International Collegiate Programming Contest

2019

Latin American Regional Contests

November 8th-9th, 2019

Contest Session

This problem set contains 13 problems; pages are numbered from 1 to 20.

This problem set is used in simultaneous contests hosted in the following countries:

Argentina, Bolivia, Brasil, Chile, Colombia, Costa Rica, Cuba, Ecuador
El Salvador, México, Perú, República Dominicana and Venezuela

General information

Unless otherwise stated, the following conditions hold for all problems.

Program name

1. Your solution must be called *codename*`.c`, *codename*`.cpp`, *codename*`.java`, *codename*`.kt`, *codename*`.py2` or *codename*`.py3`, where *codename* is the capital letter which identifies the problem.

Input

1. The input must be read from standard input.

2. The input consists of a single test case, which is described using a number of lines that depends on the problem. No extra data appear in the input.

3. When a line of data contains several values, they are separated by *single* spaces. No other spaces appear in the input. There are no empty lines.

4. The English alphabet is used. There are no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).

5. Every line, including the last one, has the usual end-of-line mark.

Output

1. The output must be written to standard output.

2. The result of the test case must appear in the output using a number of lines that depends on the problem. No extra data should appear in the output.

3. When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output. There should be no empty lines.

4. The English alphabet must be used. There should be no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).

5. Every line, including the last one, must have the usual end-of-line mark.

6. To output real numbers, round them to the closest rational with the required number of digits after the decimal point. Test case is such that there are no ties when rounding as specified.

Problem A – Algorithm Teaching

The Latin American Beginners Regional Contest is coming, and the University of Byteland wants to prepare a team of newly-admitted students to compete. The university has N teachers that can instruct students in the topic of algorithms. Each candidate student must be trained by a single teacher, in a non-empty subset of the algorithms that the teacher knows. For example, if a given teacher knows the two algorithms PRIM and KRUSKAL, then the teacher can train a student just on PRIM, just on KRUSKAL, or on both PRIM and KRUSKAL. As you can see, in this case there are three different options for this particular teacher to train a student. In general, a given teacher that knows A different algorithms can train a student in $2^A - 1$ different ways. All these $2^A - 1$ options can be carried out, because the university has a lot of new students, and there is no limit on the number of students a teacher can train.

The university would like to form a team having as many students as possible. However, each pair of students in the final team must be able to cooperate, which means that each one of them must have been trained on an algorithm that the other hasn't. For example, a student trained on BFS and DFS can cooperate with another student trained on DFS and DIJKSTRA, because the first student is trained on BFS while the second student isn't, and the second student is trained on DIJKSTRA while the first student isn't. On the other hand, a student trained on BFS and DFS cannot cooperate with another student trained just on BFS, or just on DFS, or on both BFS and DFS, among others.

Given the set of algorithms that each teacher knows, you must determine the maximum number of students in a team in which every student can cooperate with each other. Recall that each student must be trained by a single teacher, while each teacher can train as many students as needed. For example, if there is just one teacher who knows the algorithms DFS, BFS and DIJKSTRA, it is possible to prepare a team with up to three students: a first student trained on DFS and BFS, a second student trained on DFS and DIJKSTRA, and a third student trained on BFS and DIJKSTRA.

Input

The first line contains an integer N ($1 \leq N \leq 100$) indicating the number of teachers. Each of the next N lines describes a teacher with an integer A ($1 \leq A \leq 10$) representing the number of algorithms the teacher knows, followed by A different non-empty strings of at most 10 uppercase letters each, indicating the names of the algorithms that teacher knows.

Output

Output a single line with an integer indicating the maximum number of students in a team in which every student can cooperate with each other.

Sample input 1	Sample output 1
1 3 DFS BFS DIJKSTRA	3

Sample input 2	Sample output 2
2 4 BFS DFS LCA RMQ 2 PRIM KRUSKAL	8

Sample input 3	Sample output 3
4 3 BFS DFS DIJKSTRA 4 BFS DFS LCA RMQ 3 DIJKSTRA BFS DFS 3 FLOYD DFS BFS	10

Sample input 4	Sample output 4
1 1 HAVEFUN	1

Sample input 5	Sample output 5
3 4 FFEK DANTZIG DEMOUCRON FFT 4 PRIM KRUSKAL LCA FLOYD 4 DFS BFS DIJKSTRA RMQ	18

Problem B – Build the Perfect House

Alice and Bob are now 45 years old and have enjoyed a long five-years retirement maintaining their vegetable field. Going back and forth to the field from their house in the city has become a really exhausting task, so they decided to move to the field permanently. They currently have a perfect fence surrounding all the vegetable plants (it has minimum perimeter and area), but they will tear down the old fence and build a new perfect house to live in.

