Report

BY : Eng/Mohamed Badr

Frist term project 2

Student Management System

This program is simple software for student

information management system which can perform the

following operations:

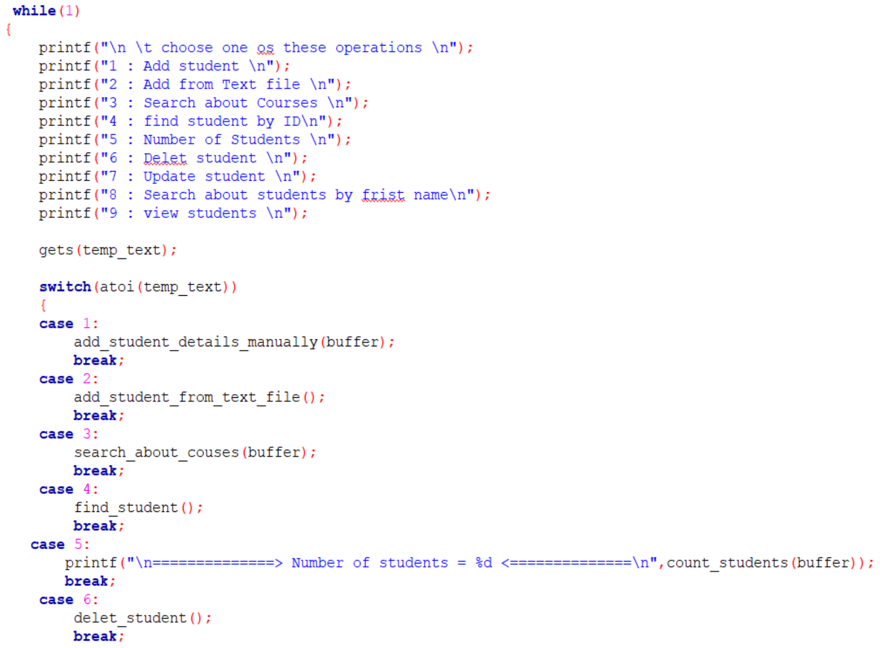
1- Store the first name of the student.

2- Store the last name of the student.

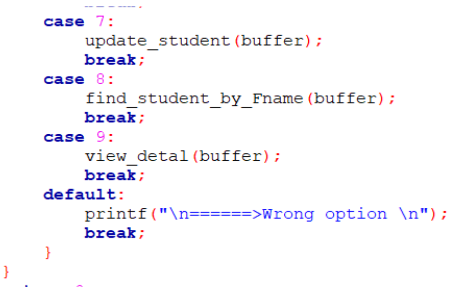
3- Store the unique Roll number for every student.

4- Store the GPA of every student.

5- Store the courses registered by the student.



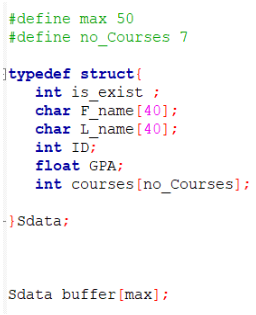
```
 while(1)
{
    printf("\n \t choose one os these operations \n");
    printf("1 : Add student \n");
    printf("2 : Add from Text file \n");
    printf("3 : Search about Courses \n");
    printf("4 : find student by ID\n");
    printf("5 : Number of Students \n");
    printf("6 : Delet student \n");
    printf("7 : Update student \n");
    printf("8 : Search about students by frist name\n");
    printf("9 : view students \n");

    gets(temp_text);

    switch(atoi(temp_text))
    {
    case 1:
        add_student_details_manually(buffer);
        break;
    case 2:
        add_student_from_text_file();
        break;
    case 3:
        search_about_couses(buffer);
        break;
    case 4:
        find_student();
        break;
   case 5:
       printf("\n==============> Number of students = %d <==============\n",count_students(buffer));
       break;
    case 6:
        delet_student();
        break;
```



```
    case 7:
        update_student(buffer);
        break;
    case 8:
        find_student_by_Fname(buffer);
        break;
    case 9:
        view_detal(buffer);
        break;
    default:
        printf("\n======>Wrong option \n");
        break;
    }
}
```

We make switch cases to give options to the user to choose which operation he want

We Identified number of courses is 7 and number of students is 50



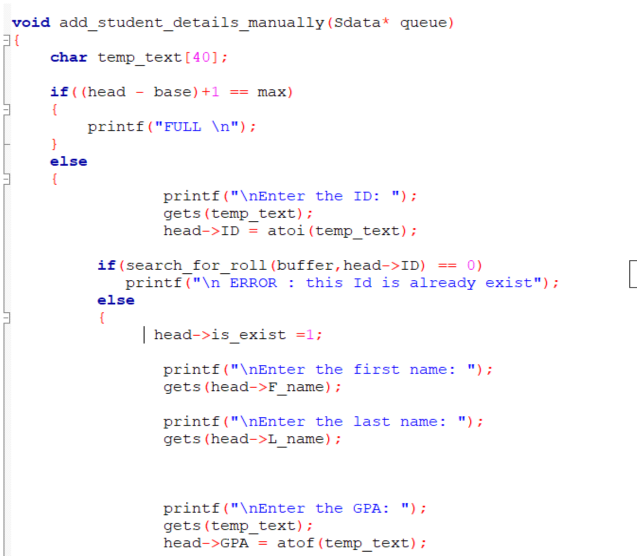
```
#define max 50
#define no_Courses 7

typedef struct{
   int is_exist ;
   char F_name[40];
   char L_name[40];
   int ID;
   float GPA;
   int courses[no_Courses];

}Sdata;


Sdata buffer[max];
```

Implementation: -

1) First approach is to take information manually from users

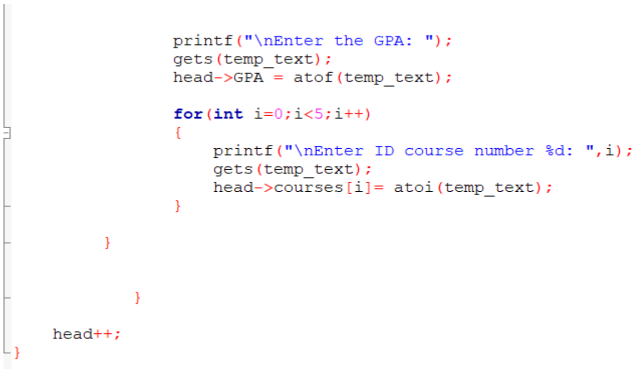
In beginning we take ID and check if it is already exist or not

