Problem A. Armour

Input file:	`standard input`
Output file:	`standard output`
Time limit (C++):	1 second
Time limit (Java):	2 seconds
Time limit (Python):	10 seconds
Memory limit:	64 megabytes
Java Class Name:	`armour`
Maximum Points Available:	100

Kate, the blacksmith, is showing William, the Knight, a range of armour in the hope that he will buy a particularly expensive piece.

William, on the other hand, has a very short attention span, and is often distracted by the passing of a lady of the court.

Kate has arranged her N wares in a line, and she and William will walk along the line, inspecting the pieces one by one, in order. Kate thinks she knows exactly how long it will be before William will become distracted and wants to show him the best pieces she can before then, by starting at an appropriate point on the line.

She has a mental catalogue of all the pieces and has assigned them each a value rating. She always wants to show William the best pieces she can. She does this by choosing a starting point in the line such that the sum of the values of the pieces that William sees before he is distracted is maximized.

Given possibilities for William's attention span, help Kate work out where to guide William.

Input

The first line will contain two space separated integers, N and T.

The next line contains N space separated integers, V_i, where the ith integer is the value of the ith item in line. All values are between 1 and 1 000 000 inclusive.

The following T lines will each contain a single integer, D_i, a possible number of items that William will browse before being distracted.

Output

Output T lines, one per each D_i. Each line should contain two space separated integers. The first is the index of the item where Kate should guide William, and the second is the sum of the values of the items he will see before getting distracted, given that he will look at D_i items.

In the case that multiple starting points will result in the same value of items seen, output the smallest starting point.

Scoring

In cases worth a minimum of 12 points, $1 \leq N \leq 100$ and $T \leq 10$.

In cases worth a minimum of 34 points, $1 \leq N \leq 100$ and $T \leq 1000$.

In cases worth a minimum of 54 points, $1 \leq N \leq 10\,000$ and $T \leq 1000$.

Example

standard input	standard output

5 2
1 3 3 1 4
1
2
	5 4
	2 6

Note

In the given example, if William will only look at one piece, he should look at the fifth one (the last one) for a value of 5. If he will look at two pieces, then Kate should direct him to the two with value 3 each, so he starts on the second piece, and sees a total value of 6.

Problem B. Square

Input file:	`input_xx.txt`
Output file:	`output_xx.txt`
Maximum Points Available:	100

Bernard the monk has been found out!

As punishment for his heinous transcription antics, the brothers have thrown Bernard into prison. Left alone with his thoughts, he has taken to examining the various graffiti on the walls. He found an interesting piece, which looks as follows:

O	F	P
I	E	F
O	R	R

Bernard has no idea what it means, but, to pass the time, has started thinking of every word he knows that consists only of those letters, and definitely includes the middle one.

Help Bernard find all the words he can. You will be given a dictionary to help you.

Input

The input will consist of nine letters, arranged in a 3 by 3 square.

Output

Output a list of words, one per line. Each word must have at least four letters and include the middle letter. Each letter in the square can be used at most once in each word. Each word must appear in the dictionary.

Scoring

Scoring for this task is in two stages.

For every test case, you will receive 1 point for every valid four letter word in the list, 2 points for every valid 5 letter word, 3 points for 6 letter words and so on. For every word in the list that is invalid (as per the "output" section above), three points will be deducted.

Thereafter your score will be scaled so that the maximum possible attainable score for the question is 100 points and each input file is worth the same number of points.

If you get a negative score, it will be capped at 0.

Example

input_xx.txt	output_xx.txt

`O F P`
`I E F`
`O R R`
	`PORE`
	`RIPER`
	`OFFER`
	`PROFFER`
	`FIREPROOF`
	`PIER`

Note

You will be given a dictionary of around 70 000 words in a file named `dict`.

This is an output only problem. That is, you will not submit any code, only answers. The input files are available on your computer, or on the submission system.

Problem C. Marbles

Input file:	`standard input`
Output file:	`standard output`
Time limit (C++):	1 second
Time limit (Java):	2 seconds
Time limit (Python):	10 seconds
Memory limit:	256 megabytes
Java Class Name:	`marbles`
Maximum Points Available:	100

Many have heard of the legendary hero of the Geats, Beowulf, who slayed the monster Grendel. However, few people know that Beowulf had an enthusiastic hobby of collecting marbles. Even fewer know that the real reason Beowulf slayed Grendel was not because of king Hrothgar's request, but really because Grendel had played a trick on Beowulf by tangling all of his marbles together with string.

