Modules

Objective

The objective of this problem is to ensure that students understand the concept of Object-oriented programming (OOP). In this case it's tested with the understanding of objects and encapsulations.

Problem Description

In this problem, you are required to create 3 classes:

Schedule
- day : String - startTime : Integer - endTime : Integer

Module
- code : String - lectureSchedule : Schedule - tutorialSchedule : Schedule - labSchedule : Schedule

Timetable
- listOfModules : List

The valid operations are:

1. **MODULE <CODE> <LECTURE SCHEDULE> <TUTORIAL SCHEDULE> <LAB SCHEDULE>**
   - This operation adds the module with code = <**CODE**> to the student's timetable if all the schedules of this modules (lecture, tutorial and lab schedule) do not clash with the schedules of all the modules in current timetable.
   - Output "**Added**" if the module can be added to the student's timetable.
   - Otherwise output "**Clashed**".

2. **COUNT <DAY>**
   - This operation counts the number of classes/schedules (not module) that the student has on <**DAY**>.
   - Output the number of classes that the student has on <**DAY**>.

*** Condition for no clash between Person1 and Person2 is given as:**
Person2.startTime >= Person1.endTime OR Person1.startTime >= Person2.endTime

Input

The first line of the input contains an integer **K** (1 <= **K** <= 30), denoting the number of operations given. The next **K** lines are the operations.

Output

There are **K** lines in the output. Output in line-i is the result of query-i.

Sample Input

```
10
MODULE CS1020 Wednesday 10 12 Tuesday 9 10 Thursday 10 12
MODULE CS1010 Wednesday 8 10 Tuesday 8 9 Thursday 9 10
MODULE CS2103 Wednesday 8 10 Wednesday 10 12 Wednesday 12 13
MODULE CS2100 Monday 10 12 Wednesday 9 10 Friday 10 12
COUNT Wednesday
MODULE CS1231 Friday 8 10 Friday 12 14 Friday 14 15
MODULE CS2105 Friday 15 16 Tuesday 16 18 Tuesday 15 16
MODULE CS2102 Friday 10 14 Monday 16 18 Thursday 16 18
COUNT Friday
```

COUNT Sunday

Sample Output
Added
Added
Clashed
Clashed
2
Added
Added
Clashed
4
0

Explanation
Query 1: The student can add module CS1020 to his timetable.

Query 2: The student can add module CS1010 to his timetable.

Query 3: CS2103 lecture schedule will clash with CS1010 lecture schedule, hence the student cannot add CS2103 to his timetable.

Query 4: CS2100 tutorial schedule will clash with CS1010 lecture schedule, hence the student cannot add CS2100 to his timetable.

Query 5: There are 2 classes to be attended on Wednesday. They are CS1020 and CS1010 lecture.

Query 6: The student can add module CS1231 to his timetable.

Query 7: The student can add module CS2105 to his timetable.

Query 8: CS2102 lecture schedule will clash with CS1231 tutorial schedule, hence the student cannot add CS2102 to his timetable.

Query 9: There are 4 classes to be attended on Friday. They are CS1231 lecture, CS1231 tutorial, CS1231 lab and CS2105 lecture.

Query 10: There is no class to be attended on Sunday.

Note
To make things simpler, the problem should be solved using OOP.

Algorithm Template
1. How to check whether a schedule clashes with other schedules?

2. How to check whether a module clashes with other modules?

3. What data structure should be used to store all taken modules?

4. How to count the number of schedules attended on a particular day?