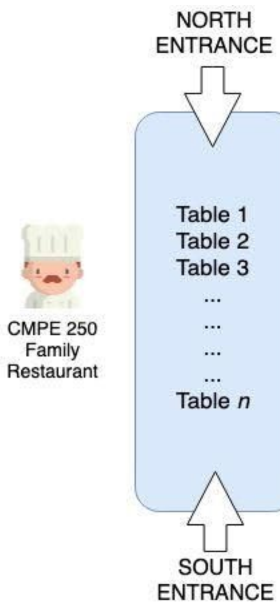
CMPE250 – Assignment 1

(Due Date: 15.10.2019 Day - 23:00)

Description

There is a restaurant called CMPE250 Family Restaurant in Rumeli Hisarüstü. This restaurant is located between the North and South Campus of Bogazici University. You are expected to assign each incoming customer to a table by considering the following constraints:

- This restaurant has n number of tables, where n is an integer between 5 to 50 ($5 \leq n \leq 50$).
- There are two entrances to this restaurant, where one of them is located at the north side of the restaurant, and the other one is located at the south side of the restaurant.
- Tables are named incrementally from north to south.
    - For instance, if $n = 12$, [north entrance –> 1 2 3 4 5 6 7 8 9 10 11 12 <– south entrance].



- The smallest table number is 1.
- If a customer enters the restaurant via the north entrance, he/she will sit to the closest empty table to the north entrance.
- If a customer enters the restaurant via the south entrance, he/she will sit to the closest empty table to the south entrance.
- Initially, each table is empty.

Here are some additional information to ease your job:

- Tables in the restaurant are for one person each, and customers arrive as single.
- You don't need to consider the case where a customer arrives and cannot find an empty table. In our test scenarios, whenever a customer arrives, there will already be an empty table for that person.

To implement this project, you will be provided some code written by the assistants, and you are going to implement the missing part.

The files you will be provided are: Table.h, Table.cpp, Restaurant.h, Restaurant.cpp, and main.cpp.

The file you should fill in is: **Restaurant.cpp**. You only need to complete the **execute()** method. So, for this project any code written outside Restaurant.cpp will be useless.

Input

Your code will be tested with a text file (i.e., input.txt). This text file consists of two lines.

- In the first line, total number of tables (n) is stated in the integer format (e.g., 15, 23).
- In the second line, there is a list of strings, where each string is separated with a comma.
  - "N" means that customer arrives from the north entrance.
  - "S" means that customer arrives from the south entrance.
  - If there is an integer (let's call it i) in the second line, it means that customer, who was sitting in table i, leaves.

The input file will not send a customer into the restaurant if it is already full. Moreover, i will always refer to a table occupied by a customer.

NOTE: The main.cpp as provided already receives the input in this format.

Output

Your code is expected to produce a text file (i.e., output.txt) as an output. This text file should output each table's final assignment status, where 0 represents an empty table, and 1 represents an occupied table.

NOTE: The main.cpp as provided already prints the output in this format.

The code as provided also produces some output to terminal for debugging purposes, which is fine. You can print anything you like onto terminal. The output to the file output.txt is all that counts.

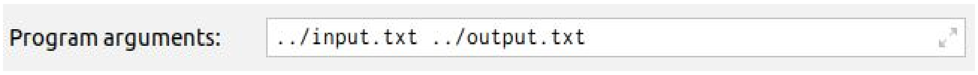
How to Compile?

Your code must be compiled by the simple *cmake CMakeLists.txt* and *make* commands. Also, your code must read the arguments from the command line. Your code will be tested with the command:

./cmpe250-assignment1 inputFile outputFile

Instructions for CLion users:

- Navigate to "VCS -> Checkout from Version Control -> Git" from the top menu bar, and paste git url to **clone** the git repository.
- You can **build** the executable by selecting "Build->Build Project" from the top menu bar.
    - CLion will create "cmake-build-debug" directory and there is an executable of the program with identical name as Clion Project Name.
- **Run** the program with command line arguments:
    - **Option1:** Navigate to "Run->Edit Configurations" and **set the "Program Arguments"** field as following:
        - *"../input.txt ../output.txt"*.
        - You can start the program by selecting "Run" from the menu.



Program arguments: ../input.txt ../output.txt

    - **Option 2:** You can run the program directly from the CLion terminal with following command:
        - ./cmake-build-debug/cmpe250-assignment1 input.txt output.txt
    - **Note:** After running, you can notice that program also writes the requested output to "output.txt" automatically. Refer to the "Details of the Given Code" section for more details.
- You can **commit** modified files by navigating to "VCS -> Git-> Commit File" at the top bar.
    - You can select the files you want to commit. Enter a commit message, then click the "Commit" button at the bottom right to finish the process.
- To **push** the changes, navigate to "VCS -> Git -> Push" then confirm the dialog.