If not the user complete the information



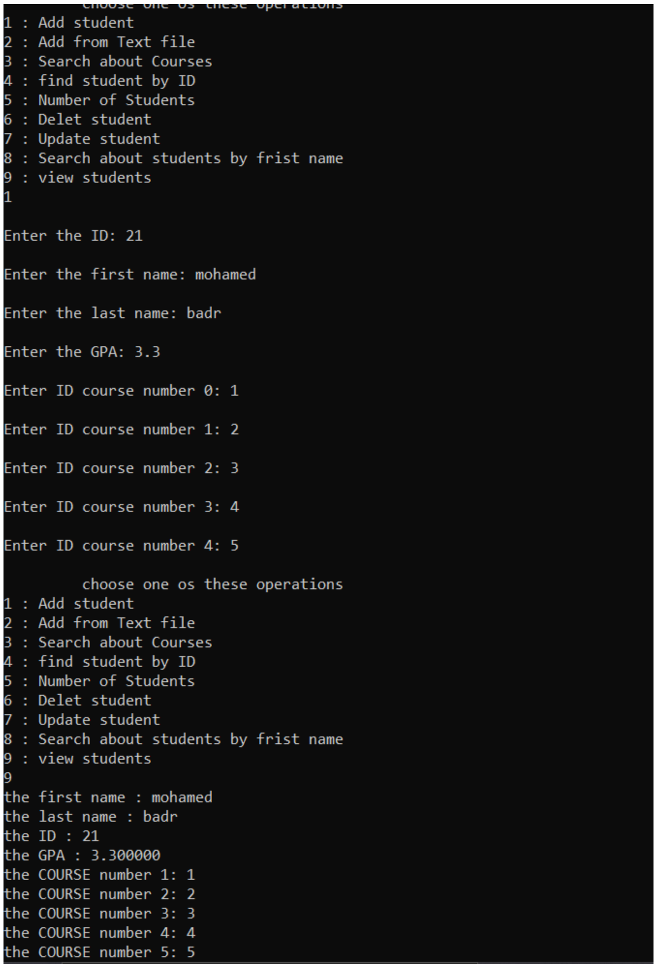
```
void add_student_details_manually(Sdata* queue)
{
    char temp_text[40];

    if((head - base)+1 == max)
    {
        printf("FULL \n");
    }
    else
    {
                printf("\nEnter the ID: ");
                gets(temp_text);
                head->ID = atoi(temp_text);

         if(search_for_roll(buffer,head->ID) == 0)
            printf("\n ERROR : this Id is already exist");
         else
         {
               head->is_exist =1;

                printf("\nEnter the first name: ");
                gets(head->F_name);

                printf("\nEnter the last name: ");
                gets(head->L_name);


                printf("\nEnter the GPA: ");
                gets(temp_text);
                head->GPA = atof(temp_text);
```

Office Online Frame



```
                 printf("\nEnter the GPA: ");
                 gets(temp_text);
                 head->GPA = atof(temp_text);

                 for(int i=0;i<5;i++)
                 {
                     printf("\nEnter ID course number %d: ",i);
                     gets(temp_text);
                     head->courses[i]= atoi(temp_text);
                 }

          }


             }

    head++;
}
```

Test function =>

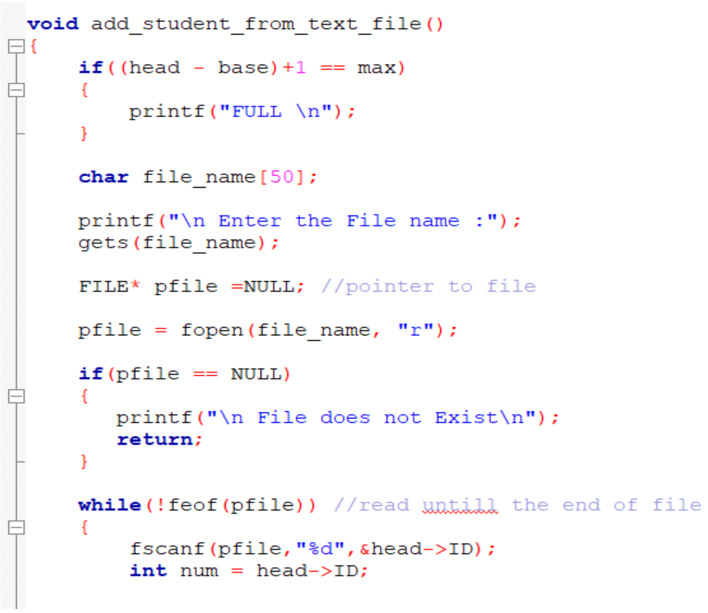


```
         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
1

Enter the ID: 21

Enter the first name: mohamed

Enter the last name: badr

Enter the GPA: 3.3

Enter ID course number 0: 1

Enter ID course number 1: 2

Enter ID course number 2: 3

Enter ID course number 3: 4

Enter ID course number 4: 5

         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
9
the first name : mohamed
the last name : badr
the ID : 21
the GPA : 3.300000
the COURSE number 1: 1
the COURSE number 2: 2
the COURSE number 3: 3
the COURSE number 4: 4
the COURSE number 5: 5
```

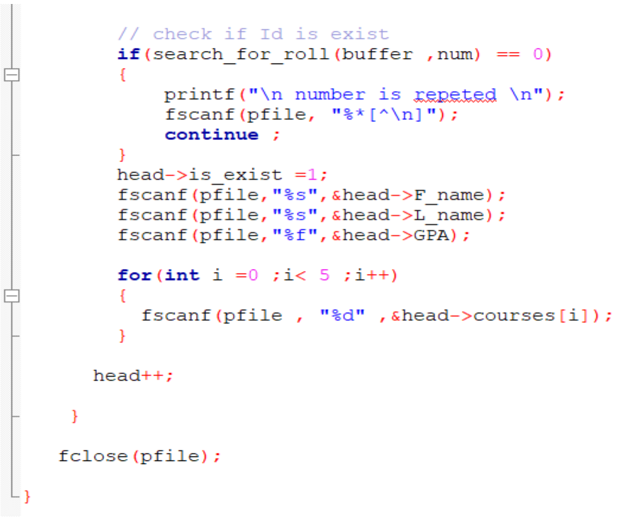
2) Take information from text file

Ask user to enter file name , read data from it and store it in buffer

Check first if any Id is already exist



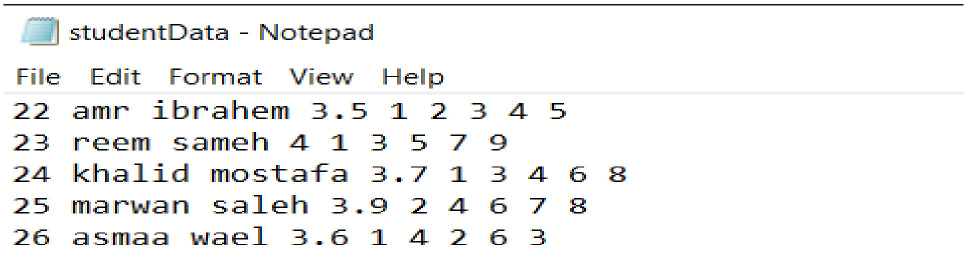
```
void add_student_from_text_file()
{
    if((head - base)+1 == max)
    {
        printf("FULL \n");
    }

    char file_name[50];

    printf("\n Enter the File name :");
    gets(file_name);

    FILE* pfile =NULL; //pointer to file

    pfile = fopen(file_name, "r");

    if(pfile == NULL)
    {
       printf("\n File does not Exist\n");
       return;
    }

    while(!feof(pfile)) //read untill the end of file
    {
        fscanf(pfile,"%d",&head->ID);
        int num = head->ID;
```



