

Problem A. Puyo Puyo

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes



Figure: Screenshot of PuyoPuyo game. This game uses 12×6 grid. There are 4 kinds of color.

PuyoPuyo is a famous video game series made by *Compile Co, Ltd.* The first game was released in 1991. The company went defunct at 2003, but PuyoPuyo team is still making the games in *Sonic Team*TM to this day, 2018.

PuyoPuyo is a 2 player battle game. Each player plays by placing Puyo to grid. The detailed game rule is as follows:

- The game starts with an empty grid.
- *Puyo* is a round, slime-like object that falls from above to the bottom of the grid
- Each Puyo has a color.
- There are K kinds of colors of Puyo.
- Puyos are controlled as a pair (two Puyos).
- Puyo pair can be moved, dropped or rotated using a game controller.
- At the topmost part of the screen, Puyo pair can be moved horizontally, and can be rotated in horizontal or vertical direction.
- When you drop the Puyo pair, each of the pair drops separately, until it reaches another Puyo or the bottom of the grid.
- You can place Puyo outside of the grid, but only over the grid.
- Four or more adjacent Puyos that have the same colors are called a *Group*, and such groups are removed. This process is called *Popping*.
- If dropping a Puyo pair makes two or more groups, they are popped simultaneously.
- After the groups pops, remaining Puyos fall to another Puyo or the bottom of the grid. If they make new groups, same procedure is repeated. This process is called *Chain*.

- During chains, you cannot place another Puyo pair.

*Sonic Team*TM requested your help to make PPAP (Puyo Puyo Algorithm for Printing). This will not be used in game software but will be used in special ceremonies.

You are given an $R \times C$ grid. Each cell of the grid contains a Puyo of a certain color or is empty. You should start from an empty grid and drop Puyo pairs to derive the given grid.

Fortunately, you can choose which color and place to drop.

Input

First line contains three space-separated integers R , C and K . ($R = C = 4$ or $R = C = 20$, $3 \leq K \leq 6$)

Next R lines contain the final grid. Each line contains C number of space-separated integers. Each color is represented by an integer 1 through K . 0 means an empty cell.

A given final grid can be made by no more than 250 Puyo pair drops.

Output

In the first line, you have to print the number of Puyo pairs D . ($D \leq 250$)

In following D lines, print four space-separated integers in each line.

- If you place Puyo pair horizontally, the first number is 0. If you place vertically, the first number is 1.
- Second number represents the column number of the left Puyo.
- Third number represents the color of left or upper Puyo of the pair.
- Fourth number represents the color of right or lower Puyo of the pair.

Example

standard input	standard output
4 4 3 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1	5 0 2 1 2 1 4 1 2 0 2 2 3 1 3 3 3 1 3 2 3

Note

In this example, Puyos are dropped like following :

```

0 0 0 0    0 0 0 0    0 0 0 0    0 0 0 0    0 0 0 0
0 0 0 0    0 0 0 0    0 0 0 0    0 0 0 0    0 0 2 0
0 0 0 0    0 0 0 0    0 0 0 0    0 0 0 0    0 0 3 0
0 0 0 0 -> 0 0 0 0 -> 0 0 0 0 -> 0 0 3 0 -> 0 0 3 0
0 0 0 0    0 0 0 0    0 0 0 0    0 0 3 0    0 0 3 0
0 0 0 0    0 0 0 1    0 2 3 1    0 2 3 1    0 2 3 1
0 1 2 0    0 1 2 2    0 1 2 2    0 1 2 2    0 1 2 2

          0 0 0 0    0 0 0 0    0 0 0 0    0 0 0 0
          0 0 2 0    0 0 0 0    0 0 0 0    0 0 0 0

```

0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
-> 0 0 0 0	-> 0 0 0 0	-> 0 0 0 0	-> 0 0 0 0
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 2 0 1	0 2 2 1	0 0 0 1	0 0 0 0
0 1 2 2	0 1 2 2	0 1 0 0	0 1 0 1

Problem B. QueryreuQ

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

A string is **palindrome**, if the string reads the same backward and forward. For example, strings like "a" "aa" "appa" "queryreuq" are all palindromes.

For given empty string S , you should process following two queries :

1. Add a lower case alphabet at the back of S .
2. Remove a character at the back of S .

After processing a query, you should count the number of **palindrome substring** in S . For string S and integers i, j ($1 \leq i \leq j \leq |S|$), $S[i, j]$ represents a substring from i th to j th character of S . You should print out the number of integer pairs (i, j) where $S[i, j]$ is palindrome.