So what is a perfect house? Well, you know, there are some requirements. The area of the house must be a square centered at the point of the field that has the most beautiful view. Besides, the house cannot be built over any of the vegetables, although vegetables are allowed right on the border of the house.

Since Alice and Bob love spacious rooms, your task is to find the maximum perimeter a perfect house can have.

Input

The first line contains an integer N ($1 \le N \le 10^4$) indicating the number of vegetable plants in Alice and Bob's field. Vegetable plants are represented as points in the two dimensional plane, being $(0,0)$ the point of the field that has the most beautiful view. Each of the next N lines describes a vegetable plant with two integers X and Y ($-10^9 \le X, Y \le 10^9$) indicating the coordinates of the plant. No two vegetable plants have the same location and none of them is at point $(0,0)$.

Output

Output a single line with a number indicating the maximum perimeter a perfect house can have. The result must be output as a rational number with exactly four digits after the decimal point, rounded if necessary. Notice that the sides of the house don't need to be aligned with the coordinate axes.

Sample input 1	Sample output 1
1 0 1	8.0000

Sample input 2	Sample output 2
2 10 4 −5 −8	74.9634

Problem C – Cut Inequality Down

Complejidonia has not always been the peaceful and egalitarian land we all know today. The wealthy Constantones were the owners of the local media and plunged Complejidonia into the tyranny of their ruthless economic system: Nlogonialism, a system that promoted extreme unfairness which, strangely enough, always benefited the Constantones.

While the Constantones owned most of the wealth, Cuadradones lived in extreme poverty, and inequality was justified by tagging Cuadradones as lazy and inefficient. The Nlogones would usually look down on Cuadradones, despite working as much as they did, believing they were better off thanks to their mix of hard work and cunning. For the Cubiones and Cuaterniones it was even worse, coming from neighboring countries they were seen as criminals and, at the same time, accused of stealing Complejidonian jobs.

Everything changed after the International Collectivist and Popular Congress (ICPC) managed to overthrown the Constantones and put a new economic system in place, a system which strives for fairness and takes into account that each inhabitant might go through good and bad economic periods in life.

In the new system an upper limit U on how much wealth an individual can accumulate and a lower bound L representing the minimum wealth required in order for an individual to keep a decent lifestyle were established. At the end of each month every inhabitant will evaluate their wealth. Those with more than U will donate what they own above the upper limit to the ICPC while the ones who sadly have less than L will receive enough from ICPC to reach the established lower bound.

The Cuadradones, which are very good farmers, need your help managing their finance. The long era of Nlogonialism has seriously harmed the environment and now the weather in Complejidonia is very volatile. This has a big impact on its agriculture that fluctuates between good and bad periods.

A farmer keeps a long record $A_1, A_2, \ldots, A_N$ of their net income (income minus expenses) on a sequence of N months. Based on this data the farmer wants to plan how to invest their wealth in order to avoid being a burden to the ICPC in the future. The farmer wants to be able to know, given an initial wealth X at the beginning of a month B, how much they would own at the end of a month E (considering that at the end of each month they might either donate or receive a donation to ensure their wealth is between L and U, inclusive).

Input

The first line contains three integers N ($1 \leq N \leq 10^5$), L and U ($1 \leq L \leq U \leq 2 \times 10^6$), indicating respectively the number of months for which the farmer has net income records, the wealth lower bound and the wealth upper bound. The second line contains N integers $A_1, A_2, \ldots, A_N$ ($-10^6 \leq A_i \leq 10^6$ for $i = 1, 2, \ldots, N$), where A_i is the net income on the i-th month. The third line contains an integer Q ($1 \leq Q \leq 10^5$) representing the number of scenarios the farmer is interested in. Each of the next Q lines describes a scenario with three integers B, E ($1 \leq B \leq E \leq N$) and X ($L \leq X \leq U$), indicating that the farmer would like to know how much they would own at the end of month E if they start owning X at the beginning of month B, and each month $j = B, B+1, \ldots, E$ their net income is A_j.