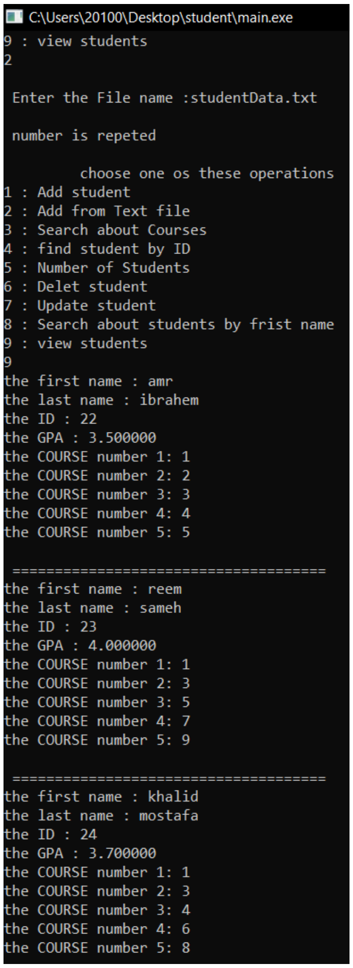
```
         // check if Id is exist
         if(search_for_roll(buffer ,num) == 0)
         {
             printf("\n number is repeted \n");
             fscanf(pfile, "%*[^\n]");
             continue ;
         }
         head->is_exist =1;
         fscanf(pfile,"%s",&head->F_name);
         fscanf(pfile,"%s",&head->L_name);
         fscanf(pfile,"%f",&head->GPA);

         for(int i =0 ;i< 5 ;i++)
         {
           fscanf(pfile , "%d" ,&head->courses[i]);
         }

       head++;

     }

   fclose(pfile);

}
```

Testing ==>

This data from text file



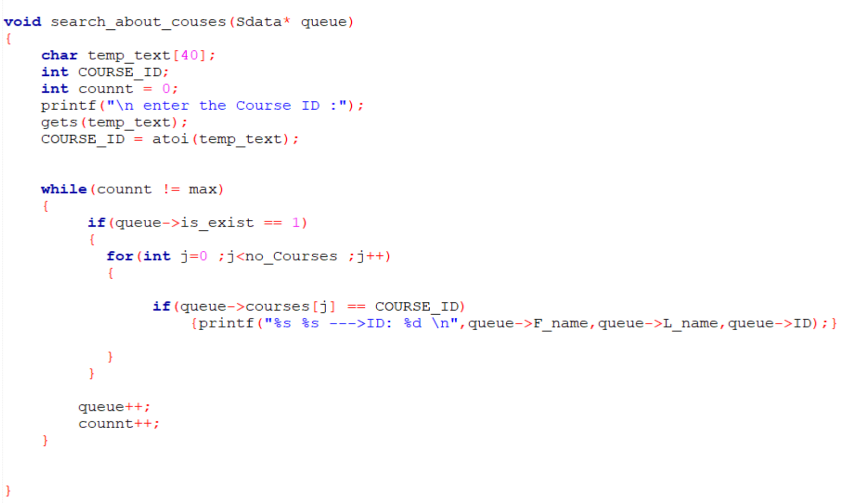
```
22 amr ibrahem 3.5 1 2 3 4 5
23 reem sameh 4 1 3 5 7 9
24 khalid mostafa 3.7 1 3 4 6 8
25 marwan saleh 3.9 2 4 6 7 8
26 asmaa wael 3.6 1 4 2 6 3
```



```
9 : view students
2

 Enter the File name :studentData.txt

 number is repeted

         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
9
the first name : amr
the last name : ibrahem
the ID : 22
the GPA : 3.500000
the COURSE number 1: 1
the COURSE number 2: 2
the COURSE number 3: 3
the COURSE number 4: 4
the COURSE number 5: 5

 =====================================
the first name : reem
the last name : sameh
the ID : 23
the GPA : 4.000000
the COURSE number 1: 1
the COURSE number 2: 3
the COURSE number 3: 5
the COURSE number 4: 7
the COURSE number 5: 9

 =====================================
the first name : khalid
the last name : mostafa
the ID : 24
the GPA : 3.700000
the COURSE number 1: 1
the COURSE number 2: 3
the COURSE number 3: 4
the COURSE number 4: 6
the COURSE number 5: 8
```

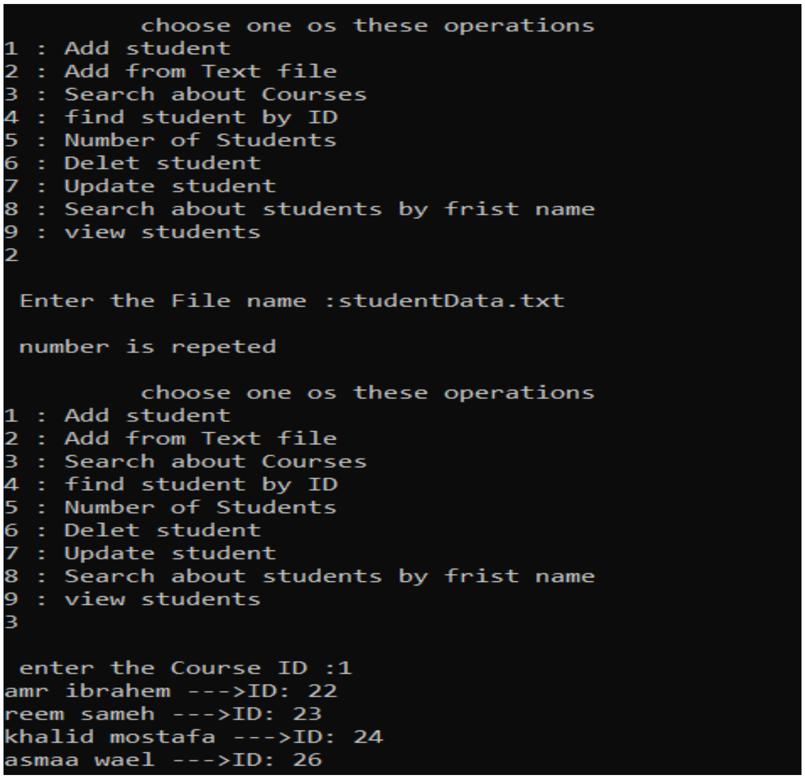
3)know who is taken a specific course

Ask user to enter course ID and search about name and ID of students of this course



```
void search_about_couses(Sdata* queue)
{
    char temp_text[40];
    int COURSE_ID;
    int counnt = 0;
    printf("\n enter the Course ID :");
    gets(temp_text);
    COURSE_ID = atoi(temp_text);


    while(counnt != max)
    {
          if(queue->is_exist == 1)
          {
            for(int j=0 ;j<no_Courses ;j++)
            {

                 if(queue->courses[j] == COURSE_ID)
                     {printf("%s %s --->ID: %d \n",queue->F_name,queue->L_name,queue->ID);}

            }
          }

        queue++;
        counnt++;
    }


}
```