Beowulf has N marbles, which are numbered from 1 to N. Grendel has attached a string between each of M pairs of Beowulf's marbles. Beowulf would like to remove the strings as quickly as possible. He does this by making a sequence of T slashes with his mighty sword. For each slash, he separates the marbles into two groups, placing some on his left hand side and the remaining marbles on his right hand side. He then slashes some of the strings that pass between these two groups which have not already been slashed. He continues to do this until all the strings have been slashed.

Help Beowulf choose how to make each cut so that he does not have to spend too long slashing.

Input

The first line will contain three space separated integers, N, M and K. Here, K is the subtask number (see "Scoring" below).

The next M lines will each contain two space separated integers, a and b, meaning that there is a string connecting marble a to marble b.

Output

The first line of output must contain a single integer T, the number of slashes made.

Subsequently, for each of the T slashes, in the order in which they are made, output:

One line containing a single integer C, the number of strings cut in this step, followed by C lines which should contain pairs of space separated integers a b, meaning that the string between marble a and b is cut in this step, and that marble a is on the left hand side of the division and marble b is on the right hand side of the division.

Scoring

This task is divided into "subtasks", each worth a different number of points.

Subtask 1

(10 points): In this subtask, $2 \leq N \leq 5$ and $1 \leq M \leq 10$.

The strings must be cut with 4 slashes or fewer.

Subtask 2

(10 points):

In this subtask, $2 \leq N \leq 1000$ and $1 \leq M \leq 1000$.

The strings must be cut with 1000 slashes or fewer.

Subtask 3

(30 points):

In this subtask, $2 \leq N \leq 1000$ and $1 \leq M \leq 1000$.

The strings must be cut with 10 slashes or fewer.

Subtask 4

(50 points):

In this subtask, $2 \leq N \leq 100000$ and $1 \leq M \leq 80000$.

Each test case will be scored as follows: if 9 or less slashes are made, 50 points will be awarded. Otherwise you will score $\lfloor \frac{50}{(T-8)^{0.7}} \rfloor$ points for the test case. The number of points scored for the subtask will be the minimum of the points scored on each test case.

Example

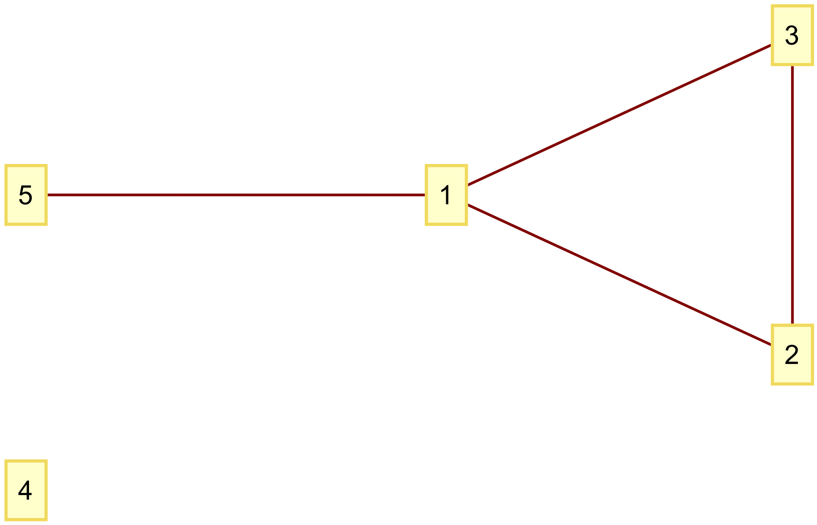
standard input	standard output

5 4 1
1 2
2 3
3 1
1 5
	2
	3
	5 1
	3 1
	3 2
	1
	2 1

Note

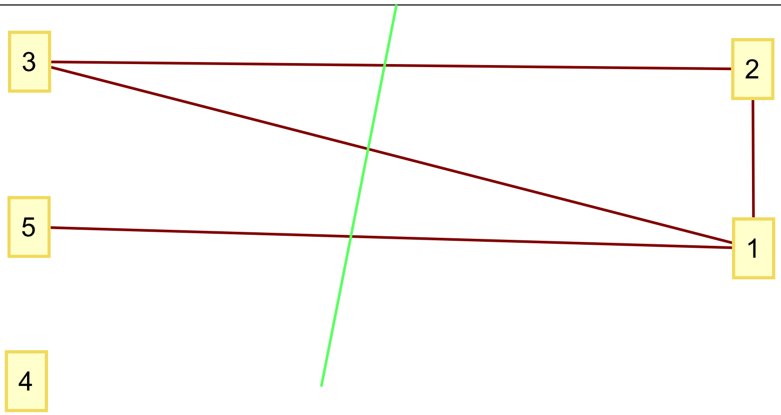
In the given example, there are 5 marbles with 4 strings between them. This task falls under subtask 1, which means that we must give an answer which uses 4 slashes or fewer.