Instructions for text editor users:

- Open the terminal, then **clone** the git repository with:
    - *git clone "repository-link"*
- Change to the project directory, then use **cmake** to create makefile describing build instructions:
    - *cmake CMakeLists.txt*
- **Build** the program using the generated makefile.
    - *make*
- **Run** the program with the recommended arguments:
    - ./cmpe250-assignment1 input.txt output.txt
- For **commit** and **push** instructions, you can refer to the CMPE250 Virtual Machine Setup Guide document shared on the moodle.

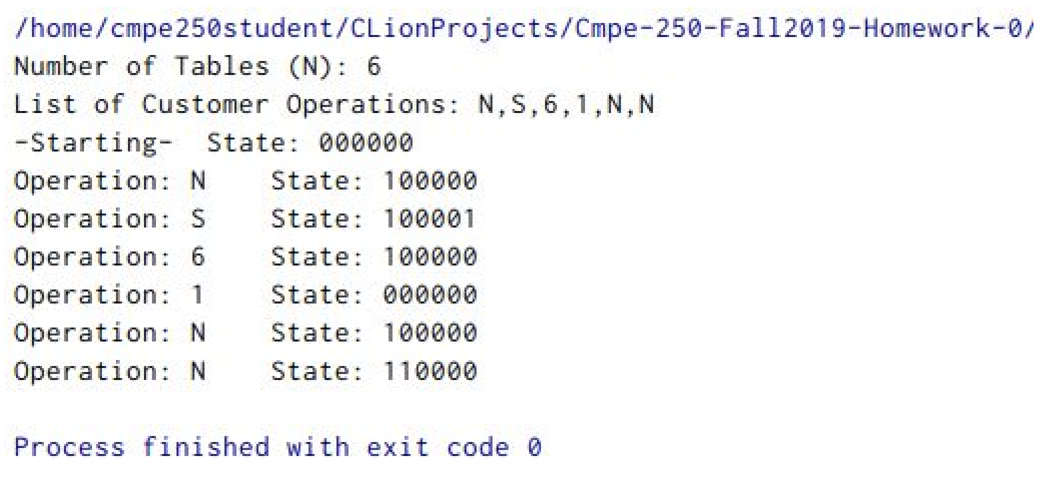
Sample Input and Output Files – 1

Sample Input File (input.txt)
6
N,S,6,1,N,N

- Initially there are 6 empty tables in the CMPE250 Family Restaurant, so the initial table assignment status is 000000.
- N: There is an incoming customer through the north entrance, who will sit to Table 1, since Table 1 is the closest empty table to the north entrance.
    - The current assignment status is 100000.
- S: There is an incoming customer through the south entrance, who will sit to Table 6, since Table 6 is the closest empty table to the south entrance.
    - The current assignment status is 100001.
- 6: Customer who was sitting in Table 6 is leaving, so Table 6 will be empty.
    - The current assignment status is 100000.
- 1: Customer who was sitting in Table 1 is leaving, so Table 1 will be empty.
    - The current assignment status is 000000.
- N: There is an incoming customer through the north entrance, who will sit to Table 1, since Table 1 is the closest empty table to the north entrance.
    - The current assignment status is 100000.
- N: There is an incoming customer through the north entrance, who will sit to Table 2, since Table 2 is the closest empty table to the north entrance.
    - The current assignment status is 110000.
- Since there are no more incoming or leaving customers, the final output is 110000.

Sample Output File (output.txt)
110000

Sample terminal output:



```
/home/cmpe250student/CLionProjects/Cmpe-250-Fall2019-Homework-0/
Number of Tables (N): 6
List of Customer Operations: N,S,6,1,N,N
-Starting-  State: 000000
Operation: N    State: 100000
Operation: S    State: 100001
Operation: 6    State: 100000
Operation: 1    State: 000000
Operation: N    State: 100000
Operation: N    State: 110000

Process finished with exit code 0
```