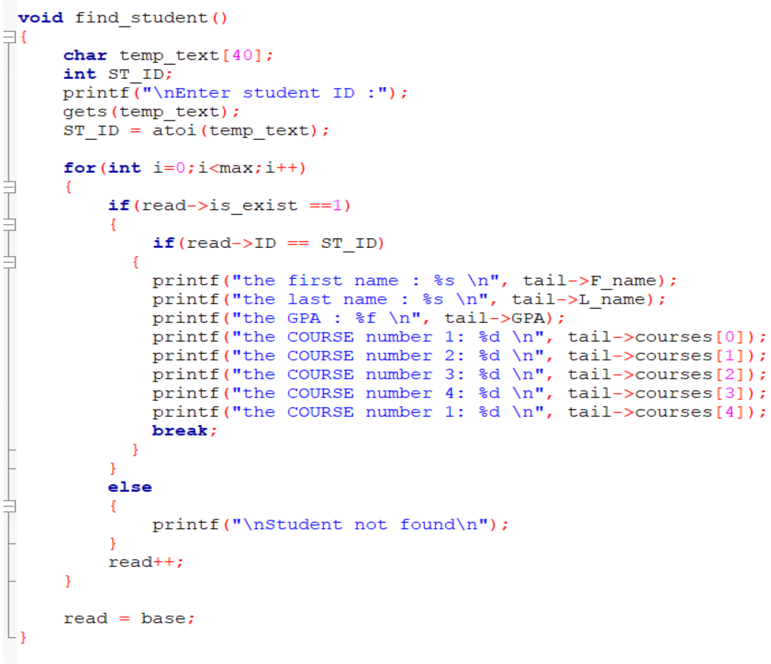
Testing ==>



```
         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
2

 Enter the File name :studentData.txt

 number is repeted

         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
3

 enter the Course ID :1
amr ibrahem --->ID: 22
reem sameh --->ID: 23
khalid mostafa --->ID: 24
asmaa wael --->ID: 26
```

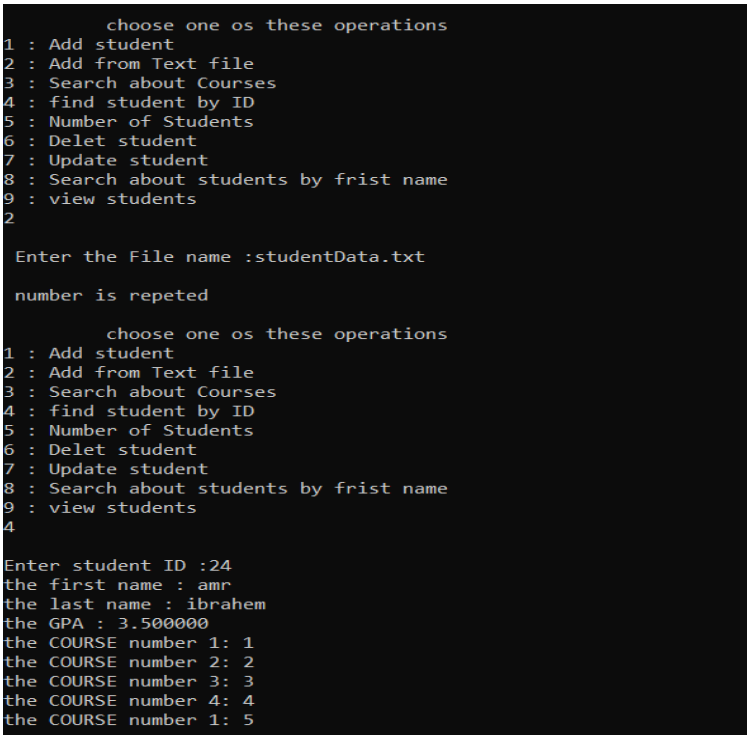
4)search for student using ID

Ask user to enter student's ID and loop until find it if it exist show his information



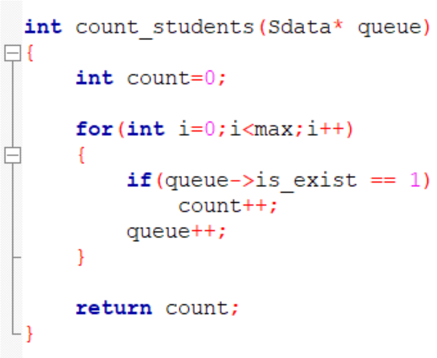
```
void find_student()
{
    char temp_text[40];
    int ST_ID;
    printf("\nEnter student ID :");
    gets(temp_text);
    ST_ID = atoi(temp_text);

    for(int i=0;i<max;i++)
    {
        if(read->is_exist ==1)
        {
            if(read->ID == ST_ID)
          {
            printf("the first name : %s \n", tail->F_name);
            printf("the last name : %s \n", tail->L_name);
            printf("the GPA : %f \n", tail->GPA);
            printf("the COURSE number 1: %d \n", tail->courses[0]);
            printf("the COURSE number 2: %d \n", tail->courses[1]);
            printf("the COURSE number 3: %d \n", tail->courses[2]);
            printf("the COURSE number 4: %d \n", tail->courses[3]);
            printf("the COURSE number 1: %d \n", tail->courses[4]);
            break;
          }
        }
        else
        {
            printf("\nStudent not found\n");
        }
        read++;
    }

    read = base;
}
```