Input

Input consists of two lines.

In the first line, Q , the number of queries is given. ($1 \leq Q \leq 10,000$)

In the second line, the query is given as string of length Q . i th character K_i denotes the i th query.

K_i is '-' or lower case alphabet ('a', 'b', ..., 'z') (without quotes).

If the character is '-', you should remove a character at the back of S . If the character is lower case alphabet, you should add a character K_i at the back of S .

It is guaranteed that length of S is always positive after the query.

Output

Print out Q space-separated integers in the first line. i -th integer should be the answer of the i th query.

Example

standard input	standard output
17 qu-uer-ryr-reu-uq	1 2 1 2 3 4 3 4 5 7 5 7 9 11 9 11 13

Problem C. Recipe

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

Jaemin likes cooking. He wants to devise several recipes for N days. He devises a recipe in the following order.

1. Buy ingredients at a market and put them in a refrigerator.
2. Think of a recipe.
3. Take out ingredients from the refrigerator and cook.

He can devise a recipe with such a simple way. He wants to cook food as delicious as possible.

There are new ingredients in the market daily. The ingredients sold on i -th day have freshness F_i . The freshness F_i of the ingredients in the refrigerator decreases by 1 everyday. If the ingredients are in the refrigerator, he doesn't buy more ingredients until he cooks with them.

He has cooking skill C_i on i -th day. His cooking skill advances everyday, so $0 < C_i \leq C_j$ for all $i < j$. If he takes out the ingredients whose freshness is F from the refrigerator and cook with cooking skill C , a dish with a flavor of $F \times C$ is made. When he cooks, he invites his friend Jaehyun, who is very hygienic, so Jaemin hopes that the ingredients in the refrigerator have freshness greater than or equal to L_i . If the ingredients don't satisfy the requirement, Jaemin cannot cook that day. Jaehyun's requirement varies everyday, and the requirements for N days are given as L_1, L_2, \dots, L_N .

After he cooks a new dish, he goes to the market the next day to buy new ingredients and think of a new recipe again. Everyday, he may go to the market to buy ingredients, cook, or do nothing for devising a recipe (It is also possible to cook on the day he purchases the ingredients). On the first day, there aren't any ingredients in the refrigerator, he goes to the market to buy some ingredients. On the N -th day, he must cook and empty the refrigerator. Let's find the maximum sum of a flavor of the dishes he cooks. If it is impossible to empty the refrigerator on the N -th day because of Jaehyun's particular requirements, print out "Impossible"(without quotes).

Input

Input consists of four lines.

First line contains N . ($2 \leq N \leq 250,000$)

Second line contains N space-separated integers F_1, F_2, \dots, F_N . ($0 < F_i \leq 50,000$)

Third line contains N space-separated integers C_1, C_2, \dots, C_N . ($0 < C_1 \leq \dots \leq C_N \leq 10,000$)

Fourth line contains N space-separated integers L_1, L_2, \dots, L_N . ($0 \leq L_i \leq 50,000$)

Output

Print the maximum sum of flavors of the dishes Jaemin cooks. If it is impossible to empty the refrigerator on the N -th day, print out "Impossible"(without quotes).

Examples

standard input	standard output
3 10 1 1 1 2 3 1 1 1	24
3 10 1 1 1 2 3 10 10 10	Impossible
10 3 4 1 5 9 2 6 5 3 5 10 11 12 13 14 15 16 17 18 19 1 4 1 4 2 1 3 5 6 2	526

Problem D. Segmentation

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

ZOYI is developing a tool called Channel which offers a tool to talk with online users in the site. Recently, ZOYI introduced a RF(Recency / Frequency) Model to distinguish users who are using the Channel and decided to classify the users through following calculations.

