



## List of Problems

Problem	Time limit	Memory limit	Name
A	2 seconds	256 MiB	Age of Retirement
B	2 seconds	256 MiB	Bridges
C	2 seconds	256 MiB	Crisis
D	3 seconds	256 MiB	Dining Room
E	6 seconds	256 MiB	Exhibition Hall
F	2 seconds	256 MiB	File Sharing
G	3 seconds	256 MiB	tourist
H	2 seconds	256 MiB	Heroes of Money and Magic
I	2 seconds	256 MiB	I, V, X, L, C, D, M Problem
J	2 seconds	256 MiB	Jams
K	3 seconds	256 MiB	King's Palace Garden
L	3 seconds	256 MiB	Lamps of the Mind

Your solution must read the input data from the standard input and write the results to the standard output. Output to the standard error stream is prohibited.

Unless explicitly stated in the problem statements, all input elements may be separated by an arbitrary number of whitespace characters. All input data are correct and satisfy the specification given in the problem statement.

The output of your program must exactly satisfy the output specification in the problem statement.

Each line of input is terminated by end-of-line marker (character with code 0A hex).

## Problem A. Age of Retirement

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

In some country it is decided to start raising the retirement age since January 1<sup>st</sup> 2012. Prior to that date, the retirement age was 60 years.

The raising of the retirement age is carried out as follows. The increase is repeated  $K$  times in total. Each time, the retirement age is increased by  $M$  months. The first increase occurs on January 1<sup>st</sup> 2012, and each of the next  $(K - 1)$  increases happens  $N$  months after the previous increase ( $N > M$ ). For example, let  $M = 3$ . Then citizens born on December 31<sup>st</sup> 1951 will retire on December 31<sup>st</sup> 2011; and citizens born on January 1<sup>st</sup> 1952 will retire on April 1<sup>st</sup> 2012.

The date of retirement for a particular person is the minimum possible date on which that person's age is not less than the retirement age approved on that date.

To determine when a citizen turns  $I$  years and  $J$  months, you need to add  $I$  years and  $J$  months to the date of his birth, leaving the day of the month the same. In case there is no such day in the calendar, the required date is the first day of the next month. Thus, given that  $M = 3$  and  $N = 5$  those born on January 31<sup>st</sup> 1952 will retire on May 1<sup>st</sup> 2012 as well as those born on February 1<sup>st</sup> 1952.

It is required to determine the date of citizen retirement given the date of his birth. It is considered that life expectancy can be arbitrarily large.

A year is leap if the number of the year is divisible by 400 or it is divisible by 4 but is not divisible by 100.

### Input

Three integers  $M$ ,  $N$  and  $K$  ( $0 < M \leq 12$ ,  $M < N \leq 24$ ,  $0 \leq K \leq 60$ ) are given on the first line of input. The second line of input contains a citizen birth date in the format "DD.MM.YYYY" (that is, day, month and year without the quotes and with leading zero for numbers below 10). All dates are between 01.01.1900 and 31.12.2100.

### Output

Print the date of retirement for the given citizen in the same format as input date.

### Examples

stdin	stdout
3 6 20 01.01.1952	01.04.2012
3 5 1 31.01.1952	01.05.2012

## Problem B. Bridges

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

In the Lemuria Country there are  $N$  islands connected by bridges. Between each two islands with numbers  $i$  and  $i + 1$  ( $1 \leq i \leq N - 1$ ) there are several parallel bridges. Among these bridges there are  $a_i$  stable, and the remaining  $b_i$  bridges are unstable. If one passes a stable bridge from island  $i$ , he reaches the island  $i + 1$ . And if he passes an unstable bridge, then just before he reaches an opposite end of the bridge, it breaks (and since it is broken now, he will never consider it in the future), and the traveler falls in the stream, where the flow brings him to the island number 1.

We are staying in the island number 1. Our goal is to reach the island number  $N$ . We will use the following strategy: on each step we randomly select one of the bridges from the current island to the next one which is not yet broken and pass it. It is clear that if  $a_i > 0$  for all  $i$ , then we will achieve our goal sometime.

The length of each bridge is 1. Find the average distance we will travel by bridges before we reach the island with number  $N$  if we follow the algorithm described above.

### Input

The first line of the input contains an integer  $N$  ( $1 \leq N \leq 1000$ ). The next  $N - 1$  lines contain the description of the bridges. The  $i$ 'th line contains two integers  $a_i$  and  $b_i$  separated by single space, ( $1 \leq a_i \leq 1000$ ,  $0 \leq b_i \leq 1000$ ). The total number of unstable bridges is less than or equal to 1000.