Sample Input and Output Files – 2

Sample Input File (input2.txt)
12 S,S,S,N,N,11,N,10,S,3,N,N,1,S,S

- Initially there are 12 empty tables in the CMPE250 Family Restaurant, so the initial table assignment status is 000000000000.
- S: There is an incoming customer through the south entrance, who will sit to Table 12, since Table 12 is the closest empty table to the south entrance.
  - The current assignment status is 000000000001.
- S: There is an incoming customer through the south entrance, who will sit to Table 11, since Table 11 is the closest empty table to the south entrance.
  - The current assignment status is 000000000011.
- S: There is an incoming customer through the south entrance, who will sit to Table 10, since Table 10 is the closest empty table to the south entrance.
  - The current assignment status is 000000000111.
- N: There is an incoming customer through the north entrance, who will sit to Table 1, since Table 1 is the closest empty table to the north entrance.
  - The current assignment status is 100000000111.
- N: There is an incoming customer through the north entrance, who will sit to Table 2, since Table 2 is the closest empty table to the north entrance.
  - The current assignment status is 110000000111.
- 11: Customer who was sitting in Table 11 is leaving, so Table 11 will be empty.
  - The current assignment status is 110000000101.
- N: There is an incoming customer through the north entrance, who will sit to Table 3, since Table 3 is the closest empty table to the north entrance.
  - The current assignment status is 111000000101.
- 10: Customer who was sitting in Table 10 is leaving, so Table 10 will be empty.
  - The current assignment status is 111000000001.
- S: There is an incoming customer through the south entrance, who will sit to Table 11, since Table 11 is the closest empty table to the south entrance.
  - The current assignment status is 111000000011.
- 3: Customer who was sitting in Table 3 is leaving, so Table 3 will be empty.
  - The current assignment status is 110000000011.
- N: There is an incoming customer through the north entrance, who will sit to Table 3, since Table 3 is the closest empty table to the north entrance.
  - The current assignment status is 111000000011.
- N: There is an incoming customer through the north entrance, who will sit to Table 4, since Table 4 is the closest empty table to the north entrance.
  - The current assignment status is 111100000011.
- 1: Customer who was sitting in Table 1 is leaving, so Table 1 will be empty.
  - The current assignment status is 011100000011.
- S: There is an incoming customer through the south entrance, who will sit to Table 10, since Table 10 is the closest empty table to the south entrance.
  - The current assignment status is 011100000111.
- S: There is an incoming customer through the south entrance, who will sit to Table 9, since Table 9 is the closest empty table to the south entrance.
  - The current assignment status is 011100001111.
- Since there are no more incoming or leaving customers, the final output is 011100001111.

Sample Output File (output2.txt)
011100001111

Development Platform

Your code should be buildable and runnable in our provided virtual machine. We will test your code in our provided virtual machine, so your code should be built and run on this platform. If your code cannot be built or run in our provided virtual machine, you might get zero.

Details of the Given Code

Table

This class has a boolean field called **occupied** that represents the status (0 for empty or 1 for occupied) of the restaurant table. This field is defined as private, so it is not directly accessible from outside. We provided get method for the access and occupy/empty methods for modification.

Restaurant

This is the class you should be editing. It consists of a list of table objects, **tables**, which represents the tables in a restaurant. It has a constructor which takes n as input argument and then append tables to the list n times. Note that we are using C++ standard library's vector for representing a list.

You will be working on the **execute()** method of the Restaurant class. It takes the list of **operations**, and then updates the tables accordingly. You are given an example code that iterates over the operation vector, then prints the table vector at each round. You only need to add some code to the region marked with the comment saying: /* YOU NEED TO IMPLEMENT THIS PART */

main.cpp

This file handles the reading data from the input file. It also creates a Restaurant object, and then calls its execute() method before writing result to the output file. You can modify this file to observe program execution, but don't forget to use the original version for testing since we will be evaluating your code on the original file.

Submission Policy

You are supposed to use the Git system provided to you for all projects. No other means of submission will be accepted. Also pay attention to the following points:

- All source codes are checked automatically for similarity with other submissions and exercises from previous years. Make sure you write and submit your own code. Plagiarism and any other forms of cheating will have serious consequences, including getting "-100" as the project score and failing the course.
- Make sure you document your code with necessary inline comments, and use meaningful variable names. Do not over-comment, or make your variable names unnecessarily long.
- If you want to make any changes on your project, you should do it before submission system closes. Your changes will not be accepted by e-mail. Connectivity problems to the Internet or to GitHub in the last few minutes are not valid excuses for being unable to submit. You should not risk leaving your submission to the last few minutes. After uploading to GitHub, check to make sure that your project appears there.
- We will test your code with different input combinations, so we recommend you to test your code with different input combinations (written by you) as well.
- Deadline is 15 October 2019, 23:00. Good luck.