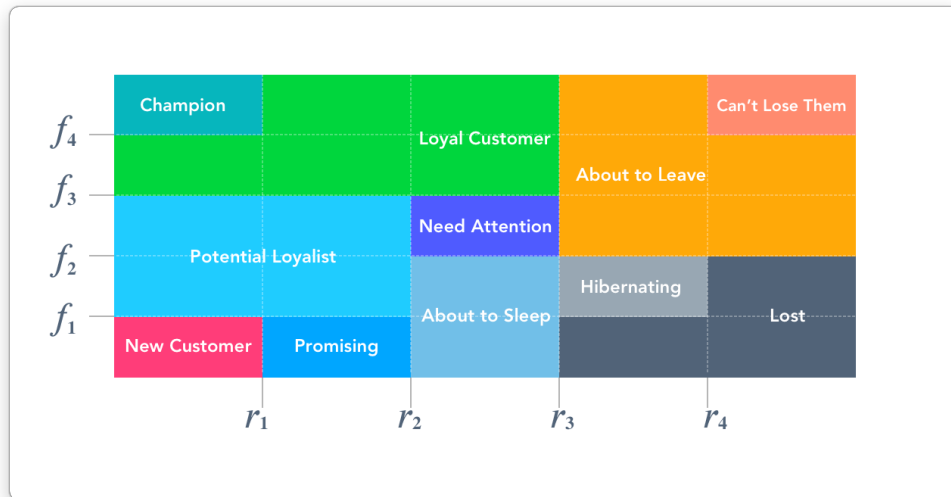


Figure : Distinguishing users in RF Channel. Horizontal axis represents Recency, while vertical axis represents Frequency.

($0 < f_1 < f_2 < f_3 < f_4$, $0 < r_1 < r_2 < r_3 < r_4$, all f_i and r_i are integers.)

x axis represents Recency and y axis represents Frequency. All online users are given values r , f by their connection record, and are classified into one of twelve conditions shown below.

- "New Customer"
- "Promising"
- "About to Sleep"
- "Hibernating"
- "Lost"
- "Potential Loyalist"
- "Need Attention"
- "About to Leave"
- "Champion"
- "Loyal Customer"
- "Can't Lose Them"
- "None"

Among those, "None" means the user has no connection record to the server. If (r, f) is located on two or more classification boundaries, it follows the classification of $(r - 0.5, f - 0.5)$. For example, if the value of (r, f) is (r_4, f_2) it is classified as "Hibernating" while if the value is (r_3, f_4) , it is classified as "Loyal Customer".

You want to investigate users' statuses who are interested in RUN, so you are trying to install the program in the following way :

- r : if the current time is t , $t -$ (most recent access time)
- f : number of visited times

Given events of site users, make a program which classifies the users following the given picture above.

Input

First line contains four space-separated integers r_1, r_2, r_3, r_4 . ($0 < r_1 < r_2 < r_3 < r_4 \leq 10,000$)

Second line consists four space-separated integers f_1, f_2, f_3, f_4 . ($0 < f_1 < f_2 < f_3 < f_4 \leq 10,000$)

Third line contains a single integer N . ($1 \leq N \leq 100,000$)

Next N lines contains events in time order, where i th element represents the event held at time i .

Each event is given as space-separated A and B , where B is the username which contains no whitespace with at most 10 alphabets. A has a value of 1 or 2, where 1 means the user entered the site while 2 means you should print how the user is classified.

Output

For events where A is 2, print how the user is classified in each line (without quotes).

Example

standard input	standard output
1 2 3 4	New Customer
1 2 3 4	Potential Loyalist
8	Need Attention
1 RUN	
1 Alex	
2 Alex	
1 RUN	
1 RUN	
1 Alex	
2 Alex	
2 RUN	

Note

The connection status of Alex is $f = 1$ (first visit), $r = 1$ (time 3 - 2 = 1) at time 3. Thus, Alex is classified as "New Customer".

At time 7, the connection status of Alex is $f = 2$ (second visit), $r = 1$ (time 7 - 6 = 1). Thus, Alex is classified as "Potential Loyalist".

At time 8, the connection status of RUN is $f = 3$ (third visit), $r = 3$ (time 8 - 5 = 3). Thus, RUN is classified as "Need Attention".

Problem E. Touch The Sky

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes



Figure: The house floats up in the sky by balloons. This picture is also used in 2018 KAIST RUN Spring Contest poster.

In the year 2117, Professor Jaemin Yu developed a linear-time algorithm for TSP(Traveling Salesperson Problem). Not long after that happened, all computer systems were destroyed, and nuclear weapons demolished all the lands. You, a great computer expert, also lost your job. With a great despair, you lost your meaning of life long ago. All those things that made your heart beat – where had they gone? After questioning yourself again and again, your conclusion is ...

“If I go to KAIST where I started my first ICPC, can I find a meaning of my life?”

All transportations were destroyed, but you were an avid ICPC participant, and you collected a lot of century-old balloons in Korean Regionals. If you could float a house with some of those balloons...