### Output

Output should contain one real number — the answer to the problem. Absolute or relative error should be less than  $10^{-9}$ .

### Examples

<code>stdin</code>	<code>stdout</code>
2 1 1	1.5000000000
3 1 2 2 1	3.8333333333

## Problem C. Crisis

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Finance minister of Lemuria reported to the King that the economic crisis is getting worse. In the first month of the crisis, the budget income was  $P_1$ , and in the second month, it was  $P_2$ . The minister's analysis showed that the excessive spendings on the military will cause the budget income to change by the following formula:

$$P_i = P_{i-2} - P_{i-1}$$

The minister said that sooner or later, the current policy would cause a budget deficit (that is, the income in some month will become negative). After that, he was retired by the King.

The ex-minister then turned to the experts. They, in turn, hired you to check whether his forecast was right. Assuming that the formula above is right, find out whether the income in some month will become negative, and if so, find the number of the first such month and the budget income in that month.

### Input

The first and only line of the input file contains integers  $P_1$  and  $P_2$  which are the budget income in the first and second months of the crisis, respectively ( $0 \leq P_1, P_2 \leq 2 \times 10^9$ ).

### Output

On the first line of the output file, write two numbers separated by a space. If the income in some month will become negative, the first number should be the number of the first such month, and the second number should be the budget income in that month. Otherwise, both numbers should be zeroes. Months are numbered starting from 1.

### Examples

<code>stdin</code>	<code>stdout</code>
34 21	12 -1
2310 2011	5 -1413

## Problem D. Dining Room

Input file: `stdin`  
Output file: `stdout`  
Time limit: 3 seconds  
Memory limit: 256 mebibytes

One famous company has employees of two types: extroverts and introverts. These two kinds of people differ in many ways, and in this problem we consider the difference in their behavior during the dinner.

The dining room in this company can be represented as a rectangular grid  $N \times M$ . In other words, each point with integer coordinates  $x, y$  ( $0 \leq x \leq N$ ,  $0 \leq y \leq M$ ) contains a table. Each table is for one person.

We say that the (Euclidean) distance between points with coordinates  $(x_1, y_1)$  and  $(x_2, y_2)$  is equal to  $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ .

When an introvert comes to the dining room, he selects such a free table that the minimal distance from it to the nearest occupied table is as large as possible. If there are several tables with the largest minimal distance, he selects the one of them with the smallest  $x$  coordinate. If there are still several tables, he selects the one of them with the minimal  $y$  coordinate.

When an extrovert comes to the dining room, he selects such a free table that the maximal distance from it to the farthest occupied table is as small as possible. If there are several tables with the smallest maximal distance, he selects the one of them with the smallest  $x$  coordinate. If there are still several tables, he selects the one of them with the minimal  $y$  coordinate.

You are given the size of the dining room and the description of  $Q$  events. Each event is either an arrival or a departure of one person. For each arrival you have to find the table that he occupies.

### Input

The first line on the input contains integers  $N, M, Q$ , separated by spaces ( $1 \leq N, M \leq 5000$ ,  $0 \leq Q \leq 100$ ). Next  $Q$  lines describe events. If the  $i$ 'th event is an arrival of an extrovert, the  $i$ 'th line contains single string "`ext`"; if the event is an arrival of an introvert, the line contains single string "`int`"; if the event is a departure of some person, the line contains the 1-based index (among all events) of the event that describes his arrival.

It is guaranteed that the input data is correct, i.e. for each arriving human there is at least one empty table in the dining room at the moment of his arrival, and for each event of the departure the index described in the line corresponds to an event of arrival and the person arrived in this event is still inside the dining room.

### Output

For each arrival print a line containing two space-separated integers — the coordinates of the table being occupied.

## Examples

stdin	stdout
5 5 5 int 1 ext 3 int	0 0 0 0 0 0
4 4 8 ext ext int int int int 2 int int	0 0 0 1 4 4 4 0 0 4 2 2 0 2

## Problem E. Exhibition Hall

Input file: `stdin`  
Output file: `stdout`  
Time limit: 6 seconds  
Memory limit: 256 mebibytes

King of Lemuria Julian XCIX is very proud of his collection of paintings by famous abstract artists. These paintings covered every inch of one wall in his former palace. When Julian moved to the new palace, he ordered to move this collection too. Paintings were removed from the wall, carefully packed and transported to the new palace where the wall of exactly the same size was prepared for them.

And now there is a problem — to hang paintings on the wall. The wall is rectangular, with height  $H$  inches and width  $W$  inches. Every painting is rectangular too. Paintings should cover the whole wall and should not overlap.

