A lecturer teaches 4 different classes of Data Structures. The number of students in these classes may be different. Using the node structures given below, perform the given assignment. Be careful to leave student list as sorted when performing the insert operation. Sort will be performed in decreasing order based on the midterm scores. If the grades are the same, the student with smaller studentID must appear before in the list. After the insert operation is complete, the midterm average of each class will be calculated. The computed midterm average will be written to classMidtermAverage in the class list. Then the id of each class and the midterm average will be printed on the screen. Write a print method that prints all the structure on the screen to show that the lists are created correctly -**void printAll(nodeClass *head)**. Students who begin with StudentID 66 are in class **1**, students starting with 77 in class **2**, students starting with 88 are in class **3**, and students starting with 99 are in class **4**.

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
```

```
struct nodeClass //Red nodes in the list
{
        int classID;
        double classMidtermAverage;
        struct nodeClass *next;      struct
nodeStudent *studentPtr;
};

struct nodeStudent //Blue nodes in the list
{
        int studentID;          int midterm;  struct
nodeStudent *next;
};
```