Currently you have N balloons, and you are trying to float the house into the sky by attaching balloons on the rooftop. Every balloon have altitude limit L_i and capacity D_i , which indicates you can blow balloons in altitude at most L_i , and the balloon busts after increasing the altitude by D_i .

Your journey starts at altitude 0. If you have more than 1 balloons enlarged, then the house will ascend too fast. Thus, you will blow one balloon and attach it at the rooftop, increase the altitude until the balloons bust, blow the other balloon and attach it to increase the altitude... to make your house float. For convenience, you may assume that **balloons can only increase the altitude**.

You don't care about your final altitude, but a balloon can move a fixed amount of distance. Thus, you want to bust as many balloons as possible. You want to calculate a maximum number of balloons you can bust, and check if you can make a journey to KAIST. Let's see whether your 100-year-old ICPC experience can help on this problem!

Input

The first line contains N , the number of balloons. ($1 \leq N \leq 250,000$)

In next N lines, the altitude limit of i -th balloon L_i , and capacity of i -th balloon D_i are given as two space-separated integers. ($0 \leq L_i \leq 10^{15}$, $1 \leq D_i \leq 10^9$)

Output

Print the maximum number of balloons you can bust.

Examples

standard input	standard output
3 30 10 30 20 30 30	3
4 0 10 2 4 5 3 8 2	3

Problem F. United States of Eurasia

Input file: `standard input`
Output file: `standard output`
Time limit: 15 seconds
Memory limit: 1024 megabytes

In the year 5013, jh05013 conquered the Eurasian Continent and found “jh land”. “jh land” encompasses the whole continent, and jh05013 aims to divide the country into “districts” to govern them efficiently, since it is extremely large and populated. There are N houses in “jh land” and each house is located at (x_i, y_i) of a 2D Cartesian coordinate system. jh05013 keeps the following conditions when dividing them into districts:

- Condition 1. Each district contains houses whose x coordinate is in a certain range. (A district might contain all or none of the houses)
- Condition 2. Each house is contained in exactly one district.
- Condition 3. There are at most K districts.

There are diverse races, religions, and nations in the Eurasian Continent. To avoid disputes, we must reduce the “splittedness” of districts. The splittedness of a district is defined as the distance between the farthest pair of houses contained in the district. The distance is measured in Euclidean metric. Help jh05013 minimize the maximum splittedness among districts.

Input

In the first line, two space-separated integers N , K are given. N denotes the number of houses and K denotes the number of districts. ($1 \leq K \leq N \leq 250,000$)

In the next N lines, two space-separated integers x_i , y_i are given. They denote that a house is located at (x_i, y_i) . ($0 \leq x_i, y_i \leq 10^9$) Every given location is distinct.

Output

When the minimum of maximum splittedness among districts is M , print M^2 .

Examples

standard input	standard output
4 2 101 100 2 5 100 101 4 3	8
4 4 3 1 4 1 5 1 9 2	0

Problem G. Voronoi Diagram

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

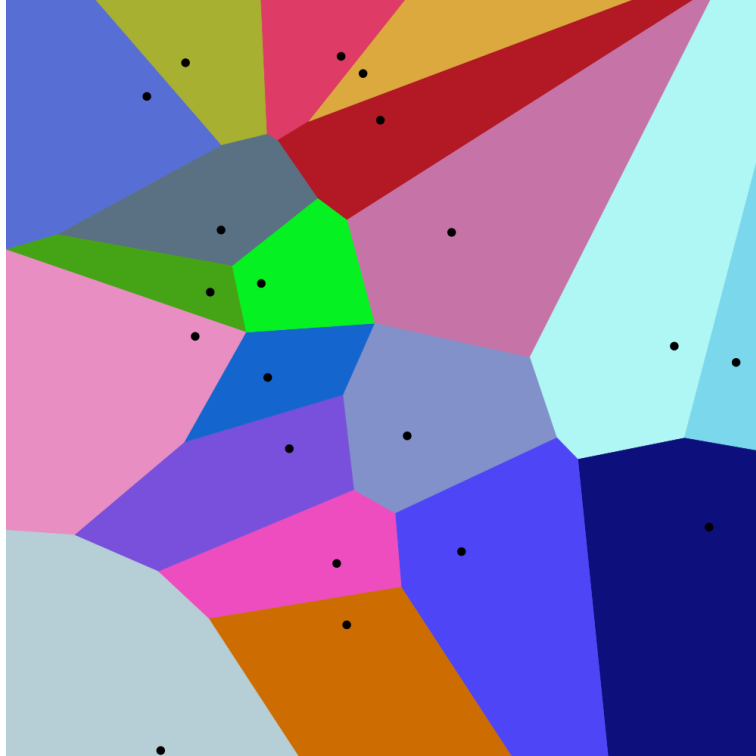


Figure: Voronoi Diagram made with 20 points, with respect to Euclidean metric. Source: *Wikipedia*