Testing==>



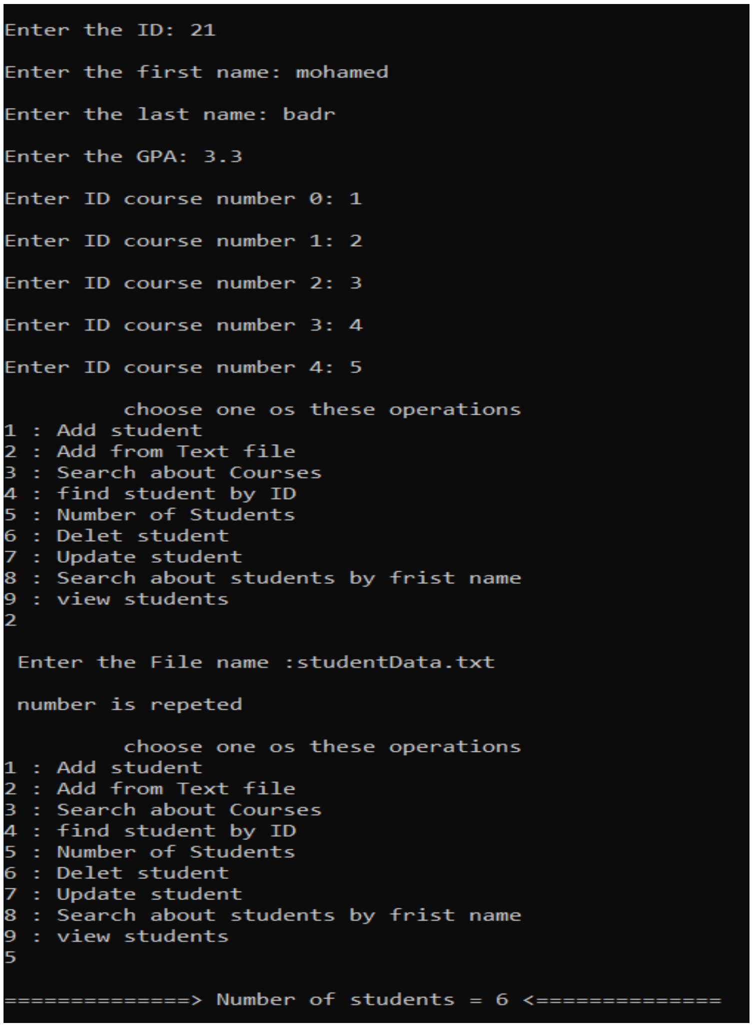
```
         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
2

 Enter the File name :studentData.txt

 number is repeted

         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
4

Enter student ID :24
the first name : amr
the last name : ibrahem
the GPA : 3.500000
the COURSE number 1: 1
the COURSE number 2: 2
the COURSE number 3: 3
the COURSE number 4: 4
the COURSE number 1: 5
```

5)count number of student in system



```
int count_students(Sdata* queue)
{
    int count=0;

    for(int i=0;i<max;i++)
    {
        if(queue->is_exist == 1)
            count++;
        queue++;
    }

    return count;
}
```

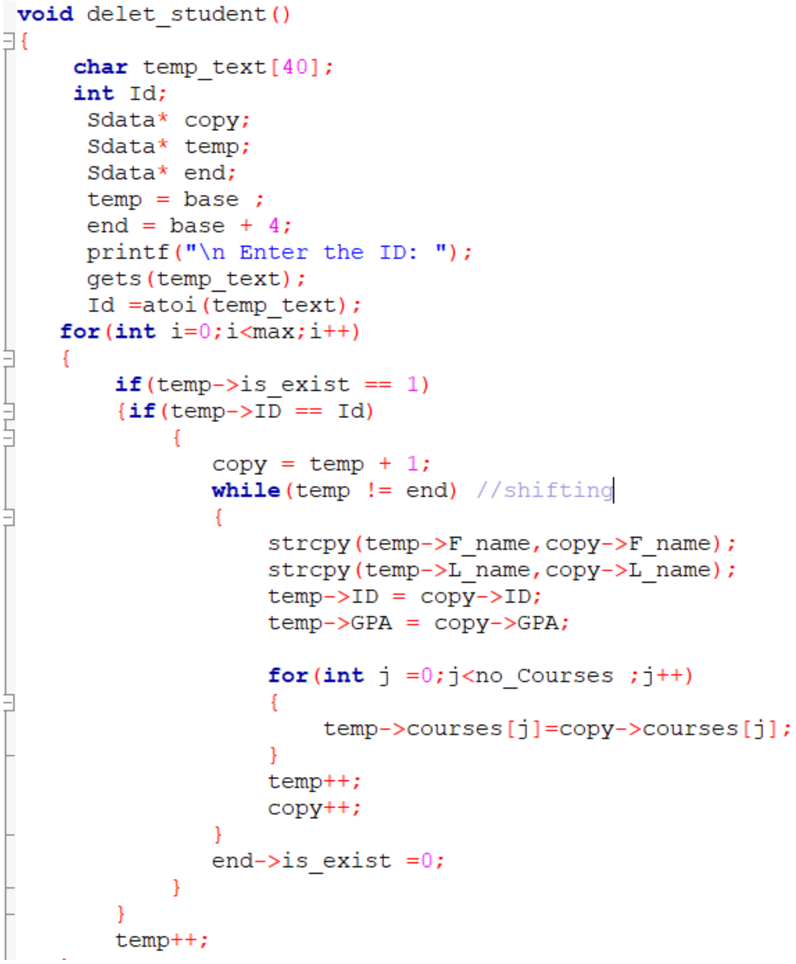
Testing==>



```
Enter the ID: 21

Enter the first name: mohamed

Enter the last name: badr

Enter the GPA: 3.3

Enter ID course number 0: 1

Enter ID course number 1: 2

Enter ID course number 2: 3

Enter ID course number 3: 4

Enter ID course number 4: 5

         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
2

 Enter the File name :studentData.txt

 number is repeted

         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
5

===============> Number of students = 6 <===============
```

