VMWare Coding Problem

Sanchita Roy
Contact : +1 352 871 5385; sroy.sanchita@gmail.com
linkedin.com/in/sanchitaroy | github.com/sanchitaroy

Requirement Statement :

Provided, n is a multiple of 2. Ratio of smart students to non-smart students is 1:1 for the whole class. The objective is to achieve a seating arrangement such that all the below are considered :

1. Every row has the same ratio of smart students to non-smart students
2. Every column has different ratios of smart to non-smart students
3. The ratio of smart to non smart students is as different as possible in every diagonal.

Estimation :

Task	**Time Estimated (in hour-per-person)**	**Time Taken (in hour-per-person)**
Analysis and Requirement Clarification	1	0.5
Technical Design	2	1.5
Implementation	2	2
Testing and Bugfixes	2	2.5
Total	**7**	**6.5**

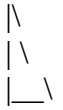
Solution Architecture :
Language used for Implementation : Java

Let Smart student be represented as S. Non-smart student be represented as N. User input 'n' is the dimension of the matrix (number of row or column)

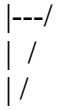
Since N:S = 1:1, and n is fixed for all rows (square matrix), hence, for each row, N:S must be 1:1. Also, if all columns have different ratios, each column will have a different number of N or S. For all ratios to be different, hence, the columns will have N or S ranging from 0 to n/2-1 and n/2+1 to n [if any column have N or S as n/2, it violates either the row ratio or the column ratio property].

Approach taken :

1. Fill the matrix till n/2+1 rows with n/2 'N' from the left and then remaining n/2 'S'
2. For the rest of the matrix :
    i. From the left, fill lower/left triangular matrix with 'N' till column n/2-1
        ie -



    ii. Fill upper left triangular matrix with 'N' from column n/2 to n-1
        ie -



    iii. Fill the rest by 'S'

Sanchita Roy
Contact : +1 352 871 5385; sroy.sanchita@gmail.com
linkedin.com/in/sanchitaroy | github.com/sanchitaroy

3. Swap column 1 and column n in the resultant matrix.
4. Swap row 1 and n/2+1 in the resultant matrix.

Testing Code :
- The program checks if 'n' is a multiple of 2. If not, the program terminates and a re-run is required.
- A function 'validate' is implemented additionally to check if for any value of 'n', any of the two mentioned properties (property 1 and property 2) is violated according to the implementation. The program exits in that case without printing the seating arrangement.
- The program also prints :
    i. The ratio N:S for the rows.
    ii. All the distinct ratios of the diagonals and the count of diagonal per ratio.
    iii. The count of distinct diagonals.

Running Instructions :

To run the program,
    Go to folder containing the jar file.
    Type : **java -jar findSeatingPlan.jar <*dimension*>**

Sample Output :