



South African Computer Olympiad Training Camp 3, 2005 Day 2



Overview

Author	Timothy Stranex	Graham Poulter	Nick Pilkington	Keegan Carruthers-Smith
Problem	mix	flights	tiles	jobs
Source	mix.c mix.cpp mix.pas	flights.c flights.cpp flights.pas	tiles.c tiles.cpp tiles.pas	jobs.c jobs.cpp jobs.pas
Input file	stdin	flights.in	tiles.in	jobs.in
Output file	stdout	flights.out	tiles.out	jobs.out
Time limit	1 second	5 seconds	1 second	5 seconds
Number of tests	10	5	10	10
Points per test	10	20	10	10
Total points	100	100	100	100

The maximum total score is 400 points.





South African Computer Olympiad Training Camp 3, 2005 Day 2



Mixed Resistors

Author

Timothy Stranex

Introduction

After a long search, Url has found a resistor of exactly π Ohms. Unfortunately, his arch-enemy Lru learned of his discovery and caused the special resistor to be mixed among Url's extensive resistor collection.

Task

Since Lru also stole Url's multimeter, all Url has to differentiate between the resistors is a median device which given three resistors will tell him which one has the median resistance. (For example, given 1, 4 and 3 Ohm resistors, the device will indicate that the 3 Ohm resistor is the median.) It turns out that there are an equal number of resistors smaller than π as there are greater than π in Url's collection. All of Url's resistors have distinct resistances. The resistors are labeled from 1 to N . Help Url find the special resistor among his collection.

Communication

Your program will communicate with the median device through stdin and stdout. Initially, the median device will output a single integer N followed by a newline. Once your program has read in this integer, it can send two types of command to the median device.

To send a median query, output a 1 followed by three space-separated integer resistor labels r_1 , r_2 and r_3 and a new line to stdout. The median device will respond with a single integer followed by a newline. This integer will be the label of the median resistor. r_1 , r_2 and r_3 should be distinct numbers. If they are not, the median device will respond with a 0. The median device can be queried at most $Y = 15000$ times before it breaks down, in which case your program will terminate.

Once the label on the π resistor is found, output a 2 followed by the label and exit.

Remember to flush stdout after writing to it:

Pascal: flush(stdout);

C: fflush(stdout);

C++: cout << flush;

To use the test facility on the web page, you may upload an input file, although this input file will not be available to your program. The first line contains N . The remaining N lines contain resistor values, as distinct positive integers.

Example

Suppose $N = 5$ and the resistor values are:

Resistor	1	2	3	4	5
Value	π	1	5	2	4

Then, the following would be a valid communication with the median device:

```
>> 3          # N = 3
<< 1 1 2 3    # median(r1, r2, r3)
>> 1          # = r1
<< 1 4 2 1    # median(r4, r2, r1)
>> 4          # = r4
<< 1 5 2 4    # median(r5, r2, r4)
>> 4          # = r4
<< 1 5 4 1    # median(r5, r4, r1)
>> 1          # = r1
<< 1 5 1 3    # median(r5, r1, r3)
>> 5          # = r5
<< 2 1        # Answer: r1 = pi
```

Constraints

- $3 \leq N \leq 2001$ and N is odd
- $Y = 15000$

50% constraints

- $3 \leq N \leq 901$

Time limit

1 second.

Scoring

If your program outputs the correct answer with at most Y queries to the median device, you will score 100% for the test-case. If your program outputs the incorrect answer or deviates from the median device communication protocol, it will score 0% for that test-case.





South African Computer Olympiad

Training Camp 3, 2005

Day 2



One Way Flights

Author

Graham Poulter

Introduction

Dogbert has recently purchased Elbonia, his latest step towards world domination, and is touring his acquisition by plane. Unfortunately Elbonian Airlines only does one-way flights, and only some of these are regular enough (more than once a year) to be usable. This means there are cities in the country that he could fly to and not be able to get back from for a year.

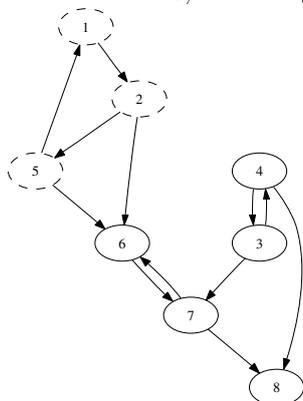
Being a busy man — um, dog — Dogbert has resigned himself to not seeing the whole country, but has enlisted your help to find the largest subset of cities that he can tour freely.

Task

Given N cities and their regular one-way flights (e.g. from 1 to 3), output the size S of the largest group of cities for which it is possible to start from any city in the group and travel to any other city in the group.

Example

Consider the figure below. There are $N = 8$ cities (and 12 flights), and the size of the largest groups of mutually reachable cities is 3, consisting of cities 1,2 and 5.



Input (flights.in)

The first line of the input contains the number of cities N . The next N lines each begin with an integer F which is the

number of flights leaving that city. This is followed by F space-separated integers, $b_1 \dots b_F$, where each $1 \leq b_i \leq N$ represents the destination of a flight leaving that city.

Sample input

```
8
1 2
2 5 6
2 4 7
2 3 8
2 1 6
1 7
2 6 8
0
```

Output (flights.out)

Output is one line, the size of the largest group of mutually reachable cities.