Output

Output Q lines, each line with an integer indicating how much the farmer would own at the end of the period described in the corresponding scenario.

Sample input 1	Sample output 1
5 1 41	1
-10 10 1 -1 -70	40
10	39
2 5 31	38
2 4 30	20
2 4 29	11
2 4 28	11
1 2 20	11
1 2 10	40
1 4 11	40
1 4 10	
1 4 40	
1 4 41	

On the first scenario the farmer's net incomes would be $[10, 1, -1, -70]$ and they start with a wealth of 31:

- At the end of first month their wealth is 41. As $1 \leq 41 \leq 41$ they won't donate nor receive money.

- At the end of second month their wealth is 42. As $42 > 41$ they donate 1, ending the month with a wealth of 41.

- At the end of third month their wealth is 40. As $1 \leq 40 \leq 41$ once again they won't donate nor receive money.

- Finally, at the end of fourth month their wealth is -30. As $-30 < 1$ they receive a donation from the ICPC ending the month with a wealth of 1.

Hence, on this scenario, the farmer ends up owning a wealth of 1.

Problem D – Dazzling Stars

Little Bernie loves to look at the stars in the sky. His favorite constellation is the Ball of Paper Constellation, because of its distinct and unmistakable shape of... a ball of crumpled paper. Bernie downloaded a picture of the constellation from the internet, and now he wants to print it and stick it to his wall. Bernie also likes to watch the paper sheets gradually coming out of the printer, and for this occasion, he made a decision: he wants the stars to be printed in non-increasing order of brightness.

The constellation has N stars. For each one, Bernie knows its brightness level B as well as its X and Y coordinates in the picture, where the X direction points rightwards and the Y direction points upwards. He knows that the pictures are printed from top to bottom (that is, in decreasing order of the Y coordinate), and that everything in a horizontal line is printed simultaneously.

Bernie's plan can be described like this: for any two stars S and T, if S is brighter than T, then S must be printed before or at the same time as T. Before printing the picture, Bernie can rotate it at any angle around any given point, but he cannot scale, reflect or distort it. Now Bernie needs your help to find out if there is any rotation that allows the stars to be printed in the order he wants.

Input

The first line contains an integer N ($3 \leq N \leq 1000$) indicating the number of stars in the constellation. Each of the next N lines describes a star with three integers X, Y ($-10^4 \leq X, Y \leq 10^4$) and B ($1 \leq B \leq 1000$), where X and Y are the coordinates of the star in the picture, and B is its brightness level. No two stars have the same location.

Output

Output a single line with the uppercase letter "Y" if there is any rotation that allows the stars to be printed in non-increasing order of brightness, and the uppercase letter "N" otherwise.

Sample input 1	Sample output 1
4 0 2 1 1 -1 2 3 3 5 4 0 2	Y

Sample input 2	Sample output 2
5 0 4 6 2 4 5 3 7 2 4 4 6 3 0 8	Y

| Sample input 3	Sample output 3
4 -1 2 5 0 0 2 3 4 1 4 2 4	N

Problem E – Eggfruit Cake

Today is Jaime's birthday and, to celebrate, his friends ordered a cake decorated with eggfruits and persimmons. When the cake arrived, to their surprise, they noticed that the bakery didn't use equal amounts of eggfruits and persimmons, but just randomly distributed the fruits over the cake's border instead.

Jaime eats persimmons every day, so he was eager to try some eggfruit on his birthday. However, as he doesn't want to eat too much, his cake slice should be decorated with at most S fruits. Since Jaime doesn't like when a fruit is cut into parts, each fruit should either be entirely in his slice or be left in the rest of the cake. The problem is, with the fruits distributed in such a chaotic order, his friends are having trouble cutting a suitable slice for him.

