Problem A
Basketball One-on-One


Photo by Pixabay

Alice and Barbara played some friendly games of one-on-one basketball after work, and you agreed to help them keep score. The rules of the game were simple:

- Each successful shot by a player earns them either one or two points;
- The first player to eleven points wins, with one exception;
- If the score is tied 10–10, the previous rule is replaced by a "win by 2" rule: the first player to lead the other by at least two points wins.

So for example, 11–7, 9–11, and 14–12 are possible final scores (but not 14–13).

Whenever Alice or Barbara scored points, you jotted down an `A` or `B` (indicating a score by Alice or by Barbara) followed by a `1` or `2` (the number of points scored). You have some records of the games Alice and Barbara played in this format, but do not remember who won each game. Can you reconstruct the winner from the game record?

Input

The input consists of a single line with no more than 200 characters: the record of one game. The record consists of single letters (either `A` or `B`) alternating with single numbers (either `1` or `2`), and includes no spaces or other extraneous characters. Each record will be a correct scoring history of a single completed game, played under the rules described above.

Output

Print a single character, either `A` or `B`: the winner of the recorded game.

Sample Input 1	Sample Output 1
`A2B1A2B2A1A2A2A2`	`A`

Sample Input 2	Sample Output 2
`A2B2A1B2A2B1A2B2A1B2A1A1B1A1A2`	`A`

This page is intentionally left blank.

Problem B
Commemorative Race

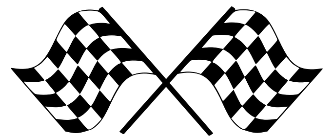


Photo by Pixabay

Filiberto is planning a race to celebrate the 2019 ICPC Mid-Central Regional. The racers compete by going down a mountain on a set of roads. As it would be dangerous to have cars speeding at each other in opposite directions, racers may only drive downhill on each road in a single direction. The endpoint of each road is called a station, and roads connect only at stations.

A path is a sequence of roads where the ending station of one road is the starting station of the next. Each racer must choose a path to race on. The length of a path is the number of roads in it.

The goal of this race is to stay on the course as long as possible – that is, to find a path of maximum length. Every racer knows the full layout of the stations and roads, and will choose some path of maximal length.

Filiberto is concerned about natural disasters (e.g., landslides) blocking off roads during the competition. If a road is blocked off, it may reduce the length of some paths of the race. This works as follows: say the road from station u to station v is blocked off. Then the racer will follow their path until station u, at which point they have to modify their path starting from station u, while still trying to maximize the length of their new path. As each racer knows the full map of the mountain, they will optimally re-route to maximize the length of their new path.

To understand the worst-case scenario, Filiberto wants to know the minimum length path that a winning racer would take if at most one of the roads is blocked off. This means that among all sets of maximum length paths that racers can choose and all possible roads that can be blocked off, what is the minimum length path that a racer may take after that road is blocked off and that path is at least as long as any other racers' path (other racers are also effected by the same blocked road)?

Input

The first line of the input contains two integers n and m $(2 \leq n \leq 10^5, 1 \leq m \leq \min(n(n-1)/2, 10^6))$ – the number of stations and number of roads on the mountain. The next m lines each contain two integers u and v $(1 \leq u, v \leq n, u \neq v)$ representing a one-way road from station u to station v. It is guaranteed that there is at most one road between any pair of stations and there are no loops in the road graph (i.e. the input is a directed acyclic graph).

Output

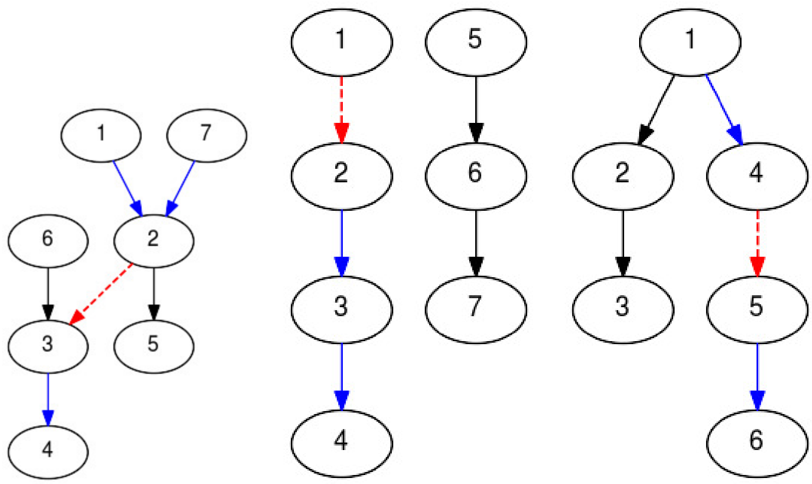
Print a line with a single integer – the minimum length path that competitors can achieve if at most one of the roads is blocked off. That is, Filiberto wants to know the minimum longest time with the worst landslide location.

