Spies, Revised

by HackerRank

Problem Submissions Leaderboard Discussions

Two spies in a grid will have their covers blown if:

1. They are both in the same row.
2. They are both in the same column.
3. They can see each other diagonally (i.e., lie in a line inclined $45°$ or $135°$ to the base of the grid).

The level of danger is now increased! In addition to the conditions above, *no* 3 *spies may lie in any straight line*. This line need not be aligned $45°$ or $135°$ to the base of grid.

Write a program in the language of your choice to place N spies (one spy per row) on an $N \times N$ grid without blowing anyone's cover. Your program must then print the following 2 lines describing a valid configuration:

1. The value of N.
2. A space-separated list of 1-indexed column numbers, where each value i is the column number of the spy in row i (where $1 \leq i \leq N$).

Solve this problem for N as large as possible, up to (and including) 1000.

Note: *Run* and *Custom Input* are not available for this challenge; you must click *Submit Code* for your submission to be scored. Your score for this challenge will always be the maximum value scored by any of your submissions.

Examples
In the examples below, S denotes a spy and * denotes an empty cell.

Sample Configuration 0

A valid configuration for $N = 11$:

```
* S * * * * * * * * *
* * * S * * * * * * *
* * * * * * S * * * *
S * * * * * * * * * *
* * * * * * * S * * *
* * * * * * * * * * S
* * * * S * * * * * *
* * S * * * * * * * *
* * * * * * * * S * *
* * * * * S * * * * *
* * * * * * * * * S *
```

Sample Output 0

This C++ code:

```
#include <stdio>
using namespace std;

int main(){
    cout << "11\n" ;
```

```
    cout << "2 4 7 1 8 11 5 3 9 6 10" ;
    return 0 ;
}
```

Produces this output:

```
11
2 4 7 1 8 11 5 3 9 6 10
```

This configuration will earn a score of $11/10 = 1.1$.

Sample Configuration 1

A valid configuration for $N = 13$:

```
S * * * * * * * * * * * *
* * S * * * * * * * * * *
* * * * * * * * * * * S *
* * * * * * * * * S * * *
* * * * * * S * * * * * *
* S * * * * * * * * * * *
* * * * * * * * * * S * *
* * * * S * * * * * * * *
* * * * * * * S * * * * *
* * * * * * * * * * * * S
* * * * * * * * S * * * *
* * * S * * * * * * * * *
* * * * * S * * * * * * *
```

Sample Output 1

This Python code:

```
print "13"
print "1 3 12 10 7 2 11 5 8 13 9 4 6"
```

Produces this output:

```
13
1 3 12 10 7 2 11 5 8 13 9 4 6
```

This configuration will earn a score of $13/10 = 1.3$.

Sample Configuration 2

An invalid configuration for $N = 7$:

```
S * * * * * *
* * S * * * *
* * * * S * *
* * * * * * S
* S * * * * *
* * * S * * *
* * * * * S *
```

Sample Output 2

The following output:

```
7
1 3 5 7 2 4 6
```

Will earn a score of 0 because the spies in the first 3 rows are in a straight line.

Input Format

There is no input for this challenge.

Constraints

- N is odd.
- $N \leq 1000$ (*Do not* submit for any value of N larger than 1000)

Scoring
A correct configuration will get a score of $\frac{N}{10}$.

Output Format

Print the following 2 lines of output:

1. The first line should be a single integer denoting the value of N.
2. The second line should contain a space-separated list of integers. Each integer i (where $0 \leq i < N$) should be the 1-indexed column number where the spy in row i is located.

Solved score: 99.90pts

Submissions:2539

Max Score:100
Difficulty: Expert

Rate This Challenge:

More

Current Buffer (saved locally, editable)

Java 7

```
1  import java.io.*;
2  import java.util.*;
3  import java.text.*;
4  import java.math.*;
5  import java.util.regex.*;
6
7  public class Solution {
8
9      public static void main(String[] args) {
10         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
11     }
12 }
```

Line: 1 Col: 1

Upload Code as File

Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature