Project

Due Date for Project Submission: Tuesday May 10th, 2016 (till 11:59PM).

Teams: No Groups, Work is individual based (Each Student submit a project).

Notes:

- Automatic Grading will be applied over an online system.
- Use the Submission Manual that explains how to register on the online system and submit, test/evaluate your code.
- The online system can automatically detect copied submissions with intelligent comparison (plagiarism detection). All detected copies of students submissions will take **zero** grade.

Project Description

It is required to develop a program to do operations on courses, exams and students data. The program use a defined string format to represent courses exams and students. All courses exams will be in one line input, and all students will be in one line input. For example:

CoursesExams=[HUM001,Technical Writing, 28/4/2016, HallA;CSE121,Computer Programming,3/5/2016, HallB]
Students=[12001,Ahmed Hassan, (CSE121,HUM001);12002,Aly Hamed,(CSE121)]

Courses Exams Data

The CoursesExams line contains a list of courses exams data separated by ';'. Fields of a course exam are:

1. Course Code (like HUM001)
2. Course Name (like Technical Writing)
3. Exam Date (like 28/4/2016)
4. Exam Location (like HallA)

Data of the same course exam are separated by comma as shown in the above example. User can enter data for up to 100 courses exams in one line separated by ; as shown in the above example.

Students Data

The Students line contains a list of students data separated by ';'. Fields of a students are:

1. Student ID (like 12001)
2. Student Name(like Ahmed Hassan)
3. Course codes of student attended courses (like (CSE121,HUM001)). One student can attend up to 10 courses.

Students data of the same student are separated by comma as shown in the above example. User can enter data for up to 100 students in one line separated by ; as shown in the above example.

Operations

When the program start, the user enters one CoursesExams Line and one Students Line in the defined above format then followed by **one or more** operations from the below table (each operation in a line). The program ends when it reads **Quit** operation.

Level	Operation	Action Required from the Program
1	**Number_Students**	Print the number of students.
	Number_Courses	Print the number of Courses.
	Number_Halls	Print the number of Halls
	Student_ID_Min	Print the minimum student ID value
	Student_ID_Max	Print the maximum student ID value
	Students_Dropped_ID	Print the IDs between minimum and maximum student ID that are not assigned to any student
	Exams_Start_Date	The date of the earliest exam
	Exams_End_Date	The date of the latest exam
	Exams_Period_InDays	The number of days from the start date to the end date. (including the start and end days)
	Quit	End program

2	**Student_Courses** StudentID	List all course codes of the student having the given StudentID **(StudentID will be a valid student ID)** (every output course code should be printed in a separate output line)
	Course_Students CourseCode	List all student IDs of the students attending the given CourseCode **(CourseCode will be a valid Course Code)** (every output student ID should be printed in a separate output line)
	List_Course_Students More n	List course codes of courses having more than n students **(n will be a valid non negative integer)** (every output course code should be printed in a separate output line)
	List_Course_Students Less n	List course codes of courses having less than n students **(n will be a valid non negative integer)** (every output course code should be printed in a separate output line)
	List_Course_Students Equal n	List course codes of courses having exactly n students **(n will be a valid non negative integer)**

		(every output course code should be printed in a separate output line)
	List_Student_Courses More n	List all student IDs of the students attending more than n courses **(n will be a valid non negative integer)**
		(every output student ID should be printed in a separate output line)
	List_Student_Courses Less n	List all student IDs of the students attending less than n courses **(n will be a valid non negative integer)**
		(every output student ID should be printed in a separate output line)
	List_Student_Courses Equal n	List all student IDs of the students attending exactly n courses **(n will be a valid non negative integer)**
		(every output student ID should be printed in a separate output line)
	List_Hall_Students Hallname,Date	List all student IDs of the students attending exam at hallname at examdate **(Hallname will be a valid Hall Name, Date will be a valid date)**

			(every output student ID should be printed in a separate output line) Repeated ID should only be printed one time
	List_Hall_Students_InAnyday HallName		List all student IDs of the students attending exam at hallname at any date **(Hallname will be a valid Hall Name)** (every output student ID should be printed in a separate output line) Repeated ID should only be printed one time
	List_Day_Students_InAnyHall Date		List all student IDs of the students attending exam given examdate **(Date will be a valid date)** (every output student ID should be printed in a separate output line) Repeated ID should only be printed one time
3	**List_Exams** Date		List all course codes of the courses having exam at the given Date **(Date will be a valid date)** (every output course code should be printed in a separate output line)
	List_StudyDays StudendID,CouseCode		StudyDays of is the number of days a student have between

	two consecutive exams he/she attend (not including the exam days) in case of first exam the StudyDays should be 7. This command should print the StudyDays of a student before certain CourseCode.
List_StudyDays_ForEveryStudent CourseCode	This command should print every student ID in a course followed by the StudyDays of that student for the given coursecode.
List_StudyDays_ForEveryCourse StudentID	This command should print every Course Code attended by the given student followed by the StudyDays of that course for the given student.
List_Minimum_Student_StudyDays StudentID	This command should print the minimum StudyDays of all courses of the given student.
List_Minimum_Course_StudyDays CourseCode	This command should print the minimum StudyDays of all students of the given course.
List_Minimum_StudyDays	This command should print the minimum StudyDays of all students of all courses.
List_Exam_InSameDay StudentID	Print all course codes of courses attended by the given student that have exams on the same day (conflict)
List_Exam_InSameDay_ForEveryStudent	Print all student IDs of all students followed by course codes of courses attended by the student that have exams on the same day (conflict)