Jaime is about to complain that his friends are taking too long to cut his slice, but in order to do so, he needs to know how many different slices with at least one eggfruit and containing at most S fruits there are. A slice is defined just based on the set of fruits it contains. As Jaime is quite focused on details, he is able to distinguish any two fruits, even if both fruits are of the same type. Hence, two slices are considered different when they do not contain exactly the same set of fruits. The following picture shows a possible cake, as well as the six different slices with at most $S = 2$ fruits that can be cut from it.
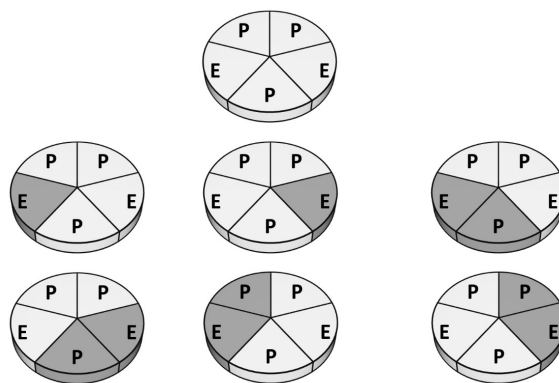
Input

The first line contains a circular string B ($3 \le |B| \le 10^5$) describing the border of the cake. Each character of B is either the uppercase letter "E" or the uppercase letter "P", indicating respectively that there's an eggfruit or a persimmon at the border of the cake. The second line contains an integer S ($1 \le S < |B|$) representing the maximum number of fruits that a slice can contain.

Output

Output a single line with an integer indicating the number of different slices with at most S fruits and at least one eggfruit.

Sample input 1	Sample output 1
PEPEP 2	6

Sample input 2	Sample output 2
EPE 1	2

Sample input 3	Sample output 3
PPPP 1	0

Sample input 4	Sample output 4
EPEP 2	6

Problem F – Fabricating Sculptures

Miguel Angelo is a great sculptor, widely recognized for his outdoor sculptures. In his hometown, it is very common to find one of his creations in squares and gardens. People love his sculptures, not only for their beauty, but also because they look like new even after decades. The sculptures do not degrade easily due to the material and technique developed by Miguel and his staff over the years.

To build the sculptures, he first constructs its base by stacking blocks of waterproof plaster (his secret material), forming several stacks of blocks in a straight line. He always uses identical blocks, and each stack has at least one block. To stabilize the structure, he surrounds it by two big glass panes, one behind the stacks and one in front of them. Then he waits for the rain for as long as it takes. If the structure is such that it doesn't accumulate water during this procedure, Miguel is sure that the base can be used to obtain a piece of long-lasting artwork. Notice that water will accumulate on a block if there are obstacles (other blocks) on both sides (to the left and to the right).

The following picture shows the front view of several different bases. All of them consist of three stacks made of a total of six blocks, with each stack having at least one block as required. However, the eight bases on the left will lead to long-lasting artwork, while the two bases on the right will not.
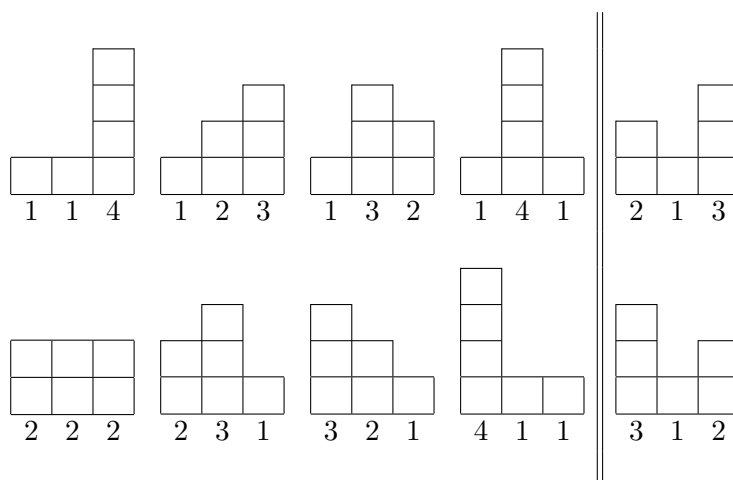
Miguel Angelo is receiving a lot of sculpture requests. Although he has all the freedom to create the artwork, he wants to be fair and use the same number of stacks and the same number blocks in each of the sculptures. Since he doesn't want to sell identical sculptures to different clients, he will construct a different base each time.

He worries that he won't be able to fulfill all the requests. Help him calculate the number of different bases given the number of stacks and the number of blocks that the base must have.

Input

The input consists of a single line that contains two integers S and B ($1 \le S \le B \le 5000$) indicating respectively the number of stacks and the number of blocks that the base must have.

