The input data for each problem had to be read from the text file and the answer had to be written to the fixed file OUTPUT.TXT. The program had to ask the user for the name of the input file. All the programs were tested on IBM PC 486, 25 MHz computers.

THE FILLING OF BARRELS (Latvia)
25 points, run time limit – 15 sec.

There are N ($1 < N \leq 100$) equal empty barrels on a horizontal surface. The volume of each barrel is 100 units. Each two of them are connected with a pipe. Each pipe is connected to the bottoms of the two barrels and it has its own valve which can be only in two positions: "OPEN" or "CLOSED". In the beginning all valves are closed. If a valve is open, then the liquid from one connected barrel can quickly and freely flow to the other so that the amount of the liquid in these barrels becomes equal (the principle of communicating tubes). If a valve is closed, the liquid flow through this pipe is not possible.

Two kinds of operations are allowed:

1) "P" (for pouring) when some amount of the liquid is poured into the prescribed barrel. The description of the operation is: "P n m", where n is the number of a barrel and m is the amount of the liquid (in units) to be poured to this barrel (n and m are integers, ($0 < n \leq N$, $0 < m \leq 1000$),

2) "V" (for a valve) when one prescribed valve is turned to the opposite position (i.e. this valve becomes closed if it has been open and becomes open if it has been closed). The description of the operation is: "V n_1 n_2", where n_1 and n_2 are the numbers of the barrels which are connected with a pipe that has this valve (n_1 and n_2 are integers, $0 < n_1 \leq N$, $0 < n_2 \leq N$, $n_1 \neq n_2$). Two different descriptions "V n_1 n_2" and "V n_2 n_1" refer to the same valve.

Your goal is to execute the given sequence of operations. You must neglect the amount of liquid in pipes. If some pouring operation cannot be executed due to the liquid overflow, the execution of this operation sequence must be stopped after an appropriate output.

Input

The input data are written to the text file and contain the numbers N and K (the number of operations $0 < K \leq 1000$) in the first line and the description of the operation in the remaining K lines (one description per each line).

Output

The output data must be written to the text file and must contain "OK" and the values of a minimum and a maximum amount of the liquid in all barrels at the end if the sequence execution is successfully completed or "OVERFLOW" and the number of the operation during which overflow happens. The values of the liquid amount must be written as real numbers, having two signs after a decimal point.

Example 1

```
INPUT                    OUTPUT
2 6                      OK 75.00 95.00
P 1 63
P 2 37
V 1 2
P 1 50
V 2 1
P 1 20
```

Example 2

```
INPUT                    OUTPUT
3 8                      OVERFLOW 7
P 1 100
V 2 1
P 2 40
V 1 2
V 1 3
V 1 3
P 3 70
V 1 3
```

The input data for each problem had to be read from the text file and the answer had to be written to the fixed file OUTPUT.TXT. The program had to ask the user for the name of the input file. All the programs were tested on IBM PC 486, 25 MHz computers.

A FAST JOURNEY (Lithuania)
   35 points, run time limit – 20 sec.

One-side traffic roads join a number of cities. The cities are numbered from 1 to n, where n is the number of cities (n ≤ 100). On odd days vehicles drive in the directions indicated by the traffic, and on even days – in the opposite directions. The length of the road between two cities is measured by an integer value – the duration of the journey from one city to the other (in hours), and it does not depend on the direction.

Write a program to find the route from the city A to the city B which allows to reach the city B as quickly as possible.

The first day of the journey is odd. No more than twelve hours can be spent in the journey during one day. The night must be spent in a city. The journey may be continued on the following day.

Input

The input data are stored in the text file. There are two numbers in the first line – the codes (numbers) of the cities A and B, followed by the total number of cities and the total number of roads k ($1 \le k \le 1000$) on the second line. There is information about each road in the remaining k lines. Each line contains three numbers – the codes of the two cities the road joins and the duration of the journey between the cities expressed in hours. The direction of the road is from the first to the second city.

Output

The output data Write the results to the text file, giving one line per every intermediate route. There should be four numbers in each line – the code of the departure city, the code of the arrival city, the number of the day and the length of the road.

Example

```
INPUT        OUTPUT
1 3          1 5 1 10
6 7          5 4 1 1
1 2 9        4 3 3 4
1 6 2
1 5 10
5 4 1
4 6 2
4 3 4
2 3 5
```

The input data for each problem had to be read from the text file and the answer had to be written to the fixed file OUTPUT.TXT. The program had to ask the user for the name of the input file. All the programs were tested on IBM PC 486, 25 MHz computers.

A NUMBER GAME (Estonia)
   40 points, run time limit – 90 sec.

Consider the following game. At first we assign to the variable n and the constant m integer values ($0 < n \le 70$, $0 < m \le 20$). The players A and B begin to make moves (A is the first) in turns. A positive integer $k \le \min \{m, n\}$ is selected with each move, which decreases the value of the variable n by k. Yet, it is not allowed to take numbers that have already been taken by one or the other player in an earlier move. The game ends when one of the players can make no moves. The player who makes the last move is the winner.

Two integers n and m are written in the first row of the input text file, separated by a space.

Write a program that says which player has a winning strategy and write the following data to the output file:

   Row 1: who has the winning strategy;
   Next rows: all possible first moves of A in an increasing order followed by the word "winning" or by one of the winning answers of B.

Example 1

```
INPUT          OUTPUT
3 2            B wins
               1 2
               2 1
```

Example 2

```
INPUT          OUTPUT
7 4            A wins
               1 winning
               2 winning
               3 4
               4 3
```