Sample Explanation

In the second sample, every racer will start from stations number 1 or 7. If the road $(2, 3)$ is blocked, then all racers will have to re-route at station 2 towards station 5.

In the third sample, the minimum path that could be taken happens by blocking the first road of the only path that racers can choose. Blocking any other road results in a longer path.

In the last sample there is only one maximum length path, but the best option is to block the road $(4, 5)$. If we block the road $(1, 4)$ the racers can re-route earlier and achieve a path of length 2.



Sample 2 Sample 3 Sample 4

Sample Input 1

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

Sample Output 1

```
2
```

Sample Input 2

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

Sample Output 2

```
2
```

Sample Input 3

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

Sample Output 3

```
0
```

Sample Input 4	Sample Output 4
6 5 1 2 1 4 2 3 4 5 5 6	1

This page is intentionally left blank.

Problem C
Convoy

Photo by B137

You and your friends have gathered at your house to prepare for the Big Game, which you all plan to attend in the afternoon at the football stadium across town. The problem: you only have k cars between you, with each car seating five people (including the driver), so you might have to take multiple trips to get all n people to the stadium. In addition, some of your friends know the city better than others, and so take different amounts of time to drive to the stadium from your house. You'd like to procrastinate as long as possible before hitting the road: can you concoct a transportation plan that gets all people to the stadium in the shortest amount of time possible?

More specifically, each person i currently at your house can drive to the stadium in t_i minutes. All k cars are currently parked at your house. Any person can drive any car (so the cars are interchangeable). After a car arrives at the stadium, any person currently at the stadium can immediately start driving back to your house (and it takes person i the same amount of time t_i to drive back as to drive to the stadium), or alternatively, cars can be temporarily or permanently parked at the stadium. Drivers driving to the stadium can take up to four passengers with them, but drivers driving back can *NOT* take any passenger. You care only about getting all n people from your house to the stadium—you do *NOT* need to park all k cars at the stadium, if doing so would require more time than an alternative plan that leaves some cars at your house.

Input

The first line of input contains two space-separated integers n and k ($1 \leq n, k \leq 20\,000$), the number of people at your house and the number of available cars. Then follow n lines containing a single integer each; the ith such integer is the number of seconds t_i ($1 \leq t_i \leq 1\,000\,000$) that it takes person i to drive from your house to the stadium, or vice-versa.

Output

Print the minimum number of seconds it takes to move all n people from your house to the stadium, if all people coordinate and drive optimally.

Sample Input 1	Sample Output 1
11 2 12000 9000 4500 10000 12000 11000 12000 18000 10000 9000 12000	13500

Sample Input 2	Sample Output 2
6 2 1000 2000 3000 4000 5000 6000	2000

Problem D
Crazed Boar


A wild boar charging (Wellcome Images)

A crazed boar has become lost in the forest! In its madness, it will charge in a random direction at blazing speed, until it has traveled a distance d, or until it hits a tree (in which case the boar will become dazed and end its charge), whichever comes first. Given the layout of trees around the boar, what is the probability the boar completes its wild charge without hitting a tree?

We will model the forest as the xy plane, with the boar a disk of radius b that begins centered at the origin $(0, 0)$. We will also represent the trees as disks, with varying radii r_i and centers (x_i, y_i). The boar charges by choosing a direction uniformly at random, and then translating in that direction for a distance d. The boar hits a tree and becomes dazed if, at any point during its charge, the boar's body has nonzero area of overlap with any tree.

Input

The first line of input contains a single integer n $(0 \leq n \leq 10\,000)$, the number of trees in the forest. n lines follow, each of which contain three integers x_i, y_i, and r_i, denoting the position and radius of the ith tree. These inputs satisfy $-10^6 \leq x_i, y_i \leq 10^6$ and $0 < r_i \leq 10^6$. The final line of input contains two integer b and d, the radius of the boar $(0 < b \leq 10^6)$ and the distance that the boar will charge $(0 \leq d \leq 10^6)$. You may assume that no tree overlaps with or touches the boar at the start of its charge (but trees might overlap or touch each other).