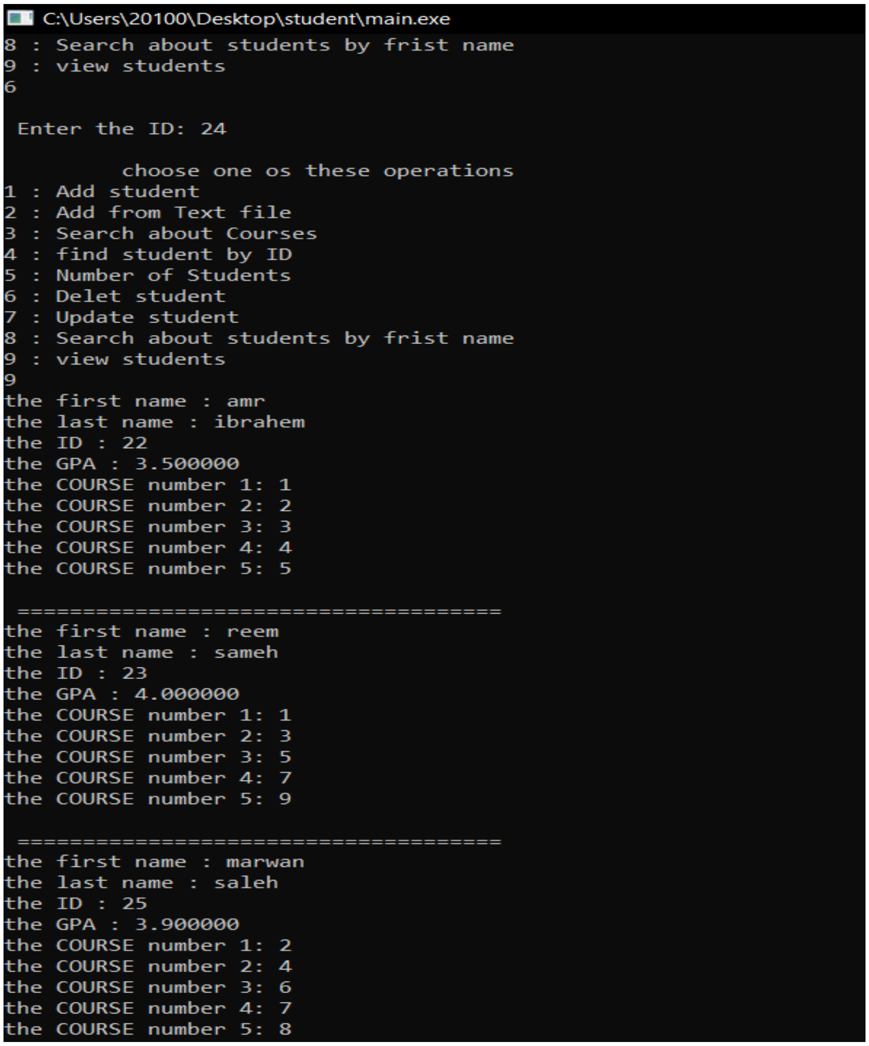
6)delete student

Ask user for student’s ID search for it and delete his data then shift student to make an empty   place in end of buffer



```
void delet_student()
{
    char temp_text[40];
    int Id;
     Sdata* copy;
     Sdata* temp;
     Sdata* end;
     temp = base ;
     end = base + 4;
     printf("\n Enter the ID: ");
     gets(temp_text);
     Id =atoi(temp_text);
   for(int i=0;i<max;i++)
   {
       if(temp->is_exist == 1)
       {if(temp->ID == Id)
           {
              copy = temp + 1;
              while(temp != end) //shifting
              {
                  strcpy(temp->F_name,copy->F_name);
                  strcpy(temp->L_name,copy->L_name);
                  temp->ID = copy->ID;
                  temp->GPA = copy->GPA;

                  for(int j =0;j<no_Courses ;j++)
                  {
                      temp->courses[j]=copy->courses[j];
                  }
                  temp++;
                  copy++;
              }
              end->is_exist =0;
           }
       }
       temp++;
```

Testing==>

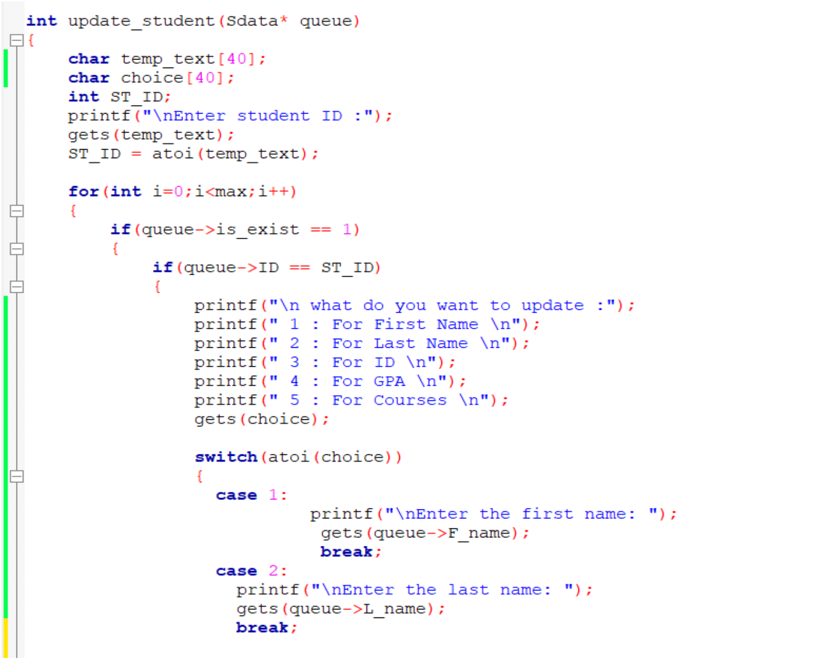


C:\Users\20100\Desktop\student\main.exe

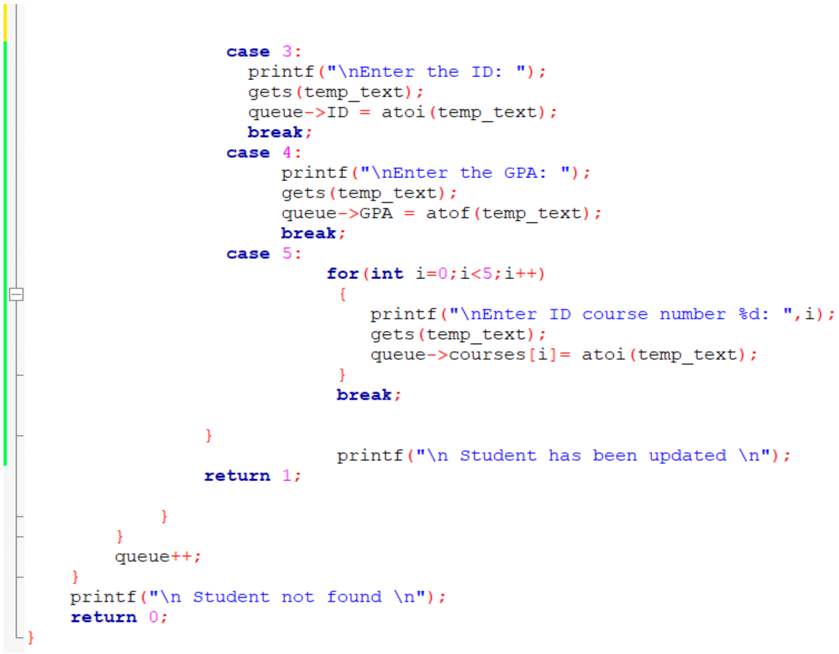
```
8 : Search about students by frist name
9 : view students
6

 Enter the ID: 24

         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
9
the first name : amr
the last name : ibrahem
the ID : 22
the GPA : 3.500000
the COURSE number 1: 1
the COURSE number 2: 2
the COURSE number 3: 3
the COURSE number 4: 4
the COURSE number 5: 5

 =====================================
the first name : reem
the last name : sameh
the ID : 23
the GPA : 4.000000
the COURSE number 1: 1
the COURSE number 2: 3
the COURSE number 3: 5
the COURSE number 4: 7
the COURSE number 5: 9

 =====================================
the first name : marwan
the last name : saleh
the ID : 25
the GPA : 3.900000
the COURSE number 1: 2
the COURSE number 2: 4
the COURSE number 3: 6
the COURSE number 4: 7
the COURSE number 5: 8
```

