**Report**

**Data**

Program inputs data from three files:

1. data.txt – file containing list of courses including information about them
2. group.txt – file containing list of groups
3. rooms.txt – file containing all rooms and types

To store data and easily manipulate with them, we have made three classes named Course, Group and Room. So, courses are stored in a vector of type class Course named courseList. Same for groupList and roomList. More than that, we also have a lot of vectors which makes it easy to solve the needed task.

**Algorithm**

First of all, forget about lectures for a short amount of time, as program makes lecture schedules’ after tutorials/labs and etc. This is made because all groups taking this course should sit in one place.

Now, we need to solve the task for tutorials/labs/complabs. To do so, we a define a term section (or class). So, section/class is typically just an instance of some course. For example, if we have a course Math which has Lab and Tutorial, we can have two sections (classes) of this course. Two classes for same courses differ only by the times when they are held and the rooms where they are held.

So, now our task is give some class of course to a group which is taking this course. Initially, there are many courses with no classes. We do this quite trivial operation: for each group G and the course C, if the group G is taking the course C, we try to find some class of course C for that group (such a class that group G does not have overlaps). If there are no such classes, then we try to add class of the course C.

In the case with lectures, for each course (not class), we try to find a room and time for this course to be held, such that all the groups taking this course can attend that lecture.

Assuming that there are plenty of rooms and 20 slots (5 days a week \* 4 times per day) this algorithm works fine.

**Challenges**

* First of all, course content did not include explanation of vectors and maps. Thus, it took time fully understand and get handle with them. Fortunately, one of us already studied it at high school, hence made learning easier.
* Accuracy was a challenging task, as there were lots of variables with similar names. For example, at once we used classList instead of courseList, which lead to improper outputs.
* Moreover, as we dealt with vectors, it was necessary to set their sizes properly to avoid calling non-existent places.
* Hopefully, we worked in group of four, thus this helped to overcome challenges. To clarify, most of mistakes were based on writing code as the was lots of data, however, at least one of us was able to notice those errors.
* Furthermore, to get perfectly working code and algorithm we tested it lots of time with different inputs.

**Roles**

All of us worked to develop algorithm and test it. Though we divided by roles, only with the help of one another we reached great results.

**Codding:**

Khanzada Kuatbay and Ali Ganiyev worked with getting inputs from .txt files, also with getting proper output.

Aisultan Kassenov and Akzhol Baktiyar worked with finding available slots for groups.