Output

Print a single real number: the probability that the boar completes its charge without hitting any tree. Your answer will be considered correct if it has absolute or relative error at most 10^{-6}.

Sample Input 1

```
1
3 0 1
1 4
```

Sample Output 1

```
0.76772047
```

Sample Input 2

```
4
6 0 3
0 6 3
-6 0 3
0 -6 3
1 3
```

Sample Output 2

```
0.19253205
```

This page is intentionally left blank.

Problem E
Dance Circle


Children Dancing in a Ring by Hans Thoma

It's Halloween, and you've organized a bonfire and dance for the neighborhood children. The n children have gathered into a ring to dance around the fire. Each child is wearing one of two fun yet spooky costumes: orange pumpkin, or black bat. Since it's dark outside, you can only see a few children at a time, as they pass behind the bonfire. The children are not standing evenly so the number of children you can see at each time differs. In particular, numbering the children $0, 1, \dots, n-1$ clockwise around the dance circle, at any given time you can see child i in the center of your view, as well as l_i children before child i and r_i children after child i around the circle (i.e., child $i-l_i, \dots, i-1, i, i+1, \dots, i+r_i$, where the indices are of course taken modulo n).

To help pass the time while the children dance, you wonder to yourself: suppose you only knew, for each child i, whether an even or odd number of the $l_i + r_i + 1$ children centered at child i is wearing the orange pumpkin costume. Would you be able to uniquely reconstruct what costume each child is wearing? Clearly this is possible when $l_i = r_i = 0$. But what if l_i and r_i are not always zero? Maybe there are multiple possible solutions, or none at all? You decide to investigate, later in the evening once you're back at your computer.

Input

The first line of the input consists of a single integer n, indicating that there are n children in the ring ($1 \leq n \leq 200\,000$). The following n lines describe the children you can see at different times. The ith line (indexed starting from zero) contains three space-separated non-negative integers l_i, r_i, x_i ($l_i + r_i + 1 \leq n, 0 \leq x_i \leq 1$): you can see $l_i + r_i + 1$ children when child i is in the center of view (l_i to the left and r_i to the right of child i). If $x_i = 0$ then an even number of them are wearing the orange pumpkin costume. If $x_i = 1$ then an odd number of them are wearing the orange pumpkin costume.

Output

Compute the number of ways of assigning a costume to each child, consistent with your observations. Since this number might be large, print the result modulo $10^9 + 7$. (If it's impossible to find any costume assignment that matches all parity constraints, print `0`).

Sample Input 1	Sample Output 1
5 1 0 0 1 0 1 3 0 1 3 0 0 3 0 1	0

Sample Input 2	Sample Output 2
5 3 1 1 0 3 1 1 3 1 1 2 1 0 4 1	4

Problem F
Dragon Ball I

Photo by Pixabay

There is a legendary tale about Dragon Balls on Planet X: if one collects seven Dragon Balls, the Dragon God will show up and help you fulfill your wishes.

One day, you are surprised to discover that the tale might possibly be true: you found a Dragon Ball radar at a flea market! The radar shows you the locations of the seven Dragon Balls on Planet X. You want to waste no time checking the truth of the old legend about wish-granting for yourself!

There are n cities in total on the Planet X, numbered from 1 to n. You are currently at city 1. To travel from one city to another, you can take any of m bidirectional teleport trips, as many times as you like. The i-th teleporter costs t_i coins to use each time, and it can teleport you between cities a_i and b_i. To collect a Dragon Ball, you simply need to visit the city where it's located, as indicated on your radar. It is possible that multiple Dragon Balls are at the same city; in this case you pick all of them all up at once if you visit that city.

Input

The first line of input contains two space-separated integers n and m ($1 \le n, m \le 200\,000$), the number of cities and possible teleport trips. Then follow m lines containing three space-separated integers a_i, b_i, and t_i each ($1 \le a_i, b_i \le n, 0 \le t_i \le 10\,000$), which, as explained above, represent the two cities connected by the teleport trip, and cost to use the teleporter. Then follows one line of seven space-separated integers, representing the city IDs of the seven Dragon Balls showing on the radar. Each ID c satisfies the bound $1 \le c \le n$.

Output

Print the minimum number of coins that you need to spend to collect all seven Dragon Balls shown on the Dragon Ball radar. If there is no way to complete this task, print -1 instead.