The strings attach marble 1 and marble 2, marble 2 and marble 3, marble 3 and marble 1, and marble 1 and marble 5. Graphically, the scenario looks like this:

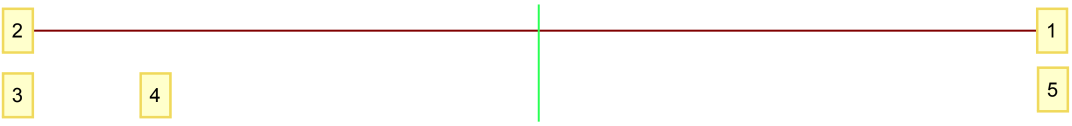


We can cut all the strings with 2 slashes as follows.

In our first slash, we can put marbles 3,4,5 on the left hand side, and marbles 1,2 on the right hand side. If we cut all the strings in the middle of these two sides, we cut the strings connecting marble 5 and marble 1, marble 3 and marble 1, and marble 3 and marble 2.



In our second slash, we can put marbles 2,3,4 on the left hand side and marbles 1,5 on the right hand side. If we cut all the strings in the middle, we cut the string connecting marble 1 to marble 2. After this, there are no more strings left to cut.



Problem D. Paths

Input file:	`standard input`
Output file:	`standard output`
Time limit (C++):	3 seconds
Time limit (Java):	6 seconds
Time limit (Python):	30 seconds
Memory limit:	256 megabytes
Java Class Name:	`paths`
Maximum Points Available:	100

Bard Geoffrey's uncle is having an extravagant feast to celebrate the victory of a recent battle. In celebration, he even invited his unliked nephew to the festivities.

Geoffrey wants to be there for the event, but cannot afford to travel without income. Thus he plans to visit a number the of towns, numbered 1 to N on the way to his uncle to sing for his supper. He can take any of the M roads between towns, which have different lengths.

However, these townspeople are strange. Some towns will turn Geoffrey away, if he has been to other towns recently.

To be precise, there are some triples, C_i, B_i and A_i, where Geoffrey will not be able to visit town C_i if the last town he visited was B_i and the town before that was A_i.

Time is short however, so he must make haste. Help Geoffrey find the shortest path he can take to his uncle's town. He need not visit all the towns, and he can visit a town multiple times.

Input

The first line contains three space separated integers, N, M and K.

The next M lines contain three space separated integers each, X_i, Y_i and W_i denoting that there is a path from town X_i to town Y_i of length W_i ($1 \leq W_i \leq 1000$). There will be at most one path between towns.

The next K lines contain three space separated integers each, A_i, B_i, and C_i, denoting that our bard may not visit the towns A_i, B_i and C_i in that order.

Output

The first line of output should contain length of the shortest path.

The second line of output should contain a sequence of space separated integers, denoting any of the shortest paths (starting with 1, ending with N).

If it is not possible for Geoffrey to get to his uncle's feast, output a single line containing `-1`.

Scoring

In all cases, $1 \leq N \leq 2\,000$, $1 \leq M \leq 50\,000$ and $1 \leq K \leq 100\,000$.

In cases worth a minimum of 10 points, $K = 0$ and $1 \leq N \leq 10$.

In cases worth a minimum of 10 points, $K = 0$ and $1 \leq N \leq 100$ and $W_i = 1$.

In cases worth a minimum of 20 points, $K = 0$.

In cases worth a minimum of 10 points, $A_i = 1$ and $C_i = N$.

In cases worth a minimum of 20 points: $1 \leq N \leq 100$ and $1 \leq M \leq 1\,000$

In cases worth a minimum of 30 points: $1 \leq N \leq 2\,000$ and $1 \leq M \leq 50\,000$

Examples

standard input	standard output
3 1 0 1 2 1	-1

4 4 1
1 2 2
2 3 2
3 4 5
1 3 5
2 3 4
	10
	1 3 4

4 4 2
1 2 2
2 3 2
3 4 5
1 3 5
1 2 3
1 3 4
	14
	1 3 2 3 4