Output

Output a single line with an integer indicating the number of different bases that don't accumulate water which Miguel can construct. Because this number can be very large, output the remainder of dividing it by $10^9 + 7$.

Sample input 1	Sample output 1
3 6	8

Sample input 2	Sample output 2
3 7	12

Problem G – Gluing Pictures

Enzo recently traveled to the city of Montevideo, where he saw a big sign with the name of the city. He decided to take pictures of the sign to make a collage and send it to his friend Demonio. Enzo wants to form the name of his friend by taking one or several pictures of sections of the sign. For example, with the string "MONTEVIDEO", he might form the name of his friend by putting together "DE-MON-I-O", using four pictures to form the entire name. It is easy to show that the result cannot be achieved with fewer pictures.

You will be given the name of a city and a list of friends' names. Return the minimum number of pictures needed to form the name of each friend. When forming the names, pictures cannot be rotated, reflected or modified in any way.

Input

The first line contains a string C indicating the name of the city. The second line contains a positive integer N representing the number of friends. Each of the following N lines contains a string indicating the name of a friend. All strings are non-empty and consist only of uppercase letters. The sum of the lengths of all strings is at most 2×10^5.

Output

Output N lines, each line with an integer indicating the minimum number of needed pictures to form the corresponding name in the input, or the value "-1" if it is not possible to generate the name.

Sample input 1	Sample output 1
MONTEVIDEO 4 DEMONIO MONTE EDIT WON	4 1 4 -1

Sample input 2	Sample output 2
SANTIAGO 3 TITA SANTIAGO NAS	3 1 3

Problem H – Hold or Continue?

Pig is a simple dice game for two or more players. Each turn, a player repeatedly rolls a dice until either a 1 is rolled or the player decides to "hold":

- If the player rolls a 1, they score nothing in their turn and it becomes the next player's turn.

- If the player rolls any other number, it is added to their turn total and the player can decide between "hold" or "continue".

- If the player chooses to "hold", their turn total is added to their score and it becomes the next player's turn. Otherwise the player continues rolling the dice.

The first player to score *exactly* 75 wins the game. If a player's score plus their turn total exceeds 75, they score nothing in their turn and it becomes the next player's turn.

Catelyn Tully is playing Pig with her father Hoster. If Catelyn begins her turn rolling a 5, she could hold and score 5 during her turn. If she chooses to continue and rolls a 2, she could hold and score 7. If she chooses again to continue and rolls a 1, she must end her turn without scoring. If at his turn Hoster rolls the sequence 4-5-3-5-5 and then he chooses to hold, he adds his turn total of 22 to his current score (unless the sum exceeds 75). Then Catelyn rolls the dice again, and so on, until one of them scores exactly 75.

Hoster finds the game very didactic and he became a pretty good player. After playing several times with Catelyn, he realized that she is very impulsive and continues rolling the dice more times than she should. Catelyn would like to improve the way she plays, but unfortunately Hoster doesn't have too much patience to teach her, so she needs your help. While playing with her father, Catelyn has to decide several times whether to hold or continue, and sometimes she is not sure about what to do. Can you advise her so that each decision maximizes her winning probability?

Input

The first line contains an integer Q ($1 \leq Q \leq 1000$) indicating the number of questions on which Catelyn wants your advice. Each of the next Q lines describes a question with three integers C, H and X ($0 \leq C, H \leq 73$, $X \geq 2$, $C + X \leq 75$), representing respectively Catelyn's current score, Hoster's current score, and Catelyn's turn total (sum of the rolls of the dice during her turn).

Output

Output Q lines, each line with a character indicating the decision Catelyn must make for the corresponding question of the input, so as to maximize her winning probability if both Catelyn and Hoster play optimally. For each question, the character must be the uppercase letter "H" if the optimal decision is to hold, or the uppercase letter "C" if the optimal decision is to continue. It is guaranteed that the optimal decision can be clearly differentiated; this means that $|p_h - p_c| > 10^{-5}$, where p_h is the winning probability if Catelyn decides to hold, and p_c is the winning probability if she decides to continue ($0 \leq p_h, p_c \leq 1$).

Sample input 1	Sample output 1
3	C
15 0 3	H
35 50 40	H
15 0 30	

Problem I – Improve SPAM

After the amazing job you did cleaning up duplicate users from the client database, your boss is eager to be impressed by your improvements to the company SPAM (System for Publishing Amazing Marketing).