In the Cartesian coordinate, we define the **Voronoi Diagram** of nonempty point set of size n as a diagram that divides the plane by the criteria "which point in a set is closest in this location?". For example, in the picture above, every location over the plane is colored by the closest black point with such location. There is an algorithm that computes Voronoi Diagram in $O(n \log(n))$, but this is infamous to be very complicated and hard.

After failing to solve a problem about Voronoi Diagram in an important competition, Minkyu was shocked, and he started living with alcohols everyday! One sunny afternoon, Minkyu was drinking beers just like the other day, and found an ingenious algorithm for solving Voronoi Diagrams! Before writing a paper about it, Minkyu wants to set a problem which requires this algorithm, to prevent any full scorers in 2018 KAIST RUN Spring Contest.

Why is Minkyu's algorithm for Voronoi Diagram great? Traditional algorithm for Voronoi Diagrams works only in cartesian coordinate, but Minkyu's algorithm works on more generalized structure – the "graph". Consider a connected graph with N vertices and M edges with positive weight. When you are given a nonempty vertex subset of size K , the "Voronoi Diagram" of this point set divides all location in the edges by the criteria "which vertex in a set is closest in this location?" If there is more than one points with equal distance, the one with smallest vertex number is considered closer.

You are given a weighted connected graph and a nonempty vertex subset of size K . For each vertex, you should calculate the total length of edges that is "closest" to the given vertex. Solve this problem, and write the paper faster than Minkyu, to scatter his high hopes!

Input

In the first line, the number of vertices N , and the number of edges M are given as two space-separated

integers. ($1 \leq N, M \leq 250,000$)

In the next M lines, two endpoint of the i -th edge s_i, e_i , and the weight of i -th edge w_i is given as three space-separated integers. ($1 \leq s_i, e_i \leq N, 1 \leq w_i \leq 10^9$)

In the next line, the size of vertex subset K is given. ($1 \leq K \leq N$)

In the next line, K distinct integer a_i is given in increasing order. Each integer denotes the number of vertex in the subset. ($1 \leq a_i \leq N$)

You can assume that the given graph is connected. In other words, there exists a path from any vertex to the any other vertex.

Output

In K lines, print one decimal number for each line. In i -th line, print the sum of length which considers vertex a_i as the closest.

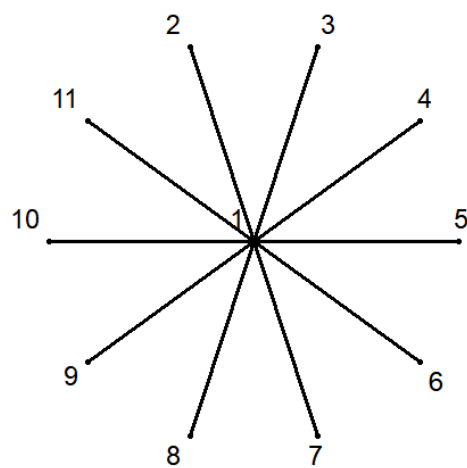
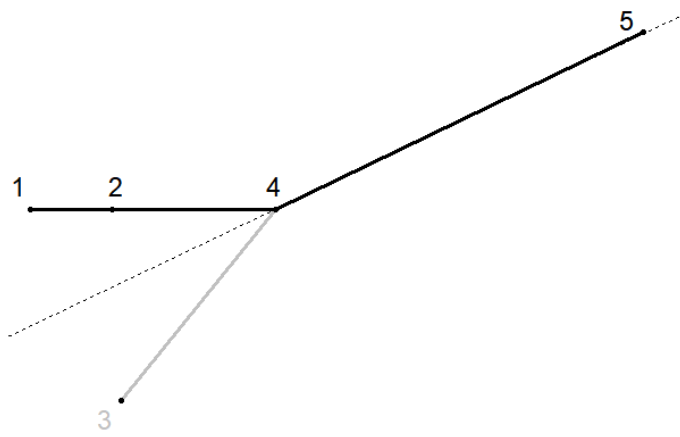
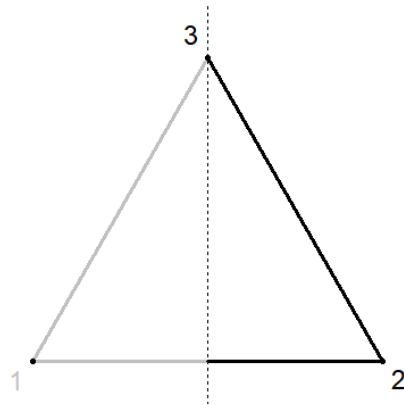
Every output numbers should be exactly rounded to the first digit after the decimal point (see the sample input/output for clarification). In accordance to the recent ACM-ICPC World Finals trend (which requires high-precision floating point management), **no precision error is allowed**.

