South African Computer Olympiad
Training Camp 1, 2006
Day 2

Overview

Author	**USACO**	**Bruce Merry**	**Carl Hultquist**
Problem	**spin**	**stack**	**flowers**
Source	spin.java spin.c spin.cpp spin.pas	stack.java stack.c stack.cpp stack.pas	flowers.java flowers.c flowers.cpp flowers.pas
Input file	spin.in	stdin	flowers.in
Output file	spin.out	stdout	flowers.out
Time limit	1 second	5 seconds	1 second
Number of tests	10	10	10
Points per test	10	10	10
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

Rotating guards

Author

USACO

Introduction

Having given up on a frontal assault of the Castle Aaah, the Knights of the Round Table are going to try to sneak in. Unfortunately, the castle is extremely well defended. It is surrounded by five rings of guards. Within each ring, all the guards walk anti-clockwise at the same speed. However, in each ring there is at least one gap through which the knights may slip. The guards in different rings move at different speeds, so there may be a point where the gaps line up and the knights can slip straight through all five layers.

Task

Write a program that, given a description of the rings and the gaps in them, will determine when the knights can first slip through the guards. Each ring will be described by the number of degrees per time unit that the guards move around the centre, and the start angle and size of each gap. All numbers in the problem will be integers.

The knights can only make their attempt at an integer time-step. They can slip through a gap of zero size e.g., if there is a gap in one layer from 0 to 90° and a gap in the next from 90 to 180°, then they can slip through both.

Example

In the example, each ring has one gap. Table 1 shows the rotation rate, gap position and gap size (all in degrees).

Speed	Gap start	Gap size
30	0	120
50	150	90
60	60	90
70	180	180
90	180	60

Table 1: Example test case

After 9 time steps, the gaps have rotated to cover angles 270–30°, 240–330°, 240–330°, 90–270°and 270–330°. At this time the knights can enter the castle at an angle of 270°.

Input (spin.in)

The input consists of five lines of integers (separated by spaces), describing the five rings of guards. The first integer R_i on each line is the speed of the guards, in degrees per time-step. The second integer, G_i, is the number of gaps in the ring. The remaining G_i pairs of integers (S_j, E_j) each describe the start angle and the extent of a gap. Gaps in the same ring will not overlap or touch.

Sample input

```
30 1 0 120
50 1 150 90
60 1 60 90
70 1 180 180
90 1 180 60
```

Output (spin.out)

If the knights will never be able to enter the castle undetected, output the single integer -1. Otherwise, output the integer time at which they can first enter the castle.

Sample output

```
9
```

Constraints

- $0 \leq R_i < 360$ for each R_i
- $1 \leq G_i \leq 5$ for each G_i
- $0 \leq S_j < 360$ for each S_j
- $1 \leq E_j < 360$ for each E_j

50% constraints

In 50% of the test cases there will be only one gap per ring.

Time limit

1 second.

Scoring

A correct answer scores 100%, an incorrect one scores 0%.

South African Computer Olympiad
Training Camp 1, 2006
Day 2

Stacking things

Author

Bruce Merry

Introduction

The society for putting things on top of other things is holding their annual meeting. They have built a large stack of things, each of which has a number on it. Apart from being obsessed with putting things on top of other things, the members of the society are obsessed with counting how far in the stack one thing is above or below another thing.

Task

You will be given various pieces of information about the relative positions of items in the stack. The society members are impatient, so you will also receive questions about the relative positions of items mixed in with the information. Sometimes the members are too impatient, and you will not yet have enough information to answer their questions.

The information will be of the form "thing A is N places above thing B". Some items of information will be redundant, but the information will always be accurate. The questions are of the form "how many places is thing A above thing B" (the answer can be negative if it is below).

Example

Suppose you are given the following information and asked the following questions, in order (answers to questions are shown in brackets):

- How many places is item A above item A? (0)
- Item A is 3 places above item B.
- Item A is 2 places above item C.
- How many places is item A above item C? (2)
- Item D is 5 places above item E.
- How many places is item E above item A? (unknown)
- Item B is 3 places above item D.
- How many places is item B above item C? (-1)

Note that the second-last question can be answered once all the information is available, but not enough information was available at the time.

Input (stdin)

You must read your input from standard input (as you would read from the keyboard). The lines of input are formatted in one of three ways:

`S` x y h This indicates that item x is h places above item y (where h may be negative).

`Q` x y This indicates the question "how many places is item x above item y?"

`E` This indicates the end of the input.

Note that after a question is read, no further reading is possible until an answer is supplied (see below). Any attempt to read further will block and your program will eventually time out.

Sample input

```
Q 1 1
S 1 2 3
S 1 3 2
Q 1 3
S 4 5 5
Q 5 1
S 2 4 3
Q 2 3
E
```

Output (stdout)

Immediately after reading a question from standard input, you must supply an answer on standard output (as you would write to the screen). The output is either a line containing an integer (if an answer is known), or else an empty line (if the answer cannot be determined). After writing a line of output, you must flush the output buffer.

Sample output

```
0
2

-1
```

South African Computer Olympiad
Training Camp 1, 2006
Day 2

Constraints

There are at most 100000 lines of input (excluding the terminating line). For each statement and question, $1 \leq x, y \leq 1000000 = 10^6$, and for each statement, $-1000 \leq h \leq 1000$.

Note that you should not make any assumptions about the total size of the stack; there may be arbitrarily many items with identifiers larger than 10^6.

50% constraints

In 50% of the test cases, there will be at most 1000 lines of input (excluding the terminating line).

Time limit

5 seconds.

Scoring

A correct answer scores 100% while an incorrect answer scores 0%.

South African Computer Olympiad
Training Camp 1, 2006
Day 2

Arranging Flowers

Author

Carl Hultquist

Introduction

Fred the manic store-keeper is arranging some flowers for the window of his store. He has one vase into which he can put the flowers, but he has bought too many flowers to all fit into this one vase so will need to choose which of the flowers he uses. Fred likes certain flowers more than others, and so he wants to pack in as many flowers as possible such that the overall beauty of the arrangement is best.

Task

Given the size of the neck of the vase, the number of flowers, and the stem-size and beauty value of each flower, you must find the arrangement of flowers that can fit into the vase and produce the greatest sum of beauty.

Example

Suppose the neck of the vase has size 5, and Fred has picked 4 flowers with the following stem-sizes and beauty values:

Stem-size	Beauty
2	4
1	3
3	4
4	7

Then the arrangement with the most beauty that can fit into the vase would consist of flowers 2 and 4, for a total beauty of 10.

Input (flowers.in)

The first line of input will contain two space-separated integers, N and M, the number of flowers and the size of the neck of the vase respectively. The next N lines of input will each contain two space-separated integers, S_i and B_i which are the stem-size and beauty respectively of the flowers available to Fred.

Sample input

```
4 5
2 4
1 3
3 4
4 7
```

Output (flowers.out)

The first line of output must contain two space-separated integers, T and K, the best total beauty and the number of flowers used in the arrangement respectively. The next K lines must each contain a single integer, which are the indices of the flowers used in the arrangement. If there is more than one arrangement which gives this total beauty, you only need to output one (and you can output any such arrangement).

Sample output

```
10 2
2
4
```

Constraints

- $2 \leq N \leq 1000$
- $1 \leq M \leq 10000$
- $1 \leq S_i \leq 1000$
- $1 \leq B_i \leq 10000$

50% constraints

- $2 \leq N \leq 20$

Time limit

1 second.

Scoring

A correct answer will score 100%. An incorrect answer, or one in which the output format is incorrect or the arrangement specified doesn't give the reported total beauty, will score 0%.