Despite the marketing campaigns being extremely useful for clients, some complaints were received by the customer service indicating that too many messages are being sent, and that certain clients even receive the same message multiple times.

SPAM is based on mailing lists. Each mailing list is composed of client emails and/or other mailing lists. Client emails might be added to existing mailing lists at any point in time, while only when a mailing list is created it can be added to any number of existing mailing lists. Notice that it is not possible to create several mailing lists at the same time.

When a message is sent to a mailing list, the system sends the message to each address in the list. If the address in the list is a client email, then the message is sent to that client email; if instead the address is a mailing list, then the process is started for that mailing list.

Due to privacy reasons, in the following example mailing lists and client emails are represented by integers. Suppose that 1, 2 and 3 are mailing lists, while 4 and 5 are client emails. Moreover, mailing list 1 contains mailing lists 2 and 3, mailing list 2 contains client emails 4 and 5, while mailing list 3 contains client email 4 and mailing list 2. Now suppose that a message is sent to mailing list 1. This means that the list is processed as described above, and then mailing lists 2 and 3 are also processed. When mailing list 2 is processed, the message is sent to client emails 4 and 5. When mailing list 3 is processed, a second message is sent to client email 4, and mailing list 2 is processed again, which yields a third message sent to client email 4 and a second message sent to client email 5. Thus, a total of five messages are sent to client emails.

Your task is to optimize SPAM in such a way that no client receives the same message multiple times. As a first step, your boss wants to know the number of messages sent before and after your improvements. In the above example, just two messages should be sent to client emails after your work is done.

Input

The first line contains two integers N and L ($2 \leq N \leq 2000$, $1 \leq L \leq \min(N - 1, 1000)$), representing respectively the number of addresses in the system, and the number of addresses that are mailing lists. Addresses are identified by distinct integers from 1 to N. Addresses from 1 to L are mailing lists, while the rest are client emails. For $i = 1, 2, \ldots, L$, the i-th of the next L lines describes mailing list i with an integer K ($1 \leq K < N$) followed by K different integers $M_1, M_2, \ldots, M_K$ ($1 \leq M_i \leq N$ for $i = 1, 2, \ldots, K$), indicating that the mailing list contains the K addresses $M_1, M_2, \ldots, M_K$. Each client address appears in at least one mailing list.

Output

Output a single line with two integers B and A indicating respectively the number of messages sent to client emails before and after your improvements, if a message is sent to mailing list 1. Because these numbers can be very large, output the remainder of dividing them by $10^9 + 7$.

Sample input 1	Sample output 1
5 3 2 2 3 2 4 5 2 4 2	5 2

Sample input 2	Sample output 2
15 6 1 6 7 10 11 12 13 9 7 8 5 6 14 4 5 15 2 14 15 2 4 14 2 5 4	5 2

Sample input 3	Sample output 3
10 5 4 8 9 10 3 3 9 10 6 3 8 9 7 6 2 3 6 7 8 10 5 9 10 3 1 7	6 4

Problem J – Jumping Grasshopper

Jazmín has a garden in front of her house, with N plants forming a line in the left-right direction. She is very proud of her plants, and measures their growing heights with great precision, because she thinks that each plant is different and so it must have a different height.

One day Jazmín found a grasshopper in her garden, and after watching it for some time, she noticed a very peculiar behavior. During each jump the grasshopper moves to the first plant taller than its current plant, in the direction it is looking at. Besides, before landing on the new plant, the grasshopper does a backflip that changes its direction. That is, if before jumping the grasshopper is looking to the left, once the jump is completed it is looking to the right, and vice versa. The grasshopper keeps jumping until there is no taller plant in the direction it is looking at.

Jazmín decided to record her sightings of the grasshopper. Each time she saw it, she wrote the plant where the grasshopper was, and the direction it was looking at. She also recorded how her plants were growing, as she always does. Now Jazmín wonders, for each sighting of the grasshopper, on which plant it stopped jumping. Jazmín's notebook is currently broken, so she cannot write her own program. Can you help her?