Your task is to find locations for all paintings. Nobody knows what side of painting is top or bottom, left or right, so you should not care about orientation. Just print coordinates of left bottom and right top corners of every painting.

### Input

First line of the input contains one integer — number of paintings  $N$  ( $1 \leq N \leq 11$ ).

In the second line there are two integers  $W$  and  $H$  — dimensions of the wall ( $1 \leq W, H \leq 10^6$ ).

$N$  following lines contain pairs of integers  $a_i$  and  $b_i$  that give size of each painting ( $1 \leq a_i, b_i \leq 10^6$ ).

### Output

Print  $N$  lines. Each line should contain four space-separated integers: the coordinates of bottom left corner  $x_1, y_1$  and the coordinates of top right corner  $x_2, y_2$  that give the location of some painting ( $0 \leq x_1 < x_2 \leq W, 0 \leq y_1 < y_2 \leq H$ ). The order of lines may be arbitrary.

It is guaranteed that at least one solution exists.

### Examples

stdin	stdout
3	0 0 2 5
5 8	2 0 5 5
2 5	0 5 5 8
3 5	
3 5	
5	0 0 1 7
10 10	1 0 10 2
1 7	1 2 6 7
4 8	6 2 10 10
3 6	0 7 6 10
2 9	
5 5	

## Problem F. File Sharing

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Boondex company has issued a critical security patch with the size of  $K$  data packets for their BromOS network operating system. BromOS infrastructure uses a distributed peer-to-peer networking solution for the delivery of OS patches. There are  $N$  nodes. Each BromOS node has a direct network connection to a BromOS repository as well as independent connections to each of the other active BromOS nodes. Each connection is capable of delivering an entire packet of data in each direction per second. Each node can use all of its connections simultaneously to send and receive data.

Security patch is fully deployed on central repository at time 0. The  $N$  BromOS nodes become active at given times  $T_1, T_2, \dots, T_N$ . If a node receives some packet, it permanently stores it and can transmit it to other nodes later (at any second it can transmit different packets to different nodes). Please help Boondex find the minimal possible amount of time when all nodes can have a full security patch.

### Input

In the first line of input integers  $N$  and  $K$  are given — the total number of BromOS nodes ( $1 \leq N \leq 10^5$ ) and the total number of data packets in a critical security patch ( $1 \leq K \leq 10^6$ ) respectively.

On the next  $N$  lines of input integers  $T_i$  ( $0 \leq T_i \leq 10^6$ ) are given — the time in seconds from time 0 when  $i$ -th BromOS node becomes active and able to communicate with others.

### Output

Output the minimal possible amount of time (in seconds) from time 0 when all nodes can have a full security patch.

### Examples

<code>stdin</code>	<code>stdout</code>
2 3 0 0	2
2 2 0 1	2

## Problem G. tourist

Input file: `stdin`  
Output file: `stdout`  
Time limit: 3 seconds  
Memory limit: 256 mebibytes

The road system of Lemuria is quite complicated. Some of the cities are connected by some kind of one way or two way roads. However, lemurs keep their roads in the proper order and their road maps are always up to date.

G. the tourist just arrived by Penguin Air flight to Lemurian city  $S$ . His return flight departs from city  $F$ . G. has a list of the most popular attractions of Lemuria and he is eager to visit all of them. Also he has a latest version of the road map, so for every city  $P$  he has a list of cities that can be reached by a direct road from  $P$ . Your task is to help the tourist to plan the route across Lemuria that starts in  $S$ , goes through all cities with attractions (in arbitrary order) and ends in  $F$ .

### Input

The first line of the input file contains four integers: total number of Lemurian cities  $N$  ( $1 \leq N \leq 2000$ ), number of cities with attractions  $K$  ( $0 \leq K \leq N$ ), and ordinal numbers of cities  $S$  and  $F$  (all cities are enumerated from 1). Second line contains  $K$  distinct numbers — cities with attractions.

Last  $N$  lines contain the road map. Each of these lines contains  $N$  characters: if  $q^{th}$  character of the  $(p+2)^{nd}$  line of input is 1 then there is a direct way from city  $p$  to city  $q$ , if it is 0 then there is no direct way. There is no road from a city to itself.

### Output

If there is no route that satisfies the conditions, output should contain one line with the word “NO”. Otherwise the first line of output should contain the word “YES”, the second line should contain the number of cities in the route  $M$  ( $M \leq N^2$ ), and in the following line(s)  $M$  numbers should follow — the numbers of cities in the route. The first city in the route should be  $S$ , the last city should be  $F$ , and for any two adjacent cities in the route, there should be a direct road from the first city to the second one.

