kingdom of Rotations (and vice-versa).

Trevor the Translated rules the Line Nation with an iron fist and has declared that all towns must get up and walk X units along the x-axis and Y units along the y-axis, then rotate R degrees counterclockwise about the origin at $(0, 0)$.

This geometric operation is plainly a power play by Trevor the Translated to instil fear in the hearts of the masses. As such, the King of Spin must respond in kind.

The King of Spin is a master of rotating his kingdom about a point, but he despises moving in a straight line. He wants to find values R', X', and Y' such that he can rotate his kingdom around the point (X', Y') by R' degrees counterclockwise so that his kingdom is exactly the same as Trevor the Translated's to mimic the translation by (X, Y) and rotation of R degrees. This single rotation in the Kingdom of Rotations must match the translate-then-rotate in the Line Nation no matter where the towns on the map start. It can be proven that there is only one such point (X', Y') and rotation of R' degrees that will achieve this.

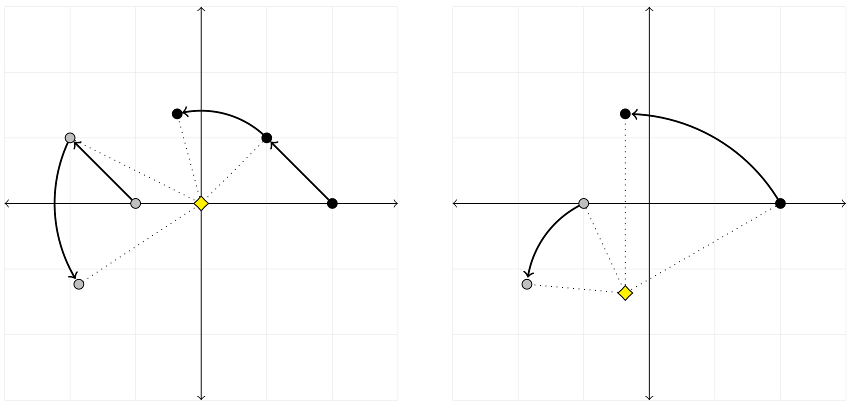


Figure K.1: The left image shows the movement of two towns in the Line Nation. The towns were originally located at $(2, 0)$ and $(-1, 0)$ and were translated by $(-1, 1)$, then rotated 60 degrees around the origin. The right image shows the same towns in the Kingdom of Rotations achieving the same result by rotating 60 degrees around the point $(-0.36, -1.36)$.

Trevor the Translated, who clearly sees his geometric operation as a fine transformation of his own kingdom and much better than any other, doesn't see the point in doing this. But please help the King of Spin find R', X', and Y'.

Input

The first line contains three integers, R $(1 \le R \le 359)$, X, and Y $(-100\,000 \le X, Y \le 100\,000)$, denoting the angle of rotation in degrees and the translation in the x and y directions, respectively.

Output

Display three real numbers: R' $(0 \le R' < 360)$, X', and Y', denoting the angle of rotation in degrees and the centre of rotation on the x- and y-axis, respectively.

Your answer should have an absolute or relative error of less than 10^{-6}. Since R' is an angle, if $|R'| \le 10^{-3}$

or $|R' - 360| \leq 10^{-3}$, then your answer will be correct if any of R', $R' + 360$, or $R' - 360$ has an absolute or relative error less than 10^{-6} to the correct answer.

Sample Input 1	Sample Output 1
`60 -1 1`	`60.0000000000 -0.3660254038 -1.3660254038`

Sample Input 2	Sample Output 2
`1 -1 -1`	`1.0000000000 57.7943250647 -56.7943250647`

Problem L
Loom of Enchantment
Time limit: 60 seconds

In a mystical land, there exists an enormous enchanted loom, which is a straight line of length 10^6. The loom is said to hold the secrets of the universe, and only the wisest and bravest can unlock its mysteries. One day, a group of adventurers discovered n enchanted points on this loom. The ith enchanted point is located p_i metres from the left end of the loom.

For any triplet of enchanted points, (i, j, k), the magical value of that triplet is

$$\text{magic}(i, j, k) = |p_i - p_j| \times |p_j - p_k|.$$

The adventurers were tasked with finding the mystical value, which is the sum of the magical values for all triplets of enchanted points (i, j, k) such that $1 \le i < j < k \le n$:

$$\text{mystical} = \sum_{1 \le i < j < k \le n} \text{magic}(i, j, k).$$

Your mission, should you choose to accept it, is to help the adventurers by calculating the mystical value.

Input

The first line of input contains a single integer n ($3 \le n \le 300\,000$), which is the number of enchanted points.

The next line contains n distinct integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le 10^6$), which are the locations of the enchanted points.

Output

Display the mystical value, modulo $1\,000\,000\,007$.

Sample Input 1	Sample Output 1
4 1 2 3 4	6

Sample Input 2	Sample Output 2
7 3 1 4 5 9 2 6	424

This page is intentionally left (almost) blank.