Examples

standard input	standard output
3 3 1 2 5 2 3 5 3 1 5 2 1 2	7.5 7.5
5 4 1 2 10 2 4 20 3 4 30 4 5 50 2 1 3	80.0 30.0
11 10 1 2 1000000000 1 3 1000000000 1 4 1000000000 1 5 1000000000 1 6 1000000000 1 7 1000000000 1 8 1000000000 1 9 1000000000 1 10 1000000000 1 11 1000000000 1 1	10000000000.0

Note

These are the pictures corresponding to each sample test.



Problem H. Winter Olympic Games

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 1024 megabytes



Figure: “Soohorang Not related to this problem, but included just because it’s cute.

2018 RUN@KAIST Winter Curling Competition women’s finals game is now ongoing. On the frozen “duck pond” of KAIST, Korean women curling team is having a fierce competition with team from country *Jwepan*!

There are N curling stones on the “duck pond”. As the competition is really fierce, every stone is placed in a line from a mark. The leftmost stone is closest from the mark, while the rightmost stone is farthest from the mark. Stones are either from Korean team (‘1’), or from *Jwepan* team (‘0’). Those arrangement of stones can be represented with length N binary sequence.

After the end of Pyeongchang Olympics, Korean team had gone through intensive training. Now with some shoutings(?), team member “Youngmi who carries the curling stone, can bounce away some consecutive stones and place her stone in that position. Formally, Korean team can pick any subsegment in a binary string (which can be empty), and replace it into a single digit “1”.

Korean team is a master in a curling strategy, and they knew the best strategy for one turn is to make the string **lexicographically maximal**! For the fast decision making in this game, they want to find a fastest algorithm which can find this. Help the Korean team to win the competition!

String $s = s_1s_2 \cdots s_n$ of length n is **lexicographically larger** than string $t = t_1t_2 \cdots t_m$ of length m , if one of the following holds:

- There exists some i such that, $s_1 = t_1, s_2 = t_2, \cdots, s_{i-1} = t_{i-1}$, and $s_i > t_i$.
- $n > m$ and, $s_1 = t_1, s_2 = t_2, \cdots, s_m = t_m$.

Input

In the first line, N , the number of stones is given. ($1 \leq N \leq 1,000,000$)

In the second line, A single binary string of length N , which consists of '0' or '1' is given. This string indicates the owner of each curling stone, in the order of distance from the mark. There are no quotes or blanks given in the string.

Output

Print two integer S , L . This means that Youngmi removed L stones after S th character. If there is more than one correct answer, print any. ($0 \leq S, L \leq N$)

Examples

standard input	standard output
8 10101101	1 3
5 11111	0 0

Problem I. Xtreme NP-hard Problem?!

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 1024 megabytes

Caution! This problem turned out to be NP-hard. But since there were no rules against writing an NP-hard problem, we decided to leave this problem here.

There is a bidirectional graph consisting of n vertices and m edges. The vertices and edges are numbered from 1 to n and 1 to m respectively, and the weight of edge i is w_i . ($1 \leq i \leq m$) Given a natural number k , find the length of the shortest simple path that starts from vertex 1 and ends at vertex n , and consists of k edges. A simple path is a path that does not visit same vertex twice, and length of a path is the sum of weight of edges that consists the path.

Input

In the first line, three space-separated integers n , m , k are given. ($2 \leq n < 10^6$, $1 \leq m$, $k < 10^6$, $\min(n, m, k) \leq 5$)

In the next m lines, three space-separated integers x_i , y_i , w_i are given. They denote that edge i is connecting vertex x_i and vertex y_i , and has weight w_i . ($1 \leq x_i, y_i \leq n$, $1 \leq w_i \leq 10^8$)

No loops or multiple edges are given.