Input

The first line contains two integers N and M ($1 \leq N, M \leq 2 \times 10^5$), indicating respectively the number of plants and the number of records. Plants are identified by distinct integers from 1 to N according to their positions in front of the house, starting from the leftmost plant. The second line contains N different integers $H_1, H_2, \ldots, H_N$ ($0 \leq H_i \leq 10^9$ for $i = 1, 2, \ldots, N$), where H_i is the initial height of plant i. The following M lines describe Jazmín's records in chronological order, one line per record. If a record represents the growing of a plant, the line contains the uppercase letter "U" followed by two integers I ($1 \leq I \leq N$) and H ($H \leq 10^9$), indicating that the new height of plant I is H; the new height H is greater than the old height of plant I, and different from the current height of each of the other plants. If a record represents the sighting of the grasshopper, the line contains the uppercase letter "L" or the uppercase letter "R" followed by an integer J ($1 \leq J \leq N$) indicating that the grasshopper starts jumping from plant J; the grasshopper starts looking to the left if the letter is "L", and it starts looking to the right if the letter is "R". There is at least one record that is a sighting of the grasshopper.

Output

Output a line for each sighting of the grasshopper. The line must contain an integer indicating the plant where the grasshopper stops jumping. Write the results in chronological order, that is, using the same order of the input.

Sample input 1	Sample output 1
10 4 1 8 5 6 10 20 12 15 2 4 L 2 R 3 U 10 16 L 9	2 5 6

Problem K – Know your Aliens

Our world has been invaded by shapeshifting aliens that kidnap people and steal their identities. You are an inspector from a task force dedicated to detect and capture them. As such, you were given special tools to detect aliens and differentiate them from real humans. Your current mission is to visit a city that is suspected of have been invaded, secretly inspect every person there so as to know whose are aliens and whose aren't, and report it all to Headquarters. Then they can send forces to the city by surprise and capture all the aliens at once.

The aliens are aware of the work of inspectors like you, and are monitoring all radio channels to detect the transmission of such reports, in order to anticipate any retaliation. Therefore, there have been several efforts to encrypt the reports, and the most recent method uses polynomials.

The city you must visit has N citizens, each identified by a distinct even integer from 2 to $2N$. You want to find a polynomial P such that, for every citizen i, $P(i) > 0$ if citizen i is a human, and $P(i) < 0$ otherwise. This polynomial will be transmitted to the Headquarters. With the aim of minimizing bandwidth, the polynomial has some additional requirements: each root and coefficient must be an integer, the coefficient of its highest degree term must be either 1 or -1, and its degree must be the lowest possible.

For each citizen, you know whether they're a human or not. Given this information, you must find a polynomial that satisfies the described constraints.

Input

The input consists of a single line that contains a string S of length N ($1 \leq N \leq 10^4$), where N is the population of the city. For $i = 1, 2, \ldots, N$, the i-th character of S is either the uppercase letter "H" or the uppercase letter "A", indicating respectively that citizen $2i$ is a human or an alien.

Output

The first line must contain an integer D indicating the degree of a polynomial that satisfies the described constraints. The second line must contain $D + 1$ integers representing the coefficients of the polynomial, in decreasing order of the corresponding terms. It's guaranteed that there exists at least one solution such that the absolute value of each coefficient is less than 2^{63}.

Sample input 1	Sample output 1
HHH	0 1

Sample input 2	Sample output 2
AHHA	2 -1 10 -21

| Sample input 3	Sample output 3
AHHHAH	3 1 -23 159 -297

Problem L – Leverage MDT

The kingdom of Nlogonia is a very prosperous one. Its king, Constantourist, expanded the kingdom by conquering nearby towns. However, now that Constantourist's life is coming to an end, his two sons, Javasar and Golangsar, need to decide the fate of the kingdom.

Instead of fighting an unnecessary war to choose the next king, the two sons are trying to negotiate an agreement so as to divide the kingdom's jurisdiction into two. Nlogonia is a rectangle-shaped land having N kilometers in the North-South direction and M kilometers in the West-East direction. Thus, during the initial stage of negotiations the two sons were able to divide the land in $N \times M$ square parcels of one kilometer on each side, using divisory lines parallel to the kingdom's border. The next step is to distribute the parcels between the two sons.

Before the negotiations can continue, Javasar needs to decide which parcels he wants to claim for himself. He has already categorized each parcel in either good or bad, according to soil quality. Javasar wants his jurisdiction to be recognized as the best in Nlogonia, and so he is planning to choose only parcels with good soil quality. Moreover, being a perfectionist, he decided that the parcels that he will claim need to form a square.

