Problem A. Luggage Distribution

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

All provident tourists know about low-cost airlines. Those who used them at least once also know that their airfare usually includes neither lunch nor luggage.

Naturally, tourists don't want to overpay luggage fee, therefore they distribute their luggage so that it satisfies all airline requirements.

A certain tourist prepaid three luggage slots. Then he calculated the number of different ways K to distribute weight N among these three slots so that all luggage is packed, none of the slots are empty and all weights are integers. The order of the slots does not matter, so, for example, two distributions $2,\ 2,\ 1$ and $2,\ 1,\ 2$ are considered the same.

After landing, the tourist realised that the airline lost all his luggage! And while filling in a lost and found application, he found out that he does not remember the weight of each slot. He remembered only the fact that the number of ways K to pack the luggage was at least L and did not exceed R.

You need to calculate number of possible integer total weights N that could have been in the tourist's luggage.

Input

The single line of input holds two integer numbers L and R ($1 \le L \le R \le 10^{17}$): the minimum and maximum number of ways to distribule the luggage in three slots.

Output

Output a single integer: the number of possible total weigths N that could have been carried by the tourist.

Examples

standard input	standard output
2 4	3

Note

In the example, the tourist could have carried the total weight of 5, 6 or 7.

Problem B. Passport Control

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

No one is safe from losing his passport. But even if the passport is lost, it is still not the end of the world. To replace it, you just need to present another picture ID and wait for a few weeks. But what should one do if the passport is suddenly lost during a vacation in an exotic country?

The migration service of a certain insular country keeps track of every single tourist entering the country. Each tourist is assigned a sequence of K fields which all together create a tourist ID. When a tourist crosses the border, the fields are filled according to the data in the passport. The tourist ID is a complex concept to handle for migration officers, so one can not be sure which fields will contain one's name or surname, or whether they will be present in it at all. It may even happen that migration officers fill some fields with arbitrary strings by mistake. The only thing known for sure is that none of the fields are empty and all fields contain only lowercase English letters.

A certain tourist tries to find his tourist ID by issuing queries to the migration service database using a query form. Each query consists of K strings. The answer for the query is the list of all tourist IDs in the database for which each of the K strings is a prefix of the corresponding field of that tourist ID.

Initially, all K strings of the query form are empty. To make the next query, the tourist modifies the previous one in one of the two following ways:

1. `<i> + <c>`: add character `<c>` to the end of string number `<i>`,
2. `<i> -`: remove the last character of string number `<i>`.

For each of the Q queries, you need to calculate how many tourist IDs match this query.

Input

The first line of the input holds an integer number N ($1 \le N \le 100\,000$) which is the number of tourist IDs in the database and an integer number K ($1 \le K \le 3$) which is the number of fields in each tourist ID.

Then follow N tourist IDs. Each tourist ID is given on K consecutive lines, each of which contains the respective field of the tourist ID. It is guaranteed that the lines are non-empty and contain only lowercase English letters. The total length of all given strings does not exceed 100 000.

The next line holds a single integer Q ($1 \le Q \le 100\,000$) which is the number of queries. Each of the next Q lines describes a modification to the query form. Each modification has one of the two possible forms: either "`<i> + <c>`" or "`<i> -`". It is guaranteed that before each modification of the second type, the respective field is not empty.

Output

For each query, print the number of matching tourist IDs in the database on a separate line.

Examples

standard input	standard output
5 2 abacaba ccad abadaba cbad abbbaaa cbab bbbba ccc abacabs fuuuu 15 1 + a 2 + c 2 + c 1 + b 2 - 2 + b 1 + a 2 - 2 - 1 - 1 - 1 - 1 + c 1 - 1 + b	4 3 1 1 3 2 1 2 3 4 4 5 0 5 1

Problem C. Tropical Stonehenge

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

One of the oldest existing buildings on Earth is the English Stonehenge. Nobody knows its purpose: some say it was an ancient sanctuary, other say a huge observatory. But recently, archeologists found another piece of the puzzle: an ancient tropical stonehenge.

The tropical stonegenge, unfortunately, was severely damaged by lianas. Fortunately, scientists managed to identify places of all stones of the building except one. The only thing that can help to find the position of the last stone is an ancient manuscript found nearby.

The manuscript tells the following.

