

Problems

Problem	PASTE	WORDS	SEARCH	POLYGON
Executable file	PASTE.EXE	WORDS.EXE	SEARCH.EXE	POLYGON.EXE
Source file	PASTE.BAS	WORDS.BAS	SEARCH.BAS	POLYGON.BAS
	PASTE.PAS	WORDS.PAS	SEARCH.PAS	POLYGON.PAS
	PASTE.C	WORDS.C	SEARCH.C	POLYGON.C
	PASTE.CPP	WORDS.CPP	SEARCH.CPP	POLYGON.CPP
Input file	PASTE.IN	WORDS.IN	SEARCH.IN	POLYGON.IN
Output file	PASTE.OUT	WORDS.OUT	SEARCH.OUT	POLYGON.OUT
Time limit per test	10 seconds	10 seconds	10 seconds	10 seconds
Number of tests	10	10	10	10
Points per test	3	4	6	7
Total points	30	40	60	70
	200			

PASTE

A document processed by a text processor consists of N lines of text. The first line contains number 1, the second line contains number 2 and so on till the N^{th} line which contains number N .

Exactly M operations 'cut and paste' have been performed in that document. It operates on a selected group of consecutive lines; 'cut' removes selected text from the document and 'paste' inserts removed text elsewhere in the rest of the document.

Write a program that will for given sequence of 'cut and paste' operations determine the contents of **the first ten lines** of final document after all operations have been performed.

Input data

The first line of input file contains two natural numbers N , number of lines in a document ($10 \leq N \leq 100,000$) and K , number of operations 'cut and paste' performed on a document ($1 \leq K \leq 1000$), separated by a space character.

Next K lines contain information of 'cut and paste' operations in the order of their execution.

Each line contain three natural numbers A , B and C , $1 \leq A \leq B \leq N$, $0 \leq C \leq N-(B-A+1)$, separated by a space character. Numbers A and B determine first and last line of selected text, and number C determines the line after which the removed text should be inserted. If C equals 0 then removed text should be inserted at the beginning of the document.

Output data

The output file should consist of 10 lines containing the numbers written in the first 10 lines of final document after all operations have been performed.

Examples

PASTE.IN

15 1
1 15 0

PASTE.OUT

1
2
3
4
5
6
7
8
9
10

PASTE.IN

13 3
6 12 1
2 9 0
10 13 8

PASTE.OUT

6
7
8
9
10
11
12
2
3
4

PASTE.IN

1000 6
3 7 4
1 100 57
50 60 200
63 70 500
1 800 4
7 77 98

PASTE.OUT

801
802
803
804
101
102
36
37
38
39

WORDS

Io and Ao are playing a word game. They alternately say words **consisting of vowels only** so that the first letter of every new word is the same as the last letter of the previous word. A game can start with any word.

It is forbidden to say any word twice. Only words from given dictionary can be used in a game.

A complexity of a game is defined as a sum of lengths of all the spoken words during the game.

Write a program that will determine the **maximal possible complexity** of a game that can be played using words from a given dictionary.

Input data

The first line of input file contains a natural number N , $1 \leq N \leq 16$, the numbers of words in a dictionary. Each of next N lines contains one word from a dictionary. A word is a sequence of characters 'A', 'E', 'I', 'O' and 'U'. Length of every word will be 100 or less. All the words will be different.

Output data

The first and only line of output file should contain the maximal possible complexity of the game.

Examples

WORDS.IN

3
AEIOU
UIU
EO

WORDS.OUT

8

WORDS.IN

4
AEEEO
OEOAEEE
AO
O

WORDS.OUT

13

WORDS.IN

5
IOO
IUUO
AI
OIOOI
AOOI

WORDS.OUT

16

SEARCH

Young Ralph ‘borrowed’ a car drove off to a town for a fun. What he did not know was that the car belonged to police and it had a device that was supposed to send information about car’s motion.

The device is rather old and it sends only information about a direction of car’s motion.

Write a program that will help police to find the car using a map of the town, its initial position and a sequence of directions the car drove. The program should determine all possible final positions of the car.

A map of the town is rectangular and characters are used to describe where a car can and where cannot drive. Dots (‘.’) denote places of town where a car can drive, characters ‘**X**’ denote places of town where a car cannot drive. The initial position of car Ralph drove is denoted with character ‘*’. A car can drive through the initial position.

A car can drive in four directions: to the north (up), to the south (down), to the west (left) and to the east (right).

A description of Ralph’s movements with a car is given with a sequence of directions. In every given direction Ralph drove his car through one or more passable places of town.

Input data

The first line of input file contains two natural numbers R and C , $1 \leq R \leq 50$, $1 \leq C \leq 50$, separated by a space character, numbers of rows and columns of town’s map.

Each of next R lines contain a sequence of C characters (‘.’ (a dot), ‘**X**’ ‘*’) describing corresponding part of the map.

The following, $(R+2)^{\text{th}}$ line contains a natural number N , $1 \leq N \leq 1000$, length of a sequence of directions.

Each of the following N lines contains one of the following sequences of characters: **NORTH**, **SOUTH**, **WEST** and **EAST**, describing directions of car’s movements.

No two consecutive directions are the same.

SEARCH

Output data

Output file should contain the map of the town in R lines (as in input file), where character “*” should denote only possible final positions of the car.

Examples

SEARCH.IN

```
3 4
....
*..X
X.X.
2
EAST
NORTH
```

SEARCH.OUT

```
.**.
...X
X.X.
```

SEARCH.IN

```
4 5
.....
.X...
...*X
X.X..
3
NORTH
WEST
SOUTH
```

SEARCH.OUT

```
.....
*X*..
*..X
X.X..
```

SEARCH.IN

```
10 9
.....X
X..XX..X.
.X.XX.X..
...XX....
...XX....
.XXX..XX.
.....X.
..XXX.X..
X.X....X.
*.....X..
4
EAST
NORTH
EAST
SOUTH
```

SEARCH.OUT

```
.....X
X..XX.*X.
.X.XX.X..
...XX....
...XX.***
.XXX..XX*
.....X*
..XXX*X.*
X.X..*.X*
....**X.*
```

POLYGON

There are N points in a plane whose coordinates are natural numbers. **A convex polygon with maximal number of vertices** is a convex polygon whose vertices are some of given points **and the origin** having maximal possible number of vertices. Origin, i.e. point with coordinates $(0,0)$, **must** be one of vertices of a convex polygon with maximal number of vertices.

Write a program that will determine number of vertices in such polygon.

A polygon is convex if every line segment whose endpoints are inside that polygon is also completely inside it.

Consecutive edges of a polygon must not be parallel.

Input data

The first line of input file contains a natural number N , $2 \leq N \leq 100$, a number of given points.

Each of the following N lines contains two natural numbers X and Y , $1 \leq X \leq 100$, $1 \leq Y \leq 100$, separated by a space character, coordinates of one point. All points will be different.

Output data

The first and only line of output file should contain number of vertices of convex polygon with maximal number of vertices.

Note: the result will always be at least 3.

Examples

POLYGON.IN	POLYGON.IN	POLYGON.IN
5	8	10
4 2	10 8	9 6
2 2	3 9	1 7
2 3	2 8	2 2
3 2	2 3	3 9
3 1	9 2	8 7
	9 10	3 2
POLYGON.OUT	10 3	9 4
4	8 10	3 1
	POLYGON.OUT	9 7
	8	6 9
		POLYGON.OUT
		7

Explanation for test data #2 (coordinates of polygon)

2 8
3 9
8 10
9 10
10 8
10 3
9 2
0 0