Sample Input 1

```
10 9
1 2 1
2 3 1
3 4 1
4 5 1
5 6 1
6 7 1
7 8 1
8 9 1
9 10 1
1 2 3 4 5 6 7
```

Sample Output 1

```
6
```

Sample Input 2	Sample Output 2
5 5 1 2 0 1 3 0 2 3 1 3 4 1 4 5 1 1 2 1 2 3 4 4	1

Problem G
Dragon Ball II

Photo by Pixabay

There used to be a legendary tale about Dragon Balls on Planet X: if one collects seven Dragon Balls, the Dragon God will show up and help you fulfill your wishes.

However, ever since a Dragon Ball radar was discovered, which pinpoints the location of each Dragon Ball, collecting all seven balls has become too easy, to the point that the Dragon God is pestered non-stop by frivolous wishes. He has had enough, and so wants to make the process of collecting the balls harder. Instead of seven balls, he has created a huge number, each with a (not necessarily distinct) serial number. To summon the Dragon God, you now need to collect seven Dragon Balls with *pairwise distinct* serial numbers: duplicate Balls do not help you.

You recently bought a Dragon Ball radar at a flea market, which tells you which city each Dragon Ball resides in, as well as that ball's serial number. Can you still find the cheapest way to summon the Dragon God?

There are n cities in total on the Planet X, numbered from 1 to n. You are currently at city 1. To travel from one city to another, you can take any of m bidirectional teleport trips, as many times as you like. The i-th teleporter costs t_i coins to use each time, and it can teleport you between cities a_i and b_i. To collect a Dragon Ball, you simply need to visit the city where it's located, as indicated on your radar. It is possible that multiple Dragon Balls are at the same city; in this case you pick all of them all up at once if you visit that city.

Input

The first line of input contains three space-separated integers n, m, and k $(1 \leq n, k \leq 1\,000, 1 \leq m \leq 10\,000)$, the number of cities, teleport trips, and Dragon Balls, respectively. Then follow m lines containing three space-separated integers a_i, b_i, and t_i each $(1 \leq a_i, b_i \leq n, 0 \leq t_i \leq 10\,000)$, which, as explained above, represent the two cities connected by the teleport trip, and cost to use the teleporter.

Then follow k lines, the i-th of which contains two space-separated integers c_i and d_i $(1 \leq c_i, d_i \leq n)$, indicating that the i-th Dragon Ball is located in city c_i and has serial number d_i. Note that there might be multiple Dragon Balls located in the same city, multiple Dragon Balls that share the same serial number, or even multiple Dragon Balls with the same serial number in the same city.

Output

Print the minimum number of coins that you have to spend to collect seven Dragon Balls with pairwise distinct serial number. If there is no way to complete this task, print -1.

Sample Input 1	Sample Output 1
11 10 10 1 2 1 2 3 1 3 4 1 1 5 1 5 6 1 6 7 1 7 8 1 1 9 1 1 10 1 1 11 1 2 1 3 2 4 3 5 1 6 2 7 3 8 4 9 5 10 6 11 7	10

Problem H
Farming Mars

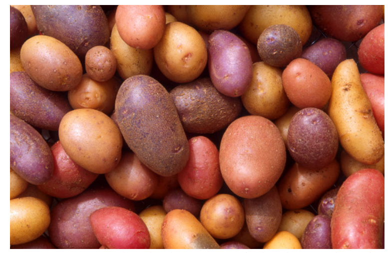


Photo by the USDA Agricultural Research Service

It has been ten years since the miserable day that you won a one-way ticket to Mars on a game show. Your colony's attempt at terraforming Mars has faced nothing but hardship during that time. The latest disaster: a complete failure of the potato crop. You will have to start planting potatoes again from scratch. Your colony has prepared an n-acre strip of arable land on which you will attempt to plant new potatoes.

Your scientists have developed bioengineered potatoes that can withstand the harsh Martian climate and almost complete lack of atmosphere. Unfortunately, these potatoes have extreme sensitivity to pH: they will only thrive if the pH of the soil is exactly right, down to six decimal digits. Your bioengineers can create new potato varieties with any specific allowable pH value, but doing so is only economical if you can then plant a large patch of the variety on a continuous interval $[l, r]$ of the n-acre strip. Given a list of pH values measured on each acre of the strip, and a list of potential subintervals $[l, r]$ where you are considering planting potatoes, compute whether a strict majority $\lfloor (r-l+1)/2 \rfloor + 1$ of the acres within the interval all share the exact same pH value (otherwise it is not worth trying to plant potatoes there). Note that these acres with equal pH value do not need to be contiguous, so long as they all lie within the interval $[l, r]$.