1. Tropical slonehenge is a convex polygon with N vertices with an ancient stone in each vertex.
2. No three vertexes lie on the same straight line.
3. All vertices have integer coordinates, the coordinates do not exceed 10^6.
4. The area of the polygon is equal to S.

Knowing the coordinates of $N-1$ stones and the total area S, find the coordinates of the last stone.

Input

The first line of input holds an integer number N ($4 \leq N \leq 10^5$) and the area of the polygon S. The area is a real number given with exactly one decimal digit ($1 \leq S \leq 4 \times 10^{12}$). The next $N-1$ lines hold coordinates of the known stones in the form X Y ($|X|, |Y| \leq 10^6$). The stones are listed in counterclockwise order of polygon traversal, excluding the missing one.

Output

Output the coordinates of the missing stone in the form X Y. It is required that the coordinates are integers and $|X|, |Y| \leq 10^6$. In case of multiple solutions, output any one of them. It is guaranteed that the judges' inputs are constructed in such a way that there is at least one possible solution satisfying the constraints above.

Examples

standard input	standard output
4 4.0 0 0 2 0 0 2	1 -2
5 10.0 5 3 1 3 2 1 4 1	1 5

Problem D. Toll Road

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	128 mebibytes

Exactly N years ago, a new highway between two major cities was built. The highway runs from west to east. It was planned as one big segment of a toll road. However, new highway was not popular: drivers still used free roads.

After analyzing the situation, the administration decided to perform a marketing trick to increase the popularity of the new highway.

The highway was divided into segments. Initially, there was only one segment. Every odd year, each of the existing segments was divided into two new segments with lengths divided as $X : Y$. This means that the length of the western segment relates to the length of the eastern segment as X relates to Y. Every even year, each of the existing segments was also divided into two new segments, but this time, the ratio was $Y : X$. After each division, the first of the two resulting segments was declared a free road and the second one a toll road. Each year, the segments are numbered from 1 from west to east. For simplicity, X and Y are positive integers which sum up to exactly 100.

As a result, the administration was able to significantly increase the income: the drivers started to drive on free segments and did not dare to turn away at the sight of the next toll segment. But after N years, the plan of the highway became so complex that it is now hard to calculate the exact lengths of the segments.

Knowing the total length of the highway P, one can calculate the length L_k of the segment with number k using the formula

$$L_k = P \times (X/100)^{A_k} \times (Y/100)^{B_k}$$

for some integers A_k and B_k. Here, A_k is the number of years in which, during the division, this segment was in the part proportional to X, and B_k is the number of years when it was in the part proportional to Y.

You need to answer to the queries containing numbers K_i of segments. To answer each query, you must print the values of A_{k_i} and B_{k_i} for the corresponding segment.

Input

The first line of the input contains three integers: the number of years N ($1 \le N \le 10^{18}$) that have elapsed since the highway was built, followed by the percentages X and Y ($1 \le X, Y \le 99$, $X + Y = 100$) used each year to divide the segments.

The second line contains one integer: the number of queries Q ($1 \le Q \le 10^4$).

The following Q lines contain queries. Each query is an integer K_i ($1 \le K_i \le 10^{18}$), the number of some segment. It is guaranteed that the segment with such number exists.

Output

For each query, print two integers A_{k_i} and B_{k_i} on a separate line.

Examples

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>3 25 75 4
1
3
5
8</pre></td><td><pre>2 1
3 0
1 2
1 2</pre></td></tr>
</table>

Problem E. The Street Escalator

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

The longest escalator in the world is located in Hong Kong. Unlike conventional escalators in the subway, this escalator goes right down the street and was built in order to reduce the number of traffic jams in the city. Now it carries tens of thousands of people every day.

Travelling in Hong Kong, a certain tourist decided to use the escalator to go down. He was surprised when it turned out that the escalator is going only up. The escalator was empty, so the tourist decided to try to walk down to the bottom. But as soon as the tourist began his movement, the passengers began to enter the escalator from the bottom...

The escalator runs upwards at a constant speed; the tourist moves downwards with constant speed which exceeds the escalator's speed by absolute value. At any given time, each person (our tourist or some other passenger) is either on the left side of the escalator or on the right side.

For each passenger who will meet the tourist before the end, we know the side initially selected by him and the probability that he changes the side: each second, the passenger changes the side he is currently standing at to the opposite one with that probability. Moreover, for each passenger, we know the time interval T_i in seconds between entering an escalator and meeting with the tourist. Thus, during the time interval of T_i seconds, the passenger will make a decision whether to change the side exactly T_i times.

