Problem A. Gift

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

The kingdom of Olympia consists of N cities and M bidirectional roads. Each road connects exactly two cities and two cities can be connected with more than one road. Also it possible that some roads connect city with itself making a loop.

All roads are constantly plundered with bandits. After a while bandits became bored of wasting time in road robberies, so they suggested the king of Olympia to pay off. According to the offer, bandits want to get a gift consisted of gold and silver coins. Offer also contains a list of restrictions: for each road it is known g_i — the smallest amount of gold and s_i — the smallest amount of silver coins that should be in the gift to stop robberies on the road. That is, if the gift contains a gold and b silver coins, then bandits will stop robberies on all the roads that $g_i \le a$ and $s_i \le b$.

Unfortunately kingdom treasury doesn't contain neither gold nor silver coins, but there are Olympian tugriks in it. The cost of one gold coin in tugriks is G, and the cost of one silver coin in tugriks is S. King really wants to send bandits such gift that for any two cities there will exist a safe path between them. Your task is to find the minimal cost in Olympian tugriks of the required gift.

Input

The first line of the input contains two integers N and M ($2 \le N \le 200$, $1 \le M \le 50\,000$) — the number of cities and the number of roads, respectively. The second line contains two integers S and G ($1 \le G, S \le 10^9$) — the prices of gold and silver coins in tugriks. The following M lines contain information about the offer. Each of the records in list is given as four integers x_i, y_i, g_i, s_i, where x_i and y_i are the numbers of cities that the road connects and g_i, s_i are minimal gold and silver coins requirements for the i-th road ($1 \le x_i, y_i \le n$, $1 \le g_i, s_i \le 10^9$). Cities are numbered from 1 to N. It is possible that there are more than one road between a pair of cities. It is possible that a road connects the city with itself.

Output

The output should contain the minimal cost of the gift in Olympian tugriks. If there is no gift that satisfies the given requirements output `-1`.

Examples

stdin	stdout
3 3 2 1 1 2 10 15 1 2 4 20 1 3 5 1	30

Problem B. Mice

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

Modern researches has shown that a flock of hungry mice searching for a piece of cheese acts as follows: if there are several pieces of cheese then each mouse chooses the closest one. After that all mice start moving towards the chosen piece of cheese. When a mouse or several mice achieve the destination point and there is still a piece of cheese in it, they eat it and become well-fed. Each mice that reaches this point after that remains hungry. Moving speeds of all mice are equal.

If there are several ways to choose closest pieces then mice will choose it in a way that would minimize the number of hungry mice. To check this theory scientists decided to conduct an experiment. They located N mice and M pieces of cheese on a cartesian plane where all mice are located on the line $y = Y_0$ and all pieces of cheese — on another line $y = Y_1$. To check the results of the experiment the scientists need a program which simulates the behavior of a flock of hungry mice.

Write a program that computes the minimal number of mice which will remain hungry, i.e. without cheese.

Input

The first line of the input contains four integer numbers N ($1 \leq N \leq 10^5$), M ($0 \leq M \leq 10^5$), Y_0 ($0 \leq Y_0 \leq 10^7$), Y_1 ($0 \leq Y_1 \leq 10^7$, $Y_0 \neq Y_1$). The second line contains a strictly increasing sequence of N numbers — x coordinates of mice. Third line contains a strictly increasing sequence of M numbers — x coordinates of cheese. All coordinates are integers and do not exceed 10^7 by absolute value.

Output

The only line of output should contain one number — the minimal number of mice which will remain witouht cheese.

Examples

stdin	stdout
3 2 0 2 0 1 3 2 5	1

Note

All the three mice will choose the first piece of cheese. Second and third mice will eat this piece. The first one will remain hungry, because it was running towards the same piece, but it was late. The second piece of cheese will remain uneaten.

Problem C. Mutation

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

Scientists of planet Olympia are conducting an experiment in mutation of primitive organisms. Genome of organism from this planet can be represented as a string of the first K capital English letters. For each pair of types of genes they assigned $a_{i,j}$ — a risk of disease occurence in the organism provided that genes of these types are adjacent in the genome, where i — the 1-based index of the first gene and j — the index of the second gene. The gene 'A' has index 1, 'B' has index 2 and so on. For example, $a_{3,2}$ stands for the risk of 'CB' fragment. Risk of disease occurence in the organism is equal to the sum of risks for each pair of adjacent genes in the genome.

Scientists have already obtained a base organism. Some new organisms can be obtained by mutation of this organism. Mutation involves removal of all genes of some particular types. Such removal increases the total risk of disease occurence additionally. For each type of genes scientists determined t_i — the increasement of the total risk of disease occurence provided by removal of all genes having type with index i. For example, t_4 stands for the value of additional total risk increasement in case of removing all the 'D' genes.