Input

The first line of input contains two space-separated integers n and m $(1 \leq n, m \leq 10\,000)$, the size of the strip of land and the number of queries, respectively. The next n lines contain a single real number: the ith such number is the pH of the ith acre along the strip. Each pH value lies between 0.000000 and 14.000000, inclusive, and contains exactly six decimal digits after the decimal point. Then follows m lines containing two space-separated integers each: the bounds l_j and r_j of the jth query $(1 \leq l_j \leq r_j \leq n)$.

Output

Print m lines of output, one for each query. On line j, print `usable` if a strict majority of the land between acres l_j and r_j (inclusive) all share the exact same pH value, and `unusable` otherwise.

Sample Input 1	Sample Output 1
8 4 7.000000 8.314634 7.000001 7.000000 2.581236 7.000000 2.581236 7.000000 1 8 1 3 4 8 5 7	unusable unusable usable usable

Problem I
Soft Passwords

pixabay image

Your favourite social media website is changing their policy on login password validation: a slight error when logging in is now acceptable! In particular, assuming the password you chose when creating the account is S, a password P entered while logging in will be accepted if any of the following conditions are met:

- P and S are identical;
- S can be formed from P by prepending a single digit (`0`–`9`);
- S can be formed from P by appending a single digit;
- S is equal to P after reversing the case of all letters in P.

To reverse the case of a string, replace all uppercase letters with their equivalent lowercase letters, and all lowercase letters with their equivalent uppercase letters, while leaving all other characters the same. For example, the case-reversal of `pa55WORD` is `PA55word`.

Any other attempted password P will be rejected. So for example, if S is `c0deninja5`, then `c0deninja` will be accepted, but not `C0deninja5` or `c0deninja51`.

Write a program which, given alphanumeric strings S and P, determines whether P should be accepted.

Input

The first line of the input is the string S, the stored password, and the second line of input is the password P that a user has entered while attempting to log in. Each string consists of only digits `0`–`9`, lowercase letters `a`–`z`, and uppercase letters `A`–`Z`. The strings won't contain spaces or any other extraneous characters, and will each contain at least one and at most 101 characters.

Output

Print `Yes` if P should be accepted according to the above rules, and `No` otherwise.

Sample Input 1	Sample Output 1
`123` `123a`	`No`

Sample Input 2	Sample Output 2
`abc` `ABC`	`Yes`

This page is intentionally left blank.

Problem J
Sum and Product

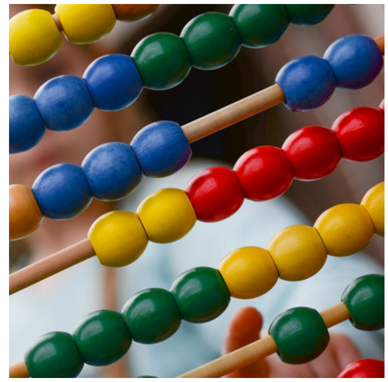


Photo by Skitterphoto

Sarah and Patricia are young, very gifted sisters. Ever since they learned arithmetic in preschool, they have been bothering their mother Marguerite day and night to practice their calculations. Marguerite bought the two girls an arithmetic practice book, which has n pages and a positive integer on each page. Those integers can be used for practicing arithmetic calculations. Marguerite hopes that the book can keep the two girls busy for a while so that she can work on her other chores.

Sarah and Patricia need to know whether their calculated answers are right, but Marguerite does not have the time to check their work. She therefore comes up with the clever idea of *adversarial toddler training*. Marguerite knows that Sarah is good at addition and Patricia is talented in multiplication. So she first chooses a range of consecutive book pages (at least two), and then asks Sarah to calculate the sum of the integers on these pages, and asks Patricia to calculate the product of these integers. If she chooses the pages carefully, then the two girls' answers would be the same, and she can simply let Sarah and Patricia compare their answers!

As Sarah and Patricia do not want to practice on the same range multiple times, Marguerite wants to know in how many different ways she can select a range of pages for her toddler training (and when to buy a new practice book).

Input

The input has a single integer n ($2 \le n \le 2 \cdot 10^5$) on the first line. The second line has n positive integers giving the numbers in the arithmetic practice book ordered by the pages. None of these integers are larger than 10^9.

Output

Output a single integer, the number of ways for Marguerite to choose a range of at least two consecutive pages so that Sarah's answer matches Patricia's.