The tourist starts moving on the left side and will change the side only if otherwise he will run into a passenger.

You need to find the expected number of times that the tourist will change the side before he will finish his journey at the bottom.

Tourist can be considered thin enough so that he can pass between passengers who stand on adjacent stairs at different sides. The time used by the tourist to change the side is negligible.

Input

The first line of the input contains the integer N ($1 \le N \le 10^5$) which is the number of passengers coming to the bottom of the escalator.

Each of the following N lines contains a description of the passenger which consists of the following three pieces. First comes the time interval T_i in seconds ($1 \le T_i \le 10^6$) between the moment when the passenger entered the escalator and the moment he meets the tourist. Then goes a character C_i which is either "`L`" if the passenger starts at the left side or "`R`" if he starts at the right side. It is followed by the probability of side change P_i ($0 \le P_i \le 1$) given with no more than three digits after the decimal point. All values of T_i in the input are pairwise different.

Passengers in the input data are listed in the order in which they enter the escalator, that is, in decreasing order of T_i.

Output

Print the expected number of side changes made by tourist with absolute or relative error no more than 10^{-6}.

Examples

standard input	standard output
2 5 L 0.5 1 R 0.4	1.000000000000

Problem F. Mexico

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2.5 seconds
Memory limit:	128 mebibytes

Mexico is one of the most interesting countries for tourists. Endless sandy beaches, unique culture, ancient Indian temples... Beautiful nature: tropical forests, mountains, deserts... And, of course, cacti!

A certain tourist arrived in Mexico and decided to explore the country. For this, he rented a car and made a plan for visiting the cities of the country on K days ahead.

We know that in Mexico, there are N towns connected by M roads. Each road connects two different cities, and no two roads connect the same pair of cities. All roads have the same length and are bidirectional, and it is possible to travel by road network from each city to any other.

In addition, if you look at a road network of Mexico as a graph where vertices are cities and roads are edges, this graph will be a *vertex cactus*: a connected undirected graph in which each vertex belongs to at most one simple cycle.

The tourist begins his journey through the cities of Mexico in city 0. Following the plan, on each day, he chooses a city for the excursion and drives to it from the city in which he was the previous day by the shortest route. If the city does not change, the tourist continues the excursion in the current city.

Each time when the tourist is driving somewhere, he should refuel the car in the city that is closest to the middle of his path. If there are several such cities, any of them can be selected.

Given the road map of Mexico and the numbers of cities that tourist plans to explore, determine the cities, in which he will refuel his car.

Input

The first line of the input contains two integers: the number of cities N ($1 \le N \le 10^5$) and the number of roads M.

Each of the next M lines describes one road and contains two integers U_i and V_i ($0 \le U_i, V_i \le N-1$) which are the cities connected by i-th road.

The next line contains an integer K ($1 \le K \le 10^5$) which is the number of cities which the tourist plans to visit.

The next line contains K integers: the numbers of these cities in the order in which he plans to visit them.

Output

Print K integers: the numbers of cities in which the tourist will refuel his car, listed in the order of refueling events. In case there are several solutions, print any of them.

Examples

standard input	standard output
6 6 0 1 0 2 2 3 3 4 2 4 4 5 5 5 1 3 3 4	4 2 0 3 4

Problem G. Free Wi-Fi

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7 seconds
Memory limit:	128 mebibytes

The National Library of France is one of the largest libraries in the world. But not only collections of rare books attracts tourists there. Many of them visit the library just to use the free Wi-Fi...

The library building can be represented as a rectangular field consisting of $N \times M$ cells. Each cell can be one of three types:

1. an obstacle denoted by "`#`",
2. a corridor denoted by "`.`",
3. a room denoted by a digit from `1` to `5` depending on the room type (the numbers may be repeated).

Visitors can pass between any two cells that share a side, but can not pass into any cell with an obstacle.

Wi-Fi coverage in the library is quite weak, so in the rooms and corridors, there are some visitors with gadgets that are walking around the library looking for a place with good signal strength.

A certain tourist came to the library with his friends to use the free internet, and after a long search, he found a place with good signal strength. Due to poor knowledge of French language, the tourist had troubles to identify this location. Nevertheless, he decided to tell his friends about this place and went out in search of them. To remember his path, he decided to record how many cells and in what direction he passed. Also, he decided to record a sequence A of room types he visited, but this part was not treated very carefully, so he may miss the numbers of some visited rooms.