Output

Print the length of the shortest simple path that starts from vertex 1 and ends at vertex n , and consists of k edges. If there is no such path, print -1.

Example

standard input	standard output
6 6 3 1 2 3 2 3 1 3 6 4 1 4 1 4 5 5 5 6 9	8

Problem J. Yut Nori

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

Yut Nori (Yut Game) is one of the most famous board game which uses Yuts. Yut Nori consists of 4 Yuts, one Yut board, and 8 tokens: 4 for each team. The Yut board has 29 stations. Each station has its own name. (Figure 1)

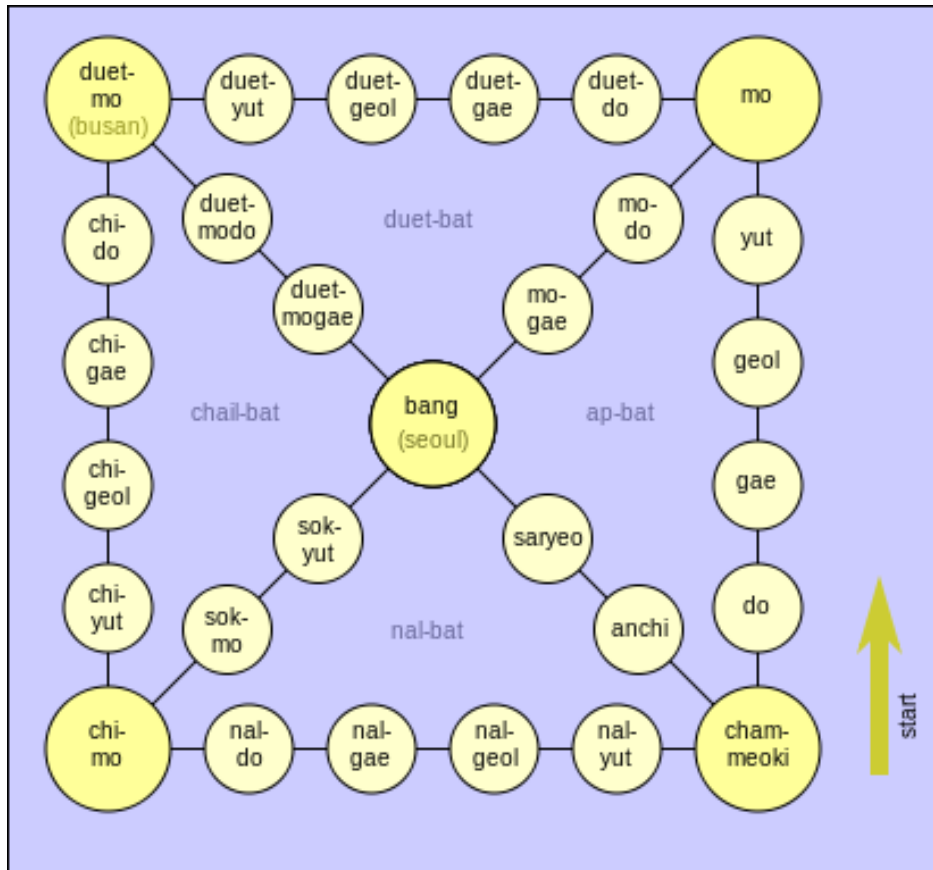


Figure 1: Yut board. There is name for each stations.

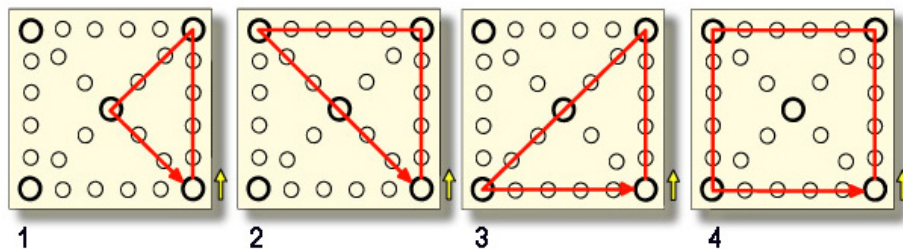


Figure 2: possible four routes.

The goal of the game is starting from cham-meoki, to walk through the board and return to cham-meoki and escape.

At the beginning of each turn, one person:

- throws 4 Yuts;

- determines the number of steps to progress based on the Yuts;
- choose a token and walks through the board.

Each Yut has a front-side and a back-side. When a Yut is thrown, its front-side or back-side is shown. The number of steps is the number of front-sides, except when every side is a back-side, the number of steps is 5.