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

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

Sample Input 3	Sample Output 3
4 5 6 7 8	0

This page is intentionally left blank.

Problem K
True/False Worksheet

Bob is completing a true/false worksheet, consisting of a list of n problems where each answer is either "true" or "false". The problems are numbered from 1 to n. They are too hard for Bob, so the TA, Alice, has given Bob m hints. For each hint i, Alice gives Bob an (inclusive) range of questions $[l_i, r_i]$, and tells him either "all answers in the range are the same" (in other words, either all are "true", or all are "false"); or "not all of the answers in the range are the same." Help Bob count how many different answer sequences satisfy the given hints. Since this number may be huge, print the answer modulo $10^9 + 7$.

Photo by Pixabay

Input

The first line of the input contains two space-separated integers n and m $(1 \le n \le 5\,000, 1 \le m \le 1\,000\,000)$, the number of problems and number of hints, respectively. The next m lines each encode a hint, and contain two space-separated integers l_i and r_i $(1 \le l_i \le r_i \le n)$ followed by either the word `same`, if all answers in the range are the same, or `different`, if all answers in the range are not the same (i.e., at least one answer is "true" and at least one other answer is "false").

Output

Print the number of different answer sequences satisfying all the hints, modulo $10^9 + 7$.

Sample Explanation

In the first sample, the four possible sequences consistent with the hints are `00000`, `10000`, `01111`, and `11111` where `0` stands for a "false" answer and `1` stands for a "true" answer. In the second sample, the third hint conflicts with the first two hints, so no answer sequence exists consistent with all hints.

Sample Input 1

```
5 2
2 4 same
3 5 same
```

Sample Output 1

```
4
```

Sample Input 2

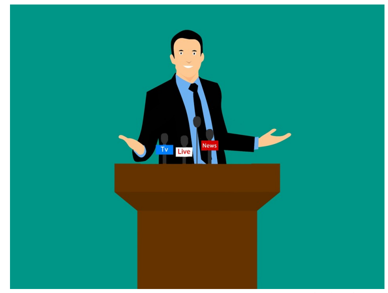
```
5 3
1 3 same
2 5 same
1 5 different
```

Sample Output 2

```
0
```

This page is intentionally left blank.

Problem L
Umm Code


pixabay image

The current programming club president, Norbit, gives speeches at the weekly club meetings. Casual viewers are underwhelmed with the quality of Norbit's elocution. Specifically, Norbit often hesitates during his speeches with interjections like "umm."

You, however, are no casual viewer–you are a computer scientist! You have noticed strange patterns in Norbit's speech. Norbit's interjections, when arranged together, form a binary code! By substituting 1's and 0's for u's and m's, respectively, you produce 7-bit binary ASCII codes that spell out secret messages.

For example, the letter 'a' has an ASCII code of 97, which translates to a binary value of 1100001 and an umm code of "uummmmu". An umm code can be split up in the speech. For example, an encoding of 'a' could be stretched across three utterances: "uum", "mmm", "u" (possibly with other non-umm code words occurring between them).

Now that you have discovered Norbit's secret, you go back through transcripts of his previous speeches to decode his cleverly concealed messages.

Input

There is one line of input of length S ($20 \leq S \leq 500\,000$), which ends with a single newline. Before the newline, the input may contain any characters in the ASCII range 32 – 126 (that is, space (' ') through tilde ('~')).

Let's define a "word" as a space-delimited sequence of characters. If a word does not contain any letters or digits except lowercase `u`'s and/or `m`'s, then it is part of the umm-coded message. If a word contains digits or letters other than lowercase `u` and `m`, then it is not part of the umm-coded message (even if it does contain `u` or `m`). Note that a word that is part of the umm-coded message *may* contain punctuation (which is defined as anything other than letters, digits, or space). Naturally, you should only consider the `u` and `m` characters (and not punctuation) when decoding the umm-coded message. Let M be the length of the entire umm-coded message (counting only its `u` and `m` characters). It is guaranteed that $M \geq 7$ and M is evenly divisible by 7.

Output

Print the de-umm-coded message. Note that for this problem, the judging is case-sensitive. It is guaranteed that each character that should be output is in the same ASCII range as the input.

Sample Input 1	Sample Output 1
`uu Friends m Romans ummuuummmuuuuumm countrymen mmuummmuu`	`icpc`

Sample Input 2	Sample Output 2
`umm ummm uum umm um ummmm u?`	`Hi!`

This page is intentionally left blank.