7)update student

Ask user for student ID if it found ask him for what he want to change

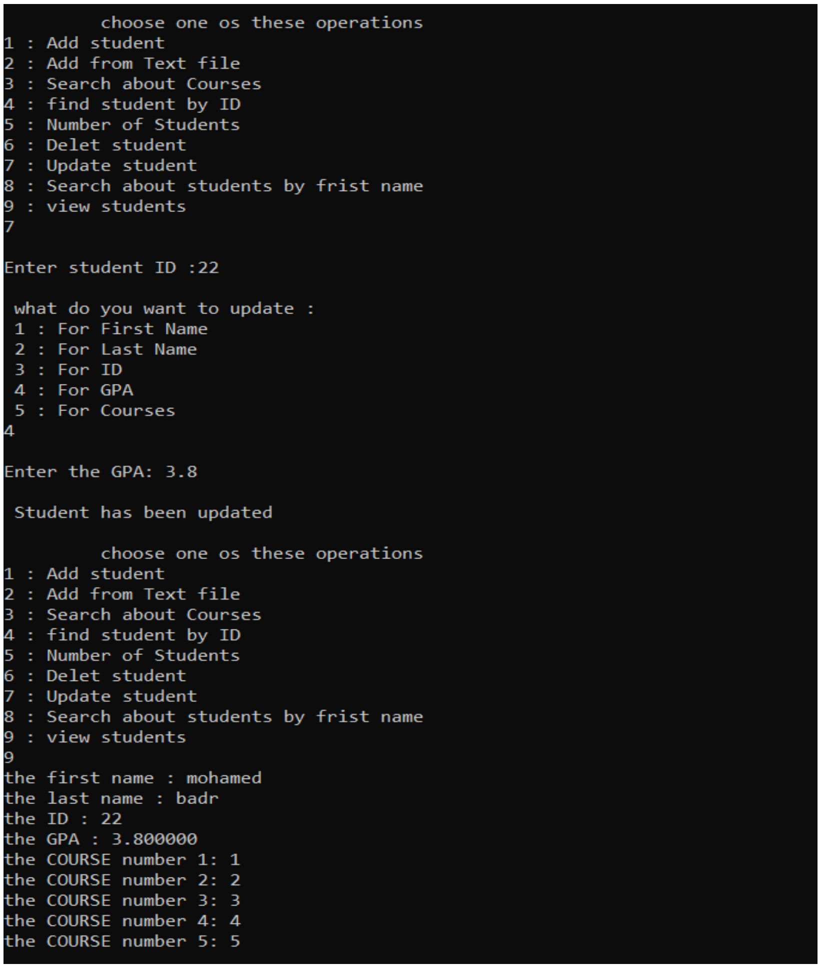


```
int update_student(Sdata* queue)
{
    char temp_text[40];
    char choice[40];
    int ST_ID;
    printf("\nEnter student ID :");
    gets(temp_text);
    ST_ID = atoi(temp_text);

    for(int i=0;i<max;i++)
    {
        if(queue->is_exist == 1)
        {
            if(queue->ID == ST_ID)
            {
                printf("\n what do you want to update :");
                printf(" 1 : For First Name \n");
                printf(" 2 : For Last Name \n");
                printf(" 3 : For ID \n");
                printf(" 4 : For GPA \n");
                printf(" 5 : For Courses \n");
                gets(choice);

                switch(atoi(choice))
                {
                  case 1:
                           printf("\nEnter the first name: ");
                            gets(queue->F_name);
                            break;
                  case 2:
                    printf("\nEnter the last name: ");
                    gets(queue->L_name);
                    break;
```



```
                   case 3:
                     printf("\nEnter the ID: ");
                     gets(temp_text);
                     queue->ID = atoi(temp_text);
                     break;
                   case 4:
                        printf("\nEnter the GPA: ");
                        gets(temp_text);
                        queue->GPA = atof(temp_text);
                        break;
                   case 5:
                            for(int i=0;i<5;i++)
                             {
                                printf("\nEnter ID course number %d: ",i);
                                gets(temp_text);
                                queue->courses[i]= atoi(temp_text);
                             }
                             break;

                 }
                             printf("\n Student has been updated \n");
                 return 1;

             }
         }
        queue++;
    }
    printf("\n Student not found \n");
    return 0;
}
```

Testing==>

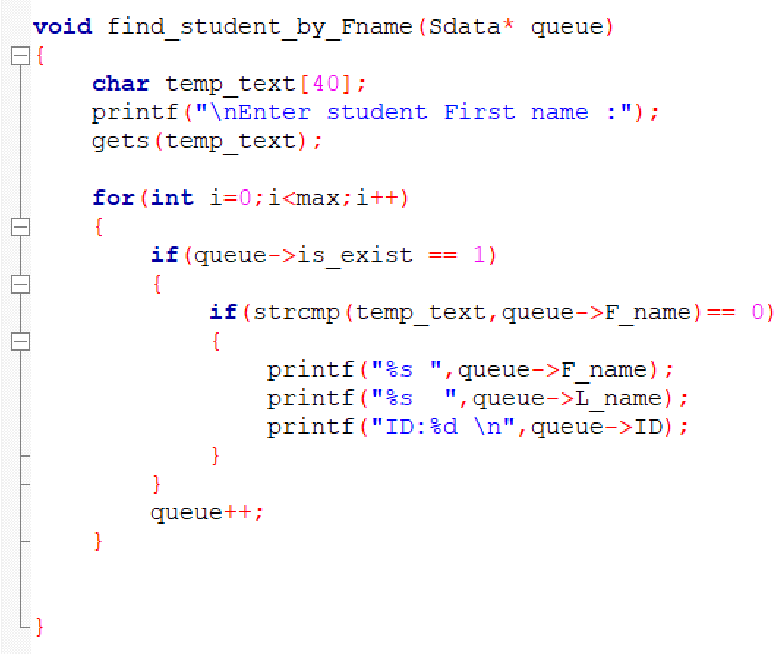


```
         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
7

Enter student ID :22

 what do you want to update :
 1 : For First Name
 2 : For Last Name
 3 : For ID
 4 : For GPA
 5 : For Courses
4

Enter the GPA: 3.8

 Student has been updated

         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
9
the first name : mohamed
the last name : badr
the ID : 22
the GPA : 3.800000
the COURSE number 1: 1
the COURSE number 2: 2
the COURSE number 3: 3
the COURSE number 4: 4
the COURSE number 5: 5
```

