

# Tetris AI

In the very heart of a well known producer of microelectronic products, a mobile phone with a built in game of Network Tetris is being prepared for release. The owners of such mobile phones can arrange duels when at a small distance from each other. Data transmission between players is carried out using the Bluetooth protocol.


However -- now we are coming to the point -- it sometimes happens that there may be no other similar phone nearby and the player may need to play alone. For this purpose it is necessary to write a computer player (AI) with a very hard difficulty level.


The rules of Network Tetris are pretty simple :

The game has two playing fields, each with the rules of standard Tetris: figures of 4 blocks keep falling from the top of the field, and have to be placed in such a way as to form horizontal lines. Once a line is filled up, it is removed and all lines above it are appropriately shifted downwards. There is however one difference with respect to standard Tetris -- a player receives additional penalty lines as soon as his opponent clears a line. The game is over when one of the players fills his own field, either on his own or with his opponent's help, to such an extent, that the next figure cannot fully enter the field.

The width of the field is 10, and the height of the field is 20. There are 6 types of figures in the game:


I (1) - 

L (2) - 

J (3) - 

Z (4) - 

S (5) - 

O (6) - 



It is your task to write a bot which starts with an empty field and, knowing the sequence of figures dropping on its field, plays in such a way as to do as much harm as possible to the opponent.

## Input

$t$  – number of test cases [ $t \leq 150$ ], then  $t$  tests follow.

Each test case starts with integer  $N$  equal to the number of figures which drop onto the field [ $10 \leq N \leq 50000$ ]. Then  $N$  integers follow, numbered from 1 to 6 – denoting one of the figures, in the same position as they appear in the pictures. (Look at numbers in parenthesis).

## Output

For each test you should output line *case 1 Y* if you wish to solve this test case or *case 1 N* otherwise. If you output *Y*, then exactly *N* lines must follow. Each of them should contain exactly two integers: *A* and *X*, where *A* is the clockwise angle of rotation in the given move, numbered from 0 to 3 (0 - 0 degrees, 1 - 90 degrees, 2 - 180 degrees, 3 - 270 degrees), while *X* is the horizontal coordinate of leftmost cube of the figure [ $1 \leq x \leq 10$ ]. The *i*-th figure at input corresponds to the *i*-th figure at output. If the figure falls outside the field or the parameters have incorrect value, or any falling figure stops with at least one cube in a line of number larger than 20, then the solution will be judged as Wrong Answer.

## Score

The score will be equal to the total number of cleared lines, taken over all test cases. For one cleared line your solution will receive 1 point, for two simultaneously cleared lines – 5 points, for three simultaneously cleared lines – 15 points, for four simultaneously cleared lines – 30 points.

## Example

### Input :

```
1
14
3
2
4
5
3
2
1
6
6
1
1
4
1
5
```

### Output :

```
case 1 Y
0 1
0 8
0 1
0 8
1 7
3 3
1 5
0 7
0 9
0 1
1 6
1 6
0 1
1 4
```

### Score :

```
score = 30 + 1 = 31
```