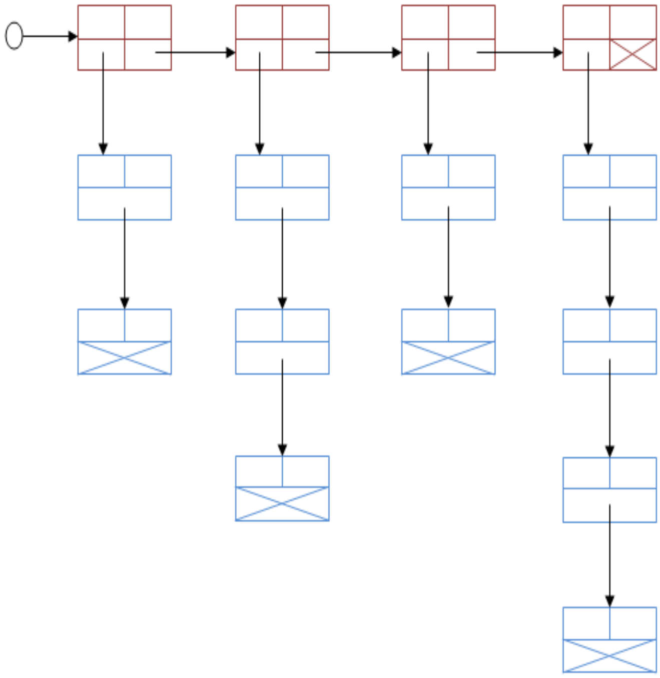
Ankara University
Computer Engineering Department
COM2067/COM267 LAB3
Deadline: 06.11.2022 23:59

As part of this lab study, we have given you a main.c and function.h file. The main.c file will contain only the int main() function, which is your main function, while the function.h file will contain the definitions of all the other functions you will use in this application. It is expected of you to fill in the function.h file so that the main.c file will run without errors to produce the expected output. First of all, carefully examine the main.c file. Here, we have placed function calls related to getting values from the user. You can review the SimpleTutorial.pdf document for the definition of functions in a separate file.

When you complete the contents of the function.h file, you will get a C program that keeps the information of the courses given by a lecturer and the list of students who took this course. This lecturer teaches 4 different classes. The number of students in each class may differ from each other. There are two different linked list structures for holding courses and students. The node structures of these lists are given below. In the structure of the class (nodeClass), information about which class it is (classID) and the average of the midterm exams of the students in the class (classMidtermAverage) are kept. Besides these, there is a pointer pointing to the next class and a pointer pointing to a node belonging to the structure (nodeStudent) used to define students' information. In the structure called nodeStudent, there is a student's id, midterm grade and a pointer showing the next student in the same class. The relationship between these two structures is given in the figure below. The red nodes are the nodes of the nodeClass structure, the blue nodes are the nodes of the nodeStudent structure.



The program takes the student number and midterm grade from the user as input. Students whose student number starts with 66 are in the 1st class, with 77 in the 2nd class, students starting with 88 are in the 3rd class, and students starting with 99 are in the 4th class. Students must appear in the linked list sequentially. Sorting will be done in descending order according to the midterm grade. If the notes are the same, the lower number should be first in the list. The sorted list should be preserved by adding it to the correct location while adding. After all students are added to the list, the midterm average of each class will be calculated and kept in the classMidtermAverage variable of the node generated from the nodeClass structure of the relevant class.

In printAll function, your program will display the id of each class, the midterm average, the ids of the students in the relevant class and the midterm grades sequentially, in accordance with the format in the output file given to you.

```
struct nodeClass
{
        int classID;
        double classMidtermAverage;
        struct nodeClass *next;
        struct nodeStudent *studentPtr;
};
```

```
struct nodeStudent
{
        int studentID;
        int midterm;
        struct nodeStudent *next;
};
```

Example Input (studentId midterm)

```
99215  75
66123  45
66127  50
99321  90
88234  90
88313  45
77245  65
77248  70
99218  70
99219  80
77445  75
-1
```

Example Output (classId classMidtermAverage)

```
1      47.50
66127  50
66123  45
2      70.00
77445  75
77248  70
77245  65
3      67.50
88234  90
88313  45
4      78.75
99321  90
99219  80
99215  75
99218  70
```