### Examples

stdin	stdout
4 4 1 2 1 2 3 4 0100 0011 1000 1000	YES 8 1 2 4 1 2 3 1 2
4 2 1 4 2 3 0110 0001 0001 0000	NO
4 1 2 3 4 0010 1001 1000 0100	YES 7 2 4 2 1 3 1 3

## Problem H. Heroes of Money and Magic

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Boondex company has recently got through an IPO procedure and now trades its shares on a stock exchange. Now it has a history of its share prices for some period of time. For the new marketing event company wants to calculate the maximum amount of money one could have earned if he speculated on Boondex shares in that period of time.

Boondex shares are non-divisible for the purpose of buy/sell operations. They are also so highly liquid that any operation on stock exchange with Boondex shares can be executed in any amounts at any time.

Stock exchange has some fee for each operation that is executed by the investor. The fee is flat rate and fixed. It is collected before each transaction for "buy shares" operations and after each transaction for "sell shares" operation. If fee collection is not possible due to insufficient amount of investor's money, such operations are not permitted.

Investor must sell all his Boondex shares by the end of the last day.

### Input

In the first line of input integers  $N$ ,  $M$  and  $F$  are given — the number of days with price information ( $1 \leq N \leq 10^5$ ), the amount of money investor has initially ( $1 \leq M \leq 10^5$ ) and a fee  $F$  ( $1 \leq F \leq 10^5$ ) for each exchange operation. On the next  $N$  lines of input integers  $P_i$  ( $1 \leq P_i \leq 10^5$ ) are given — the price of one Boondex share on  $i$ -th consecutive day. It is guaranteed that the answer for this problem is less than  $10^{18}$ .

### Output

Print on the first line the maximum amount of money investor could possibly have after  $N$  days. Next  $N$  lines should indicate investor's actions for each day on stock exchange to achieve this result: output positive number of shares he should buy on that day, negative number of shares he should sell on that day or zero if he should abstain of any operations on that day.

### Examples

stdin	stdout
3 10000 1 4000 4004 4002	10006 2 -2 0
3 10000 1 4001 4000 4004	10006 0 2 -2

## Problem I. I, V, X, L, C, D, M Problem

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

You are given a string consisting of characters with ASCII codes from 34 to 126, inclusive. Capital and lowercase letters `I`, `i`, `V`, `v`, `X`, `x`, `L`, `l`, `C`, `c`, `D`, `d`, `M`, `m` in this string are considered as Roman digits; lowercase and uppercase letters are considered the same. Your task is to make up the longest possible number in the Roman numeral system from these characters (given letters can be rearranged arbitrarily). If several such numbers exist, choose the largest one.

The rules in the Roman numeral system are the following. Mentioned letters `I`, `V`, `X`, `L`, `C`, `D` and `M` (Roman digits) correspond to numbers 1, 5, 10, 50, 100, 500 and 1000 respectively. A number in the Roman numeral system cannot have more than three identical letters in a row. Each of the characters `V`, `L`, `D` can appear at most once. Letters `I`, `X` and `C` can be put before letters `X`, `C` and `M` to make up 9, 90 and 900, or before letters `V`, `L` and `D` to make up 4, 40 and 400 respectively.

`IX` and `IV` cannot be followed by anything else, `XL` and `XC` can be followed only by `V` or `I`, `CD` and `CM` — only by `L`, `X`, `V` or `I`. For example, `IVII` is not a correct Roman numeral.

In other cases, one digit put after another means just the sum of the digits.

Lastly, a smaller digit must be put after a bigger or identical one (here combinations `IV`, `XC`, ... are considered like digits). So representation of a number in the Roman numeral system is unique.

### Input

On the first line of input file, there is a non-empty string consisting of characters with ASCII codes from 34 to 126, inclusive. The string length does not exceed 100 000 characters.

### Output

Please output the largest of the longest possible numbers in the Roman numeral system which can be formed from the given string. Denote all Roman numerals by capital letters.

### Examples

<code>stdin</code>	<code>stdout</code>
<code>Xiva</code>	<code>XVI</code>
<code>zEr0</code>	
<code>ixvIii</code>	<code>XVIII</code>

## Problem J. Jams

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Boondex has decided to improve its traffic jams coloring algorithm to be more consistent with drivers' expectations. For this purpose Boondex has collected drivers' feedback as a set of  $N$  integer pairs  $V_i, C_i$  where  $V_i$  is the speed of the driver's vehicle and  $C_i$  ( $C_i \in \{0, 1, 2\}$ ) is the color that is expected to be seen on the map for this speed by this driver.