Javasar is worried that these requirements might make him get only a few amount of parcels. Luckily for him, during one of his adventures to Byteland, he found an ancient Magical Divine Tool (MDT) which, when active, is able to reverse the soil quality of the parcel Javasar is currently standing in. In other words, if active, the MDT turns a bad quality parcel into a good one, and vice versa.

With this handy tool, Javasar came up with the perfect plan! He will travel outside the kingdom, to the West of the parcel that is at the North-West corner, and then he will visit each parcel exactly once following the route that the picture below shows. Notice that Javasar will enter and leave Nlogonia several times. In this way he will avoid activating or deactivating the MDT when he is inside the kingdom, and then nobody will see him manipulating the tool. Although the MDT is magical and divine, it does not activate or deactivate by itself.
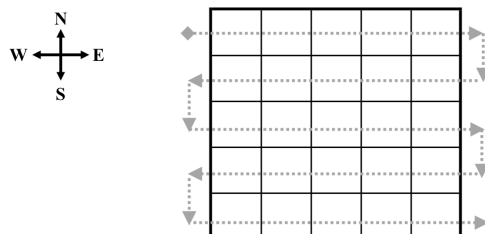
As Javasar's main advisor, you must tell him the maximum number of parcels it is possible to obtain, meeting his requirements, if he optimally leverages the MDT.

Input

The first line contains two integers N and M ($1 \leq N, M \leq 1000$), indicating the length (in kilometers) of Nlogonia in the North-South and West-East directions, respectively. Each of the next N lines contains a string of M characters, where each character is either the uppercase letter "G" or the uppercase letter "B", representing that the soil quality of a parcel is good or bad, respectively. The description of the kingdom's parcels is made from North to South, and from West to East.

Output

Output a single line with an integer indicating the maximum number of parcels Javasar can obtain, whilst meeting his requirements, if he optimally leverages the MDT.

Sample input 1	Sample output 1
2 2 GG GG	4

Sample input 2	Sample output 2
5 5 GGGGG GBBBG GBBBG GBBBG GGGGG	9

Problem M – Mountain Ranges

Famous for its mountain ranges, Nlogonia attracts millions of tourists every year. The government has a dedicated budget for continuous maintenance of the hiking trails spread across the country and most of them are filled with scenic viewpoints, accessible through wooden walkways and stairs.

Currently on a trip through Nlogonia and with hopes of going back home with lots of breath-taking pictures, Lola and her husband want to visit as many viewpoints as possible. They plan to hike a different trail each day and explore its viewpoints. However, to avoid being exhausted at the end of the day, if moving from one viewpoint to the next requires going up more than X meters they simply call it a day and go back to their hotel in order to get some rest. Fortunately, every hiking trail in Nlogonia is equipped with modern chairlifts, so the couple can start hiking the trail at any viewpoint they decide. Once the hiking begins the couple only moves towards the peak of the mountain.

To make sure she doesn't waste a day Lola only wants to hike on trails where she'll get to a reasonable number of viewpoints. Given the altitudes of the scenic viewpoints on a hiking trail, you must determine the maximum number of viewpoints that the couple can visit.

Input

The first line contains two integers N ($1 \leq N \leq 1000$) and X ($0 \leq X \leq 8848$), indicating respectively the number of scenic viewpoints on the hiking trail, and the maximum number of meters that Lola and her husband are willing to go up from one viewpoint to the next. The second line contains N integers $A_1, A_2, \ldots, A_N$ ($1 \leq A_i \leq 8848$ for $i = 1, 2, \ldots, N$), where A_i is the altitude (in meters) of the i-th viewpoint. Viewpoints are given in the order they appear on the hiking trail and their altitudes are non-decreasing, that is, $A_i \leq A_{i+1}$ for $i = 1, 2, \ldots, N - 1$.

Output

Output a single line with an integer indicating the maximum number of scenic viewpoints that can be visited without going up more than X meters from one viewpoint to the next, and considering that the journey can be started at any viewpoint.

Sample input 1	Sample output 1
9 2 3 14 15 92 653 5897 5897 5898 5900	4

Sample input 2	Sample output 2
9 0 3 14 15 92 653 5897 5897 5898 5900	2

Sample input 3	Sample output 3
9 8848 3 14 15 92 653 5897 5897 5898 5900	9