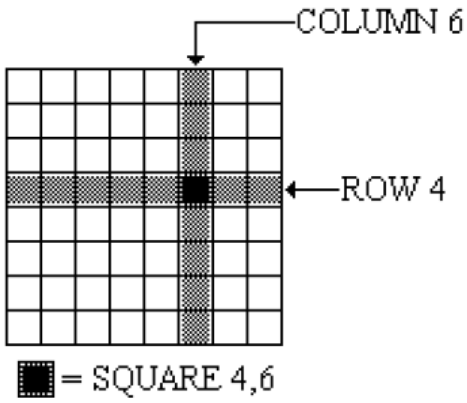
In chess it is possible to place eight or N queens on the board so that no one queen can be taken by any other. Write a program that will determine all such possible arrangements for N queens given the initial position of one of the queens.

Do not attempt to write a program which evaluates every possibility. For example there are 8 configuration of 8 queens placed on the board. This would require 8^8 evaluations and would bring the system to its knees. There will be a reasonable run time constraint placed on your program.

Input

Input to your program will be the N size of the board and two numbers separated by a blank. The numbers represent the square on which one of the N queens must be positioned. A valid square will be represented; it will not be necessary to validate the input. To standardize our notation, assume that the upper left-most corner of the board is position (1,1). Rows run horizontally and the top row is row 1. Columns are vertical and column 1 is the left-most column. Any reference to a square is by row then column; thus square (4,6) means row 4, column 6.



Output

Output from your program will consist of a one-line-per-solution representation. Each solution will consist of N numbers. Each of the N numbers will be the ROW coordinate for that solution. The column coordinate will be indicated by the order in which the N numbers are printed. That is, the first number represents the ROW in which the queen is positioned in column 1; the second number represents the ROW in which the queen is positioned in column 2, and so on, as follow:

Solution
c1 c2 c3 c4 c5 c6 c7 c8 … cN
…

Include the headings as shown below in the sample output and print the solutions in lexicographical order.

Sample Input

8
1 1
5
1 1

Sample Output

Solution
1 5 8 6 3 7 2 4
1 6 8 3 7 4 2 5
1 7 4 6 8 2 5 3
1 7 5 8 2 4 6 3
Solution
1 3 5 2 4
1 4 2 5 3