While the tourist was trying to find his friends, he was forced to leave the building because it was late and the library has closed.

The next day, the tourist went to the library again. Given the map of the library, he decided to find the same place again.

The room must satisfy the following criteria:

1. If the tourist starts the movement from this room and strictly follows his records, he will never walk out of the library or go try to move into an obstacle.
2. If the tourist will record the sequence B of visited room types during his way, A must be a subsequence of B.

You need to determine how many rooms in the library will satisfy the criteria above.

Input

The first line of the input contains two integers N and M ($1 \leq N, M \leq 1000$) which are the horizontal and vertical dimensions of the library's map.

Then map itself follows: each of next N lines contains exactly M characters. Valid characters are "`#`", "`.`", "`1`", "`2`", "`3`", "`4`" and "`5`". The lines are ordered from top to bottom, the characters in each line are ordered from left to right.

Then follows a line containing the number of tourist's movements K ($1 \leq K \leq 100$).

The following K lines specify the tourist's movements. Each movement is described by a pair of integers C_i and P_i where C_i is a character indicating the direction of travel ("`L`" for moving to the left, "`R`" to the right, "`U`" up and "`D`" down), and P_i ($1 \leq P_i \leq 999$) is the number of cells travelled in the given direction.

The last line contains a sequence A of digits: the types of rooms listed by tourist on the first day ($1 \le |A| \le 100$, $1 \le A_i \le 5$).

Output

Print one integer: the number of rooms, which satisfy all requirements from the problem statement. Note that it may happen that the tourist made an error recording the sequence A. In that case, the number of rooms satisfying the requirements may be zero.

Examples

standard input	standard output
4 5 22.#. 1211# ..#.1 .#122 2 L 2 U 1 121	1
4 5 ..3#. 213.# .#33# 1214. 3 L 1 R 2 U 1 1213	3

Note

In the first example, only the room $(2, 3)$ in row 2 and column 3 fits.

In the second example, three rooms fit: $(2, 2)$, $(4, 2)$ and $(4, 3)$.

Problem H. Football Bets

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

While traveling to England, it is impossible not to catch the English passion for football. Almost everyone here is involved in football life: some play, others are watching, and the most risky ones bet on the results.

Before the start of the next season of English Premier League, a certain bookmaker office has launched a new option. In order to participate, players must bet a fixed (same for all) amount of money on one of the N teams participating in the championship. All players who guessed a team that will be the champion get back their owh bets. Additionally, they share equally one half of all bets made on the other teams.

During this event, each player made at least one bet on some of the teams, no teams received more than K bets, and at the end of the tournament, the bookmaker's office reported a profit of exactly P percent of the total amount of bets.

Find any distribution of bets between teams which satisfies the above requirements, or determine that no such distribution exists. It is guaranteed, that at least one player bets.

Input

The input contains one line with three integers N, K and P: the number of teams in the football tournament, the maximum possible number of bets on the one team and the profit reported by the bookmaker's office ($2 \le N \le 100$, $1 \le K \le 100$, $0 \le P \le 100$).

Output

If no distribution satisfying the requirements exists, print 0 on a separate line.

Otherwise, on the first line, print one integer W: the number of the winning team ($1 \le W \le N$). The second line must contain N integers $A_1, A_2, \ldots, A_N$ where A_i is the number of bets on i-th team ($0 \le A_i \le K$). Note that the team order is arbitrary, but the number of winning team must fit this order. If there are several solutions, print any one of them.

Examples

standard input	standard output
`4 100 10`	`2` `10 80 5 5`
`2 4 40`	`1` `1 4`
`5 10 23`	`0`

