LAB REPORT: LAB 1

Task 1: Printing the student ID having highest GPA.

Working code:

```c
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

int main()
{
    FILE *fp;
    char s[10000];
    char *info[3];

    fp = fopen("E://files//grades.txt", "r");

    if(fp == NULL)
        printf("File could not be opened!");
    else
    {
        int id;
        float gpa;
        int sem;

        float max = 0;
        int mxid;
        while(fgets(s, 10000, fp))
        {
            char *token = strtok(s, ";");

            int i=0;
            while(token != NULL)
            {
                info[i++] = token;
                token = strtok(NULL, ";");
            }

            id = atoi(info[0]);
            gpa = atof(info[1]);
            sem = atoi(info[2]);

            if(gpa > max)
            {
                max = gpa;
                mxid = id;
            }
        }
        printf("%d\n", mxid);
    }
    return 0;
}
```

Explanation: The code requires operation on file. To open a file, we need a file pointer referring to the location of the file. Then we will check if the pointer is null or not. If the pointer is null, then the file couldn't be opened, and we will have to check for errors. I took a character array and an array of pointers. Then I used while loop to traverse through line by line in the file. On each iteration array s contains a line from "grade.txt" file which is ordered like, student id;gpa;sem. Then I used another character pointer and **strtok()** function to split up the line based on semi-colon. Then I traversed through that pointer and stored single words in pointer array such as, info[0] = id, info[1] = gpa, info[2] = sem. Then I converted gpa to float and on each iteration checked if it was the max gpa, if it was max then the id was stored. Then at the end of the iteration I printed the id.

Problems: No problems occurred during solving this task.

Task 2 : Appending new information in "grades.txt" if the information is valid.

Working Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

int main()
{
    FILE *fp, *fp1;
    char s[10000];
    char s1[10000];
    char *info[3];
    char *info1[5];
    char id[20], gpa[20], sem[20];
    scanf("%s%s%s", &id, &gpa, &sem);

    fp = fopen("E://files//grades.txt", "r");
    fp1 = fopen("E://files//studentInfo.txt", "r");


        int flg = 0;
        int flg1 = 0;

        while(fgets(s1, 10000, fp1))
        {
            char *token = strtok(s1, ";");

            int i=0;
```

```
27              while(token != NULL)
28              {
29                  info1[i++] = token;
30                  token = strtok(NULL, ";");
31              }
32
33              if(strcmp(info1[0], id) == 0)
34                  flg1 = 1;
35          }
36
37          if(flg1 == 1)
38          {
39              while(fgets(s, 10000, fp))
40              {
41                  char *token = strtok(s, ";");
42
43                  int i=0;
44                  while(token != NULL)
45                  {
46                      info[i++] = token;
47                      token = strtok(NULL, ";");
48                  }
49
50                  if(atof(gpa) < 2.50 || atof(gpa) > 4.00)
51                  {
52                      flg = 1;
53                      printf("CGPA not in limit!");
54                      break;
55                  }
56                  if(atoi(info[2]) == atoi(sem))
57                  {
58                      flg = 1;
59                      printf("Semester already exists!");
60                      break;
61                  }
62              }
63          }
64          else
65          {
66              flg = 1;
67              printf("Student doesn't exist!");
68          }
69
70          fclose(fp);
71          fclose(fp1);
72          if(flg == 0)
73          {
74              fp = fopen("E://files//grades.txt", "a");
75
76              if(fp == NULL)
77                  printf("File not found");
78              else
```

```
79              {
80                  fprintf(fp, "%s,%s,%s\n", id, gpa, sem);
81                  fclose(fp);
82              }
83          }
84
85      return 0;
86  }
```

Explanation: The starting is same as task 1. Here, instead of one file pointer we need two, one referring to “grades.txt” and another to “studentInfo.txt”. The code takes id, gpa, semester as input. We need to check every input and verify them. First, we need to check the student id if it is in studentInfo. So, at first, I traversed through the “studentInfo.txt” file and checked if id matches. If the id matches, then a flag variable is set to one. Now for the next phase, we check if the flag value is one or zero, if zero then print “student doesn’t exist”, else we traverse through the “grade.txt” file again and check semester, if there is a same semester already, then the code will print “semester already exists” and break the loop setting another flag value to one. Now we check for gpa. If the gpa is less than 2.50 or greater than 4.00, then the input is invalid. For this case, the code will print “gpa not in limit” and then break setting flag value to one. Then, after the iteration is over, we check for that flag value, if the value is one then we have invalid information and we don’t write to the file. On the other hand, if the value is zero we write to the file.

Problems: Checking the conditions was the tricky part, and setting the flags also. I needed to convert the info to other data types and then compare them.

Task 3: Take student id as input and show his/her name and average gpa as output.

Working Code:

```c
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

int main()
{
    FILE *fp, *fp1;
    char s[10000];
    char s1[10000];
    char *info[3], *info1[5];

    fp = fopen("E://files//grades.txt", "r");

    if(fp == NULL)
        printf("File could not be opened!");
    else
    {
        int id;
        float gpa;
        int sem;
        char name[10];

        int idinpt;
        scanf("%d", &idinpt);

        int f = 0;
        float sum = 0;
        int cnt = 0;

        fp1 = fopen("E://files//studentInfo.txt", "r");


        while(fgets(s1, 10000, fp1))
        {
            char *token = strtok(s1, ";");

            int i=0;
            while(token != NULL)
            {
                info1[i++] = token;
                token = strtok(NULL, ";");
            }

            if(atoi(info1[0]) == idinpt)
            {
                f = 1;
```

```
47                  strcpy(name, info1[1]);
48              }
49          }
50          while(fgets(s, 10000, fp))
51          {
52              char *token = strtok(s, ";");
53
54              int i=0;
55              while(token != NULL)
56              {
57                  info[i++] = token;
58                  token = strtok(NULL, ";");
59              }
60
61              id = atoi(info[0]);
62              gpa = atof(info[1]);
63              sem = atoi(info[2]);
64
65              if(id == idinpt)
66              {
67                  sum += gpa;
68                  cnt++;
69              }
70          }
71          if(f == 0)
72          {
73              printf("No student found!");
74          }
75          else
76          {
77              printf("%s %f", name, sum/cnt);
78          }
79      }
80      return 0;
81  }
```

Explanation: This code is also similar to task 2. Here, we have to take student id as input. Then, traversing through the “studentInfo.txt” we check if the student id exists or not. If id exists we store the name in a character array. Then traversing through the “grades.txt” file we again check the id. If match is found, we add the gpa and count the number. Then, after the iteration, if the id exists then we print the name and sum of gpa/total number, otherwise we print “No student found”.

Problems: No problem occurred in this task too.