Scientists want to find a number of different organisms that can be obtained from the given one which have the total risk of disease occurence not greater than T. They can use only the process of mutation described above. Two organisms are considered different if strings representing their genomes are different. Genome should contain at least one gene.

Input

The first line of the input contains three integer numbers N ($1 \le N \le 200\,000$) — length of the genome of base organism, K ($1 \le K \le 22$) — the maximal index of gene type in the genome and T ($1 \le T \le 2 \cdot 10^9$) — maximal allowable risk of disease occurence. The second line contains the genome of the given organism. It is a string of the first K capital English letters having length N.

The third line contains K numbers $t_1, t_2, \ldots, t_K$, where t_i is additional risk value of disease occurence provided by removing of all genes of the i-th type.

The following K lines contain the elements of the given matrix $a_{i,j}$. The i-th line contains K numbers. The j-th number of the i-th line stands for a risk of disease occurence for the pair of genes, first of which corresponds to the i-th letter and second of which corresponds to the j-th letter. The given matrix is **not** necessarily symmetrical.

All the numbers in the input are integer, non-negative and all of them except T are not greater than 10^9. It is guaranteed that the maximal possible risk of organism that can be obtained from the given organism is strictly smaller than 2^{31}.

Output

Output the number of organisms that can be obtained from the base one and which have the total risk of disease occurence not greater than T.

Examples

stdin	stdout
5 3 13 BACAC 4 1 2 1 2 3 2 3 4 3 4 10	5

Note

Explanation: one can obtain the following organisms (risks are stated in brackets): BACAC (11), ACAC (10), BAA (5), B (6), AA (4).

Problem D. Plus and xor

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

Bitwise exclusive OR (or bitwise addition modulo two) is a binary operation which is equivalent to applying logical exclusive OR to every pair of bits located on the same positions in binary notation of operands. In other words, a binary digit of the result is equal to 1 if and only if bits on the respective positions in the operands are different.

For example, if $X = 109_{10} = 1101101_2$, $Y = 41_{10} = 101001_2$, then:

$$X \ \texttt{xor} \ Y \ = \ 68_{10} \ = \ 1000100_2.$$

Write a program, which takes two non-negative integers A and B as an input and finds two non-negative integers X and Y, which satisfy the following conditions:

- $A = X + Y$
- $B \ = \ X \ \texttt{xor} \ Y$, where `xor` is bitwise exclusive or.
- X is the smallest number among all numbers for which the first two conditions are true.

Input

The first line contains integer number A and the second line contains integer number B $(0 \le A, B \le 2^{64} - 1)$.

Output

The only output line should contain two integer non-negative numbers X and Y. Print the only number `-1` if there is no answer.

Examples

stdin	stdout
142 76	33 109

Problem E. Points

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

You are given N points on a plane. Write a program which will find the sum of squares of distances between all pairs of points.

Input

The first line of input contains one integer number N ($1 \le N \le 100\,000$) — the number of points. Each of the following N lines contain two integer numbers X and Y ($-10\,000 \le X, Y \le 10\,000$) — the coordinates of points. Two or more points may coincide.

Output

The only line of output should contain the required sum of squares of distances between all pairs of points.

Examples

stdin	stdout
4 1 1 -1 -1 1 -1 -1 1	32

Problem F. Tourist

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

Tourist walks along the X axis. He can choose either of two directions and any speed not exceeding V. He can also stand without moving anywhere. He knows from newspapers that at time t_1 in the point with coordinate x_1 an interesting event will occur, at time t_2 in the point with coordinate x_2 — another one, and so on up to (x_n, t_n). Interesting events are short so we can assume they are immediate. Event i counts visited if at time t_i tourist was at point with coordinate x_i.

Write program tourist that will find maximum number of events tourist if:

- at the beginning (when time is equal to 0) tourist appears at point 0,
- tourist can choose initial point for himself.

Yes, you should answer on two similar but different questions.

Input

The first line of input contains single integer number N ($1 \le N \le 100000$) — number of interesting events. The following N lines contain two integers x_i and t_i — coordinate and time of the i-th event. The last line of the input contains integer V — maximum speed of the tourist. All x_i will be within range $-2 \cdot 10^8 \le x_i \le 2 \cdot 10^8$, all t_i will be between 1 and $2 \cdot 10^6$ inclusive. V will be positive and will not exceed 1000. The input may contain events that happen at the same time or in the same place but not in the same place at the same time.

Output

The only line of the output should contain two space-sepatated integers — maximum number of events tourist can visit in he starts moving from point 0 at time 0, and maximum number of events tourist can visit if he chooses the initial point for himself.

Examples

stdin	stdout
3 -1 1 42 7 40 8 2	1 2