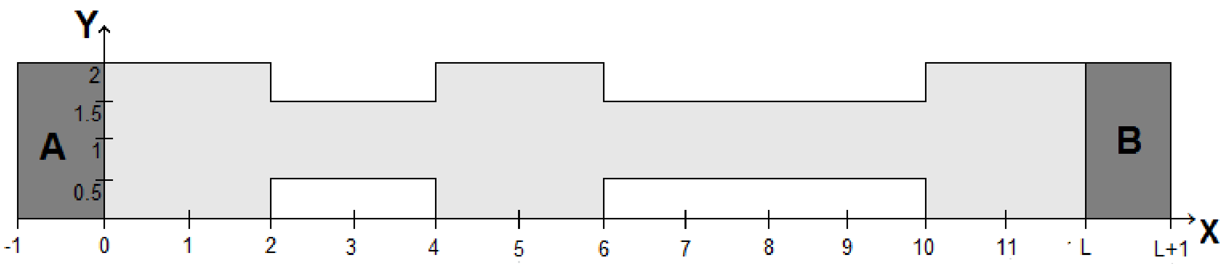
Problem I. Road to Dragobat

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

Dragobat is the highest and the most elite ski-park of Ukraine. Apart from beautiful pine forest and breathtaking view, it is famous for its lacet roads.

The main road starts in the Yasinya-village (A) and ends in Dragobat (B). At the zero moment, two cars start to move from point A to point B and from point B to point A respectively. As soon as a car reaches its destination, it disappears from the road, and another car starts moving from the starting point. There are no other cars on the road. As a result, there are at most two cars on the track at any given moment of time.

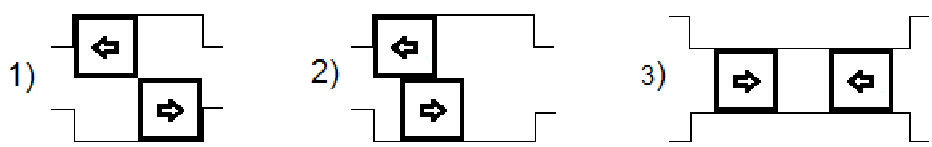
The total length of the road is L. The road is split into one-lane and two-lane segments of given lengths H_i such that $\sum_{i=1}^{K} H_i = L$. The first and K-th segments are two-lane, and each pair of adjacent segments has a different number of lanes. Points A and B are located outside the road. Both one-lane and two-lane segments of the road are adjusted by the center of the road (see the image below).



A car is a square 1×1 which moves by the following rules:

- sides of the car are always parallel to the axes;
- the car which started from point A is always moving by the lower side of the road;
- the car which started from point B is always moving by the upper side of the road;
- the absolute value of X-speed of each car is equal to 0 or 1 at any given moment of time, the speed changes momentarily;
- you can ignore the time spent by cars to move among the Y axis (while changing from one-lane to two-lane and backwards).

If two car-squares intersect (have positive area of their intercestion), an accident is registered.



On the picture above, case 1 won't lead to an accident.

In case 2, an accident can happen if the upper car will start to move on to the one-lane part of the road before the lower car will pass the upper car.

In case 3, an accident will happen if none of the cars change their direction to the opposite.

Because of heavy snowfalls, cars can only see each other if the distance by the X-axis doesn't exceed R. A car can not be seen if it is still in point A or B.

When two drivers can see each other, they try to avoid any possible accident using the following rules:

1. Go on moving with initial speed if it won't lead to an accident.
2. If the current speeds of the cars will lead to an accident, then one of the cars should change its speed, while the second one should keep the initial speed. The driver which adjusts the speed is determined in such a way that the *time of passage* is minimal possible. If the time of passage is equal, the driver which started at point B changes its speed.

The *time of passage* is an interval of time between the moment when drivers first see each other and the moment when the car started in A will be to the right of the car started in B.

Calculate the number of cars which have successfully moved from A to B and from B to A at the moment T. A car arrives to the village only after it completely leaves the road.

Input

The first line of input holds the time T ($1 \le T \le 10^9$) and the visibility radius R ($1 \le R \le 10^6$). The second line holds an odd number K ($1 \le K \le 1000$) which is the number of segments of the road. The third line holds K numbers H_i ($2 \le H_i \le 1000$) which are the lengths of the road segments. All numbers in the input are integers.

Output

Output two integers: the number of cars which have successfully moved from A to B and the number of cars which have successfully moved from B to A at the moment T.

Examples

standard input	standard output
40 2 5 2 2 2 4 2	2 3
30 7 3 2 2 2	3 3

Note

In the first example, cars will arrive at village B at moments 15 and 28, and at village A at moments 13, 26 and 39.

In the second example, cars will arrive at village B at moments 7, 15 and 23, and at village A at moments 11, 19 and 27.

Problem J. What, Where, When? (Added for Division 1!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	128 mebibytes

An intellectual quiz game "What, Where, When?" is very popular among students. The idea of the game is that several competing teams sequentially answer questions using logic, erudition, and often just intuition. The winner is the team which got the maximum number of correct answers.