	List_Exam_InSameHall Date		Print all course codes of courses have exams on the same hall at the given date (conflict)
	List_Exam_InSameHall_Simultaneously _InAnyDay		Print all dates from the start exam date to the end exam date and each date is followed by course codes of courses have exams on the same hall at that date (conflict)
Extra Marks	**Reschedule_Course_InSameHall_InDiff erentDay** Code,StartDate,EndDate		Choose another date for the course with Course Code (Code) and keeping the same hall without having a conflict with other exams. The chosen date should be in the given date range.
	Reschedule_Course_InSameDay_InDiff erentHall Code		Choose another hall for the course with Course Code (Code) and keeping the same date without having a conflict with other exams.
	Reschedule_Course_InSameHall_To_Re ach_Minimum_StudyDays Code,MinDays,StartDate,EndDate		Choose another date for the course with Course Code (Code) and keeping the same hall without having a conflict with other exams. And such that the StudyDays of all students of that course should be at least (MinDays). The chosen date should be in the given date range.

Project Marks Distribution:

40% to support operations in Level 1

30% to support operations in Level 2

30% to support operations in Level 3

20% for Extra Marks Part

General Constraints:

1- User can input up to 100 courses and up to 100 students.
2- Each student can attend up to 10 courses.
3- Graphics library and Time class can not be used in that project.
4- All your code should be in one file.
5- Do not prompt user to enter anything , just read the input directly and print the output of operations directly.
6- At any operation if the output is empty the operation should print "none"
7- Output should not include any extra white spaces or any extra texts more than the results.

Hints:

1. You should write `int main()` and you should not write `void main()`
2. Never use `system("pause");`
3. While parsing data, you will need to convert from string to a number.
   Use atof() function defined in <cstdlib>.
   The conversion code is f=atof(s.c_str());
   Use the following example as a guide:

```cpp
#include<iostream>
#include<string>
#include<cstdlib>

using namespace std;

int main() {
// To convert from string to float use atof function
// atof needs #include<cstdlib>

string s = "1.5";
float f;
f = atof(s.c_str());


return 0;
}
```

Input / Output Samples

Assuming the user entered:

CoursesExams = [HUM001,Technical Writing, 28/4/2016, HallA;

          CSE121,Computer Programming, 3/5/2016, HallA;

          PHM110,Math 3, 5/5/2016, HallC;

          PHM103,Physics, 5/5/2016, HallA;

          EPM120,Electric Circuits, 6/5/2016, HallD;

          ECE132,Electronics, 7/5/2016, HallB]

Students= [12001,Ahmed Hassan, (CSE121,HUM001,PHM110);

          12002,Aly Hamed, (CSE121,PHM110,PHM103);

          12003,Ayman Ali, (PHM110,EPM120,ECE132);

          12006,Hazem Elshenawy, (PHM103,ECE132)]

In the following, we show samples of the program interaction with commands. (Just for Clarification, user input is in Blue and program output in red and this not required in the program)

Number_Students

4

Number_Courses

6

Number_Halls
4

Student_ID_Min
12001

Student_ID_Max
12006

Students_Dropped_IDs
12004
12005

Exams_Start_Date
28/4/2016

Exams_End_Date
7/5/2016

Exams_Period_InDays

10

Quit

Thanks!

Note: Quit should end reading any additional operations and end your program.

Student_Courses 12003

PHM110

EPM120

ECE132

Course_Students CSE121

12001

12002

List_Course_Students More 1

CSE121

PHM103

PHM110

ECE132

List_Course_Students Less 2

HUM001

EPM120

List_Course_Students Equal 3

PHM110

List_Student_Courses More 2

12001

12002

12003

List_Student_Courses Less 3

12006

List_Student_Courses Equal 2

12006

List_Hall_Students HallA, 5/5/2016

12002

12006

List_Hall_Students_InAnyDay HallA

12001

12002

12006

List_Hall_Students_InAnyHall 5/5/2016

12001

12002

12003

12006

List_Exams 5/5/2016

PHM110

PHM103

List_StudyDays 12006,ECE132

2

List_StudyDays_ForEveryStudent ECE132

12003 1

12006 2

List_StudyDays_ForEveryCourse 12006

PHM103 14

ECE132 2

List_Minimum_Student_StudyDays 12006

ECE132 2

List_Minimum_Course_StudyDays ECE132

1

List_Minimum_StudyDays

1

List_Exam_InSameDay 12002

PHM110

PHM103

List_Exam_InSameDay_ForEveryStudent

12001 none

12002 PHM110 PHM103

12003 none

12006 none

List_Exam_InSameHall 28/4/2016

HUM001

CSE121

List_Exam_InSameHall_Simultaneously_InAnyDay

28/4/2016 HUM001 CSE121

3/5/2016 none

5/5/2016 none

6/5/2016 none

7/5/2016 none

Reschedule_Course_InSameHall_InDifferentDay CSE121, 7/5/2016, 10/5/2016

8/5/2016

Reschedule_Course_InSameDay_InDifferentHall PHM110

HallB

Reschedule_Course_InSameHall_To_Reach_Minimum_StudyDays

ECE132, 2, 7/5/2016, 10/5/2016

8/5/2016