Each token starts at cham-meoki (it is not placed actually, but it is assumed that the token is placed there), and follows the Yut-gil (route).

Basically, tokens move through the fourth route (Figure 2). For example, if a token walks two steps from yut, it reaches duet-do. But in special cases, they go through the first, second, or third route. If a token starts from corner (mo, duet-mo, bang, chi-mo, or cham-meoki), it chooses the faster way to cham-meoki.

More formally, if a token starts moving at mo, the route changes to the third one. For example, if it walks one step from mo, it goes to mo-do; two steps from mo, it goes to mo-gae. If it walks one step twice from yut, it goes to mo-do, but if you walk two steps from yut in one turn, it goes to duet-do.

Also, if it start from duet-mo, it follows the second route, so walking one step leads to duet-modo, while walking two steps leads to duet-mogae.

Moreover, if a token following the third route starts from bang, then it starts to follow the first route. Walking one step leads to saryeo; two steps, anchi.

Moving through these routes, a token escapes the board after it returns to cham-meoki then takes another step. This does not require separate turns. For example, if you walk two steps from nal-yut, it arrives at cham-meoki then escapes. Similarly, taking 3, 4, or 5 steps from nal-geol counts as an escape.

Until now, this is just a game about throwing Yuts to escape. To make the game more fun, there are two more rules.

The first rule is moving together. If there are two or more tokens possessed by the same player at the same station, moving just one of them will cause all other tokens to move together.

The second rule is catching a token. If opponents' tokens are placed at the next destination of your token, all the opponents' tokens at that station move to the initial state.

You are given a task to develop the Yut Nori game. Given a series of turns, output the final state of the board.

Input

The first line of the input consists of N , the number of the turns. ($1 \leq N \leq 100$)

In the i -th line of following N lines, the information of the i -th turn. It is represented by a token and Yuts.

The information for the token is given as one character, one of "ABCDabcd". The information for Yuts is given as a string with four characters, each character being either 'B' for back-sides, or 'F' for front-sides. Two pieces of information are space-separated.

Upper-case letters represent tokens in one team; lower-case letters represent tokens in the other team.

Every move is valid; escaped token will not be given again at board.

Output

The basic Yut board is given as the following 32×32 string. By the technical difficulty, it is posted as an attachment, with sample input/output. (If you have better idea, please let us know.)

This string consists of "\. | space, and newline character. (quotation marks are not included.) This string denotes the Yut board, and each station is represented as 2×2 '.' characters. From the upper-leftmost character, the stations are duet-mo, duet-yut, duet-geol, duet-gae, duet-do, mo, duet-modo, mo-do, chi-do,

yut- duet-mogae, mo-gae, chi-gae, geol, bang, chi-geol, gae, sok-yut, saryeo, chi-yut, do, sok-mo, anchi, chi-mo, nal-do, nal-gae, nal-geol, nalyut, and cham-meoki, read row major order.

You should indicate where each token is located. Replace one of the ‘.’ characters with the token in the station.

A or a should replace the upper-left ‘.’, B or b should replace the upper-right ‘.’, C or c should replace the lower-left ‘.’, and D or d should replace the lower-right ‘.’ of the station.

File URL: <http://run.kaist.ac.kr/contest18s/Y.zip>

Basic Yut board: `yut_board.txt`

Sample Input 1: `samp1.in`, `samp1.out`

Sample Input 2: `samp2.in`, `samp2.out`

Problem K. Zigzag

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 1024 megabytes

A sequence is called “**Zigzag**” if no three of its consecutive elements are monotone.

More formally, if sequence A of length N is Zigzag if, for all i ($1 \leq i \leq N - 2$), neither $A_i \leq A_{i+1} \leq A_{i+2}$ nor $A_i \geq A_{i+1} \geq A_{i+2}$ holds.

For given sequence A of length N , you should find a longest subsegment of A which is a Zigzag sequence.

Sequence B of length M is subsegment of sequence A of length N if, for some i , $B_1 = A_i$, $B_2 = A_{i+1} \cdots$, $B_M = A_{i+M-1}$ holds.

Input

Input consists of two lines.

The first line contains integer N , length of sequence A . ($3 \leq N \leq 5,000$)

The second line contains space-separated N integers. i th number is A_i . ($1 \leq A_i \leq 10^9$)

Output

Print out the length of longest subsegment of A which is a Zigzag sequence.

Example

standard input	standard output
5 1 3 4 2 5	4