



boxes

100 points

Source code: `boxes.c`, `boxes.cpp`, `boxes.pas`

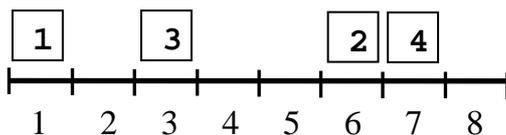
Time limit: 10 s

Memory limit: 32 MB

Task

A circular belt has N positions (numbered from 1 to N), each of which can hold a single box. At any point in time, some positions have boxes, and the remaining are empty. You can move a box by pushing it on the belt to any position, provided you do not skip over a box. Pushing is circular (a box at position N can be pushed to position 1 if it's free).

Below you have an example belt with $N=8$ and 4 boxes (each with an id from 1 to 4). In this example, you can push box 3 to positions 2, 4 or 5, box 1 to positions 2 and 8, box 4 to position 8, and box 2 to positions 4 and 5.



Your task is to support a number of box insertions on the belt. Each new box comes with a requirement for its location: it must be placed after a specific box b_i and before the next one on the belt (in circular order). Before placing the new box, you can move the boxes already on the belt to make space. In the above example, to insert a new box between 2 and 4, at least one of them has to be moved.

Boxes are identified by id numbers starting at 1, in order of insertion. Initially, there are $N/4$ boxes on the belt that your program can place at its discretion. Thereafter, $N/4$ more boxes will be inserted. Before inserting each new box, you are allowed to make at most 200 moves (you can receive partial credit if you make at most 500 moves).

Interaction

Your program should not read or write any files. Instead, it will interact with another program provided by the scientific committee that will run at the same time with your program. The interaction will take place in the following way:

1. Your program will read one line from standard input containing an integer N .
2. Initially, you will place $N/4$ boxes on the band as you wish. Your program should write $N/4$ lines to the standard output, in the following format: $\mathbf{I \ p}$
The k -th line specifies the position \mathbf{p} on the belt, where you want to place box number k . All positions must be distinct.
3. You will be required to insert $N/4$ more boxes, following this protocol:
 - Your program will read one line from standard input containing an integer b_i . The new box must be inserted on any position after box number b_i and before the next box on the belt, and it will have the next available box number.



Central European Olympiad in Informatics
Tîrgu Mureş, Romania
July 8 – 14, 2009
Day 1

- Your program will write several lines to standard output, each describing a move operation in the following format: **M b p**
Such a line means “move box number **b** to position **p**.” Remember that there should be an empty path from the current position of box **b** to the final position **p**.
 - When you are done moving boxes, your program will write one line to standard output in the following format: **I p**
Such a line means “insert the new box on position **p**”; remember that **p** should be after the position of box **b_i** and before the next box in circular order.
4. After exactly **N/4** insertions your program must finish without any further interaction.

Constraints

- In all test cases, **N = 20 000**
- The committee’s program will use various strategies for deciding the position of new boxes.
- Your score for each test is based on the **maximum** number of moves made before any insertions:
 - You gain 100% if you never use more than 200 moves
 - You gain 70% if you never use more than 300 moves
 - You gain 40% if you never use more than 500 moves

Programming instructions

After every complete line written to the standard output, C programmers must use `fflush(stdout)` function while Pascal programmers must use `flush(output)` procedure.

<i>C</i>	<i>C++</i>	<i>Pascal</i>
<code>printf("I %d\n", p); fflush(stdout);</code>	<code>cout<<"I "<<p<< '\n'; cout.flush();</code>	<code>writeln('I ', p); flush(output);</code>

Submission tests

At submission time, your program will be evaluated with three different interactive programs. The behavior of these is as follows:

1. All insertions are after box **1**
2. Insertions are made after randomly selected boxes
3. Finds an interval of the belt that contains many boxes, and inserts inside it.

The interactive programs used during official grading will use other strategies.



Central European Olympiad in Informatics
Tîrgu Mureş, Romania
July 8 – 14, 2009
Day 1

Example

Read N	8
Write I 1 Write I 5	
Read b1 Write I 2	1
Read b2 Write M 1 7 Write I 1	1

The belt will have the following arrangement:

