Problem E
A Smart Knight's Tour on a Chessboard

Input File: *testdata.in*
Time Limit: 10 seconds

Problem Description

Given an n by n chessboard, a knight is allowed to move according to the rules of chess. A *knight's tour* is a sequence of moves of a knight on a chessboard such that the knight visits every square exactly once. The knight's tour problem is an instance of the Hamiltonian path problem in graph theory. Given a starting position (i, j) in an n by n chessboard, you are asked to find out a knight's tour. For example, for a 6×6 chessboard, if the starting position is $(4, 2)$, then the last matrix shown in sample output records the sequence of the knight's tour. That is, the first move is at position $(4, 2)$, the second move is at position $(6, 1)$, the third move is at position $(5, 3)$, and so on.

A smart searching strategy is to compute a weight for each candidate field. The weight of a candidate field is the number of feasible moves from that field. The weight of next move must be as small as possible. In case of equal weight, the field is chosen if it is encountered earliest in clockwise order (i.e., from the top to the right, then down and then to the left, and back up to the top).

Technical Specifications

1. In each test case, the chessboard is at least 5 by 5.

2. The starting field can be any field in the chessboard except the field with no solution.

Input Format

The first line contains an integer indicating the number of test case. For each test case, the first line contains an integer indicating the size n of the chessboard and the second line contains two integers, separated by a space, indicating the starting position of the knight's tour.

Output Format

For each test case, output one solution of the knight's tours with a chessboard matrix. In the output matrix, the length of each entry is exactly 5 digits with right alignment. Besides, output an empty line between two test cases.

Sample Input

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

Sample Output

```
   21    12     7     2    19
    6    17    20    13     8
   11    22     1    18     3
   16     5    24     9    14
   23    10    15     4    25

   21    14     3     8    19
    4     9    20    15     2
   13    22    11    18     7
   10     5    24     1    16
   23    12    17     6    25
```

```
31   14    1   22   25   34
 2   23   32   35    8   21
13   30   15   24   33   26
16    3   36    9   20    7
29   12    5   18   27   10
 4   17   28   11    6   19

25   32    7   16   19   34
 8   17   26   33    6   15
31   24    9   18   35   20
10    1   36   27   14    5
23   30    3   12   21   28
 2   11   22   29    4   13
```