Sample output

```
3
```

Constraints

- No city will have a flight to itself.
- Each city has at most one flight to any particular city (no double-flights)
- $1 \leq N \leq 10000$
- There will be between 0 and 10^6 flights (inclusive).

40% constraints

- $1 \leq N \leq 1000$
- There will be between 0 and 10^4 flights (inclusive)

Time limit

5 seconds.

Scoring

There are 5 test cases. In each test case, a correct answer is worth 20 marks and an incorrect answer is worth 0 marks.





South African Computer Olympiad Training Camp 3, 2005 Day 2



Sliding Tiles, Again!

Author

Nick Pilkington

Introduction

Not too long ago an unnamed IOI squad was given a practice task of solving the famous sliding tile puzzle. Unfortunately the team didn't perform all that well in the task and wild rumors of perversely obscure test data ran riot after the competition. Of course not doubting their own abilities as programmers the squad suspected foul play on the part of the problem author. They have managed to get hold of the data used and now wish to ascertain whether or not it is possible to solve the problem at all!

A sliding tile puzzle is simply a square grid of tiles, with one tile missing. The tiles have the numbers 1 to $N^2 - 1$, and the goal of the puzzle is to use the blank space to move the tiles around into a specific order. In particular, the tiles must be in ascending order starting from the top-left corner, moving from left to right and then continuing at the left side of the next row down. Finally the empty tile should be in the bottom right corner. See Table 1 for a graphical illustration:

1	2	3
	4	6
7	5	8

Table 1: A graphical illustration of the sliding tile puzzle. This is solved by first moving tile 4, then tile 5, and finally tile 8.

Task

You will be given the initial state of the sliding tiles puzzle and will have to determine whether or not the puzzle can actually be solved by some finite number of legal moves.

Example

Consider the sliding tile puzzles in Table 2.

The left puzzle can be solved; however, the right one cannot be solved!

1	2	1	3
	3	2	

Table 2: An example

Input (tiles.in)

Line 1: Two integers T , N . T represents the number of squares to follow in the file (note there is possibly more than one square to solve in each file). N represents the dimensions of all of the squares to follow.

Line 2.. $N \times T + 1$: Each line contains N integers representing a row of tiles (0 indicates the blank)

Sample input

```
1 3
1 2 3
4 5 6
7 0 8
```

Output (tiles.out)

The output file should contain a single line containing a bit string of length T . Each character should refer to the square of the same position. A 1 should indicate that the case is possible, and a 0 that it is not.

Sample output

```
1
```

Constraints

- $1 \leq T \leq 5$
- $1 \leq N \leq 150$

50% constraints

- $1 \leq T \leq 3$
- $1 \leq N \leq 4$

Time limit

1 second.





South African Computer Olympiad Training Camp 3, 2005 Day 2



Scoring

If the output is in any way invalid you will score 0%. If it represents the correct bit string for the input data you will score 100% and will have redeemed the pride of the unnamed squad!





South African Computer Olympiad

Training Camp 3, 2005

Day 2



Jobs

Author

Keegan Carruthers-Smith

Introduction

A businessman recently approached you to help him improve his business' efficiency. He is having problems with his current program he is using. The program is too slow and cannot deal with removal of clients. The program helps him know which client he has to deal with next.

Task

Your task is to develop a program to help him find out what client he needs to deal with next, quickly. The businessman gives each client an unique ID and a priority.

He requires the program to do the following:

- Tell him the next client he needs to deal with and then automatically remove the client. The next client is always the client with the highest priority. If two or more clients have the same priority, pick the client with the lowest ID.
- Update a client's ID and priority (if the client does not already exist, it should be added).
- Remove a client (do nothing if the client does not exist).

He wants to test the program before using it, so he has given you 10 test files which contain what would have been typed into the program. You must output in what order he would deal with the clients.

Example

Suppose he takes the following actions:

- Add/update client 34 to priority 30
- Add/update client 21 to priority 30
- Add/update client 50 to priority 30
- Asks for the next client: you return 21 and delete the client.
- Asks to remove client 50.
- Asks for the next client: you return 34 and delete the client.

- Asks for the next client: you return nothing, since there no clients left.
- Add/update client 21 to priority 999
- Add/update client 21 to priority 1
- Removes client 5 (do nothing, since the client does not exist)
- Asks for the next client: you return 21.

Input (jobs.in)

The first line of input contains N , the number of commands in the file. The following N lines can each possibly contain the following:

- 1 asking for the next client.
- 2 ID_n $PRIORITY_n$ Three space-separated integers, indicating to add/update a client.
- 3 ID_n Two space-separated integers, indicating to remove a client.

Sample input

```
11
2 34 30
2 21 30
2 50 10
1
3 50
1
1
2 21 999
2 21 1
3 5
1
```

Output (jobs.out)

For each time the command 1 is in the input file, you need to write the next client's ID on a line, if there is another client. The final line contains a 0.

Sample output

```
21
34
21
0
```





South African Computer Olympiad Training Camp 3, 2005 Day 2



Constraints

- $1 \leq N \leq 150000$
- $1 \leq ID_n \leq 10000$
- $1 \leq PRIORITY_n \leq 1000000$

50% constraints

- $1 \leq N \leq 1000$
- $1 \leq ID_n \leq 500$

Time limit

5 seconds.

Scoring

You will 100% for the correct output file, otherwise you will score 0% for the test case.