A team consists of N players. In the game, during the work on one question, there should always be exactly K players at the table. After each question, a team can either leave the squad of players at the table unchanged or make one substitution: remove one player from the table and take a different player there.

Each player has a version of how to answer each question. It is known that a certain team is guaranteed to find the correct answer to the question only if all the players at the table have the correct version.

Suppose that, for each player, we know ahead for which questions this player will have the correct version of the answer. You need to find a starting lineup for the game and the sequence of substitutions in which the team is guaranteed to correctly answer all Q questions in the game, or determine that it is impossible.

Input

The first line of the input contains three integers: the number of players in team the N ($1 \le N \le 100$), the number of players at the table while working on each question K ($1 \le K \le N$) and the number of questions Q ($1 \le Q \le 1000$).

The next N lines contain information about players: i-th of these lines contains information about player i. Each of these lines contains a string S_i ($|S_i| = Q$) such that $S_{i,j} = 1$ if player i got a correct version of answer on j-th question, and $S_{i,j} = 0$ otherwise. Players are numbered by integers from 1 to N.

Output

If no starting lineup and substitution strategy leads to solving all Q given questions, print "`FAIL`" on a line by itself.

Otherwise, print "`WIN`" on the first line. On the second line, print a space-separated sequence A of pairwise distinct integers: the numbers of players for starting lineup ($1 \le A_i \le N$, $|A| = K$). On the third line, print one integer Z: the number of replacements. Then next Z lines must specify replacements in chronological order. Each of those lines must contain three integers: the number of question T_i after which the replacement was made ($1 \le T_i \le Q - 1$) the number X_i of player which leaves the table, and the number of player Y_i which comes to the table ($1 \le X_i, Y_i \le N$, $X_i \ne Y_i$). If there are several possible solutions, print any one of them.

Examples

standard input	standard output
4 3 7 1110111 0111111 1011111 1111100	WIN 1 3 4 3 1 3 2 2 1 3 5 4 1
2 2 4 1100 1010	FAIL

Problem K. Solar Energy (Added for Division 1!)

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

In the recent years, engines running on solar energy are becoming more and more popular. In a few days from now, a plane powered by solar energy alone is going to fly around the world!

The preparations to the flight include a series of test flights. During each of these test flights, the plane takes off at an airport A_i and flies to airport A_{i+1} by the shortest route. Thus the plane always starts the next flight from the airport where it landed on the previous flight, with the exception of the very first test flight.

It is known that the airports involved in the test flights are located along a straight line and numbered from 1 to N along that line. For each pair of consecutive airports, the weather conditions and distance determine the *energy delta* $\Delta Q_{i,i+1}$ which is spent ($\Delta Q_{i,i+1} < 0$) or acquired ($\Delta Q_{i,i+1} \geq 0$) during the corresponding flight in any of the two possible directions. During a flight between non-consecutive airports, the energy delta is the sum

$$\Delta Q_{u,v} = \sum_{i=\min(u,v)}^{\max(u,v)-1} \Delta Q_{i,i+1}.$$

Your task is to find a sequence A containing K numbers of airports visited during the test flights such that the sum of absolute values of energy deltas of all the test flights is minimal possible.

Input

The first line of the input contains three integers: the number of players in team the N ($1 \leq N \leq 100$), the number of players at the table while working on each question K ($1 \leq K \leq N$) and the number of questions Q ($1 \leq Q \leq 1000$).

The first line of input contains the total number of airports N ($2 \leq N \leq 10^5$) and the length K of the required sequence of airports ($2 \leq K \leq N$).

The second line contains $N - 1$ integers $\Delta Q_{1,2}$, $\Delta Q_{2,3}$, $\ldots$, $\Delta Q_{n-1,n}$ which are the energy deltas when traveling between consecutive airports ($|\Delta Q_{i,i+1}| \leq 10^6$).

Output

On the first line, print the minimal possible sum of absolute values of energy deltas for the route you found.

On the second line, print K integers separated by spaces: the sequence A of airport numbers. The airports are numbered from 1 in the order they are specified in the input.

If there is more than one possible solution, print any one of them.

Examples

standard input	standard output
4 3 -3 2 10	3 2 3 1
6 4 -3 2 1 -2 -1	2 2 6 5 3