8)search for student by his first name

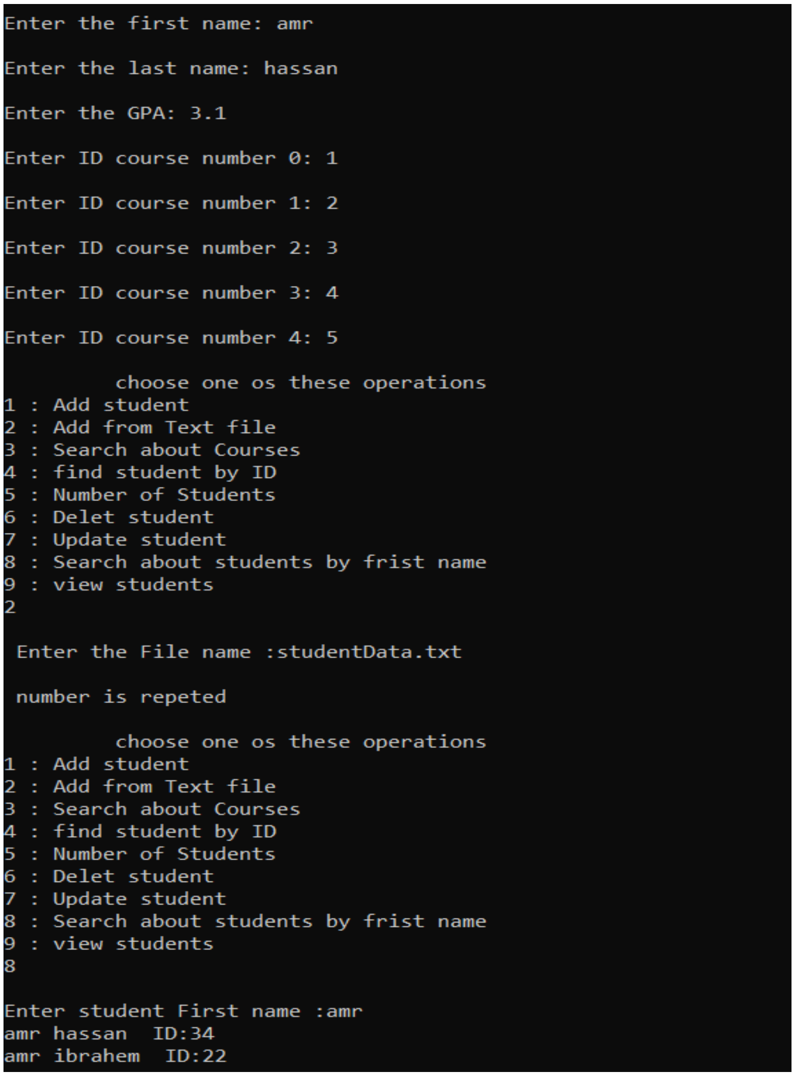
Ask user to enter the name and if there is more than one has this name the program

Show all of them



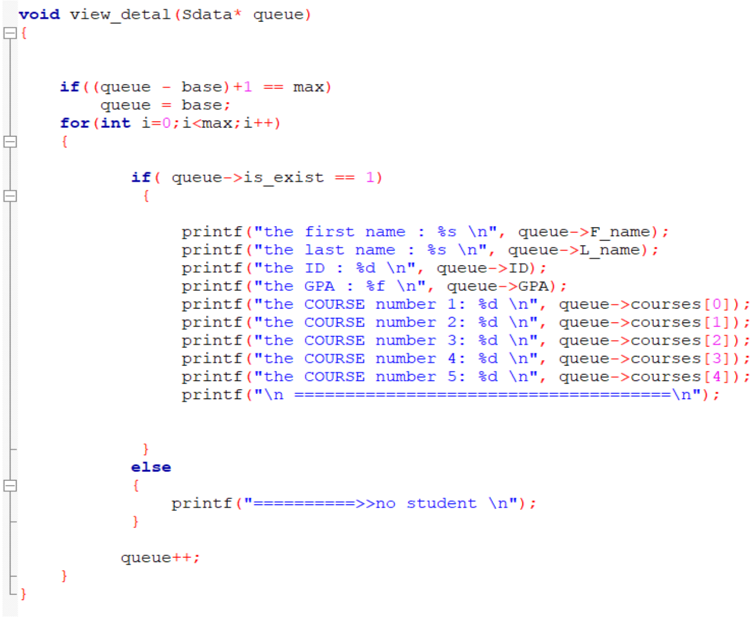
```
void find_student_by_Fname(Sdata* queue)
{
    char temp_text[40];
    printf("\nEnter student First name :");
    gets(temp_text);

    for(int i=0;i<max;i++)
    {
        if(queue->is_exist == 1)
        {
            if(strcmp(temp_text,queue->F_name)== 0)
            {
                printf("%s ",queue->F_name);
                printf("%s  ",queue->L_name);
                printf("ID:%d \n",queue->ID);
            }
        }
        queue++;
    }


}
```

Testing==>



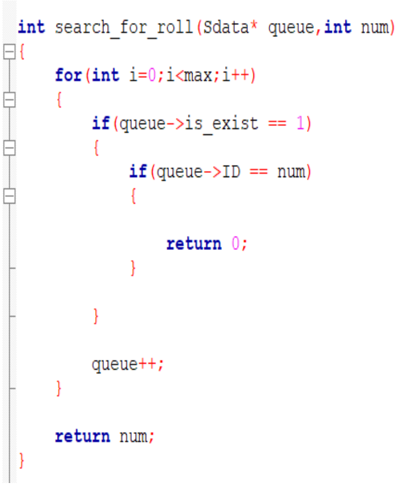
```
Enter the first name: amr

Enter the last name: hassan

Enter the GPA: 3.1

Enter ID course number 0: 1

Enter ID course number 1: 2

Enter ID course number 2: 3

Enter ID course number 3: 4

Enter ID course number 4: 5

         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
2

 Enter the File name :studentData.txt

 number is repeted

         choose one os these operations
1 : Add student
2 : Add from Text file
3 : Search about Courses
4 : find student by ID
5 : Number of Students
6 : Delet student
7 : Update student
8 : Search about students by frist name
9 : view students
8

Enter student First name :amr
amr hassan  ID:34
amr ibrahem  ID:22
```

9)function used to view student’s Data



```
void view_detal(Sdata* queue)
{

    if((queue - base)+1 == max)
        queue = base;
    for(int i=0;i<max;i++)
    {

           if( queue->is_exist == 1)
            {

                printf("the first name : %s \n", queue->F_name);
                printf("the last name : %s \n", queue->L_name);
                printf("the ID : %d \n", queue->ID);
                printf("the GPA : %f \n", queue->GPA);
                printf("the COURSE number 1: %d \n", queue->courses[0]);
                printf("the COURSE number 2: %d \n", queue->courses[1]);
                printf("the COURSE number 3: %d \n", queue->courses[2]);
                printf("the COURSE number 4: %d \n", queue->courses[3]);
                printf("the COURSE number 5: %d \n", queue->courses[4]);
                printf("\n =====================================\n");


            }
           else
           {
               printf("==========>>no student \n");
           }

          queue++;
    }
}
```

This function used to check if ID is already found or not



```
int search_for_roll(Sdata* queue,int num)
{
    for(int i=0;i<max;i++)
    {
        if(queue->is_exist == 1)
        {
            if(queue->ID == num)
            {

                return 0;
            }

        }

        queue++;
    }

    return num;
}
```

Gmail / badr89270@gmail.com

Learn in depth profile / https://www.learn-in-depth.com/online-diploma/badr89270%40gmail.com