Please help Boondex find two integers  $A$  and  $B$  ( $0 \leq A \leq B$ ) that will be used for their new traffic jams coloring algorithm. Traffic color will be considered to be color 0 if  $0 \leq V \leq A$ , color 1 if  $(A + 1) \leq V \leq B$  and color 2 if  $(B + 1) \leq V$ . Values  $A$  and  $B$  should be chosen to minimize the number of cases where traffic color from the new coloring algorithm is different from the driver's feedback. Among the possible combinations of  $A$  and  $B$  minimizing the number of such cases, the one with minimal value of  $(A + B)$  should be chosen as an answer.

### Input

In the first line of input integer  $N$  is given — the total number of feedback pairs ( $1 \leq N \leq 10^5$ ). On the next  $N$  lines of input integers  $V_i$  ( $0 \leq V_i \leq 10^6$ ) and  $C_i$  ( $C_i \in \{0, 1, 2\}$ ) are given — the driver's speed and expected traffic color for that speed respectively.

### Output

Print two integers  $A$  and  $B$  — the answer for this problem.

### Examples

stdin	stdout
3 5 0 20 1 40 2	5 20
3 10 2 20 1 30 0	0 0

## Problem K. King's Palace Garden

Input file: `stdin`  
Output file: `stdout`  
Time limit: 3 seconds  
Memory limit: 256 mebibytes

King of Lemuria Julian XCIX has built a new palace for himself in the middle of Nowhere. Nowhere is a large plain savanna with several baobab trees. Now Julian wants to design the Palace Garden around his palace. Lemuria customs dictate that the Garden should have a shape of a convex polygon with corners in centers of baobab trees and without any baobabs inside the Garden. But the Palace should be strictly inside it.

Your task is to help Julian to select the Garden of the maximal area. If there is more than one garden of the same maximal area possible, find any of them.

### Input

In the first line of input there is one integer  $N$  — number of baobabs,  $0 \leq N \leq 500$ . Second line contains two integers — coordinates of the palace. Each one of the following  $N$  lines also contains two integers — coordinates of baobab tree. All coordinates (of the Palace and of baobabs) satisfy the condition  $-10^9 \leq x, y \leq 10^9$ . All  $N + 1$  points are pairwise distinct and no three of them lie on the same line.

### Output

The first line of the output should contain one number with exactly one digit after the decimal point — area of the garden.

The second line should contain one integer — number of corners of the garden.

The third line should contain the numbers of baobabs which stand in the corners of the garden in counterclockwise order. Baobabs in the input data are numbered from 1.

If there is no garden satisfying the conditions, the output should contain one number  $-1$ .

### Examples

<code>stdin</code>	<code>stdout</code>
4 5 5 1 1 1 9 7 5 10 6	24.0 3 1 3 2
3 0 0 1 1 0 1 1 0	-1

## Problem L. Lamps of the Mind

Input file: `stdin`  
Output file: `stdout`  
Time limit: 3 seconds  
Memory limit: 256 mebibytes

The great mathematician John spent hours staring at his exciting abat-jour. Lamps in it formed a regular polygon. Some of them, with the red light, interested him a lot, other with the blue light less, and some of them didn't interest him at all. His mind chased them, made triangles out of them, memorized and compared those triangles, and after a while triangles forced out of his mind all the rest.

You were hired by a researcher to help unlift the veil over John's genius. It is known that John preferred some colors over others, so there were restrictions on the number of lamps of each color that can be in the considered triangles. Moreover only the geometric form (triangles with the same set length of sides have the same form), but not the position or color of the lamps, was memorized by John. As the first task you have to calculate the number of triangles in the mind of the great mathematician.

### Input

The first line of the input contains two integers  $N$  and  $K$  ( $3 \leq N \leq 3000, 1 \leq K \leq N$ ), separated by single space, the number of lamps forming the regular polygon and the number of their colors respectively. The second line contains  $N$  integers  $c_i$  ( $1 \leq c_i \leq K$ ) separated by single space — colors of the lamps. The third line contains  $K$  integers  $l_i$  ( $0 \leq l_i \leq 3$ ). They mean that the triangles that have more than  $l_i$  lamps of  $i$ -th color are not considered by the great mathematician.

### Output

Output should contain one integer — the number of geometrically different triangles in the mind of the great mathematician.

### Examples

<code>stdin</code>	<code>stdout</code>
5 3 1 2 2 1 3 2 1 0	1
4 2 1 2 1 2 1 1	0
6 3 1 2 3 1 2 3 2 2 2	3