American Computer Science League

2020 Finals ● Program 1: Spot the Y ● Junior Division

●	2	●	4	5
6	●	■	9	10
11	●	13	■	■
16	17	■	19	20
21	22	23	24	25

1	2	3	4	■
6	7	■	■	10
11	●	13	14	■
16	●	18	19	20
●	22	●	24	25

PROBLEM: *Spot-The-Y* is a game played by 2 players on an N x N grid. The grid squares are numbered as above, with 1 is in the upper left corner, and N^2 is in the lower right corner. Players in turn place a marker (circles for the first player and squares for the second player) in an empty grid space or remove one of their own markers from the grid. The winner is the first player to create a "Y" with his own markers.

A "Y" pattern has two markers adjacent horizontally or vertically, and two other markers diagonally connected to one of the adjacent ones. The grids above show the 4 possible orientations of a "Y" pattern.

INPUT: There will be 10 lines of data with each representing a game of *Spot-The-Y.* Each line will contain an integer, *N*, giving the size of the square grid. That will be followed by a series of integers (1 through N^2), giving the moves of the two players in alternating order. If the move directs a player to a grid square that is empty, that player places his marker in that square. If the grid square contains his marker, it is removed. There will not be any input where the grid square is occupied by an opponent's marker.

OUTPUT: For each game, print the sum of the grid square numbers that produced the first ''Y" found. If there is no Y found after all of the moves, print 0.

SAMPLE INPUT:

```
 5 1 14 12 18 3 15 7 8
 5 1 14 24 20 12 18 3 15 12 20 17 8
 6 23 1 21 5 23 8 14 36 16 12 27 7
```

SAMPLE OUTPUT:

```
1. 23
2. 55
3. 78
```

TEST DATA

TEST INPUT:

```
5 8 1 25 24 25 11 8 7 22 19 15 8 22 16 21 3
6 25 21 13 15 20 10 27 1 14 3 32 28 9 36 14 29 22  1
6 36 24 29 22 17 11 29 27 17 27 34 17 23 35 29
8 3 10 46 12 3 13 37 12 54 19 39 27 39 12
8 12 1 19 10 3 17 19 1 5 2 28 19 28 17 20
4 3 5 10 11 10 12 6 10 1 14 13 12 4 14 14 7 9 11 8 2 8 11 13 15
7 10 31 15 23 18 25 16 45 18 30 46 38 24 25 15 25 32 22 10
9 10 20 30 2 4 6 8 3 9 12 15 18 21 24 27 5 25 7 14 11 17 1
10 99 88 80 78 69 97 60 68 99 78 67 68 67 78 68 99
3 1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1 1 3 5 7 8 6 4 2 5 5
```

TEST OUTPUT:

1. 27
2. 86
3. 74
4. 176
5. 48
6. 0
7. 117
8. 77
9. 277
10. 23