You have a classroom of N^2 students, that make up N rows and N columns. Half of the class are smart students and the other half aren't. The teacher needs to position the smart and non-smart students in a way such that he/she thinks that the student can't cooperate (cheat) during the exam. The teacher comes up with a strategy that he/she thinks will work:

1. Every row should have the same ratio of smart students to non-smart students
2. Every column should have different ratios of smart to non-smart students (i.e. no two columns should have the same ratio of smart to non-smart students)
3. The ratio of smart to non-smart students sitting in the diagonals of the class should be maximized (i.e. the ratio of diagonals should be as different from each other as possible)

Your task is to help the teacher come up with a seating plan for the students (i.e. propose a seating plan), that will conform to the teachers strategy outlined above.

Example:

S - Smart student
N - Non-smart student

S S S N N **N**
S **S** S N **N** N
S S **S** **N** N N
S S **S** **N** N N
S **S** S N **N** N
S S S N N **N**

The ratio is calculated as the number of smart student to non-smart students = # of smart students/ # total number of students in (row, column or diagonal)

The example above follows rule (1), where the ratio of smart students to non-smart students is the same for each row, but violates rule (2).

Also, the four corner students highlighted in red are sitting on 4 of the 22 diagonals that exist.

Diagonal:	Ratio:
S	1/1 = 1
S	1/1 = 1
N	0/1 = 0
N	0/1 = 0

The two longest diagonals have ratios:

Diagonal:	Ratio:
S S S N N N	3/6 = 1/2
N N N S S S	3/6 = 1/2

Input:
The input will be N, a single positive even number (an integer that is divisible by 2). N will be passed to the program as an argument.

Output:
N lines each with N characters, either 'S' (smart student) or 'N' (non-smart student).

Example Execution:
root@dev:~# make
root@dev:~# ./ findSeatingPlan 2
SN
SN

Note that you should include a make file that will generate an executable called findSeatingPlan

In a programming language of your choice, write a program that takes N as an argument, and outputs a solution as a matrix (N x N), N lines each with N characters, either 'S' (smart student) or 'N' (non-smart student).

Please submit the following:

1. Source code of the solution (in any language of your preference)
2. Output generated by the code
3. Initial estimate of time required for the task (i.e. planning, design, implementation, testing)
4. Actual time taken for the task (actual time you end up spending)
5. A few words explaining the approach/algorithm (as a separate file or inside the code itself)
6. DO NOT include compiled binary code (.exe) as it could be rejected by mail servers/clients.