

Report

Student management system

Using queue

Problem statement:

Write a program to build a 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.

Approach:

- 1- Add student details manually.
- 2- Add students details from text file.
- 3- Find student by a given roll number.
- 4- Find students by a given first name.
- 5- Find students registered in specific course.
- 6- Number of students registered.
- 7- Delete student by Roll number.
- 8- Update student information.

Implementation:

- 1- create structure to represent student details
- 2- create array of structures to act as a buffer that contains
All student's database limited by 100 record.
- 3- create structure to control the buffer and navigate through it
using pointers and counter.
- 4- create functions that achieves our target
- 5- create Enum to represent status of each function

Lets see code

1-Structures, Enum and buffer:

as we mentioned before

-structure to represent student

object.

- Buffer to contain the whole database.

- structure to control buffer.

- Enum to represent status.

2- initialization:

this function takes pointer to structure

(controller buffer) and pointer to

Buffer and buffer length, then return status

-first it checks if these addresses exist or

Not then adjust pointers of controller to

Points to the buffer and initialize counter.

3- Check for Roll number

this function takes pointer to buffer controller and take integer number represents Roll number wanted to be check if is exist before or not because the roll number must be a unique number for a student.

It returns 0 if roll number found and 1

If not found.

```
18 typedef struct { // student information structure
19     char fname[50];
20     char lname[50];
21     int roll;
22     float GPA;
23     int cid[10];
24 } st;
25 st buffer[100]; // the buffer contains students data
26 typedef struct{
27     st *head;
28     st *tail;
29     st *base;
30     int counter;
31     int length;
32 } x;
33 typedef enum{
34     fifo_no_error,
35     fifo_full,
36     fifo_empty,
37     fifo_null,
38     fifo_error
39 } fifo_buffer_state;
40 }
```

```

9 fifo_buffer_state fifo_init(x*fifo,st* buf,int lenght)
10 {
11     if(!fifo || !buf)
12     {
13         return fifo_null;
14     }
15     fifo->base=buf;
16     fifo->head=buf;
17     fifo->tail=buf;
18     fifo->length=lenght;
19     fifo->counter=0;
20     return fifo_no_error;
21 }
```

```
22 int check_roll(x*fifo,int x) // to check if roll number is exist before or not
23 { int y;
24     st* ptr=fifo->base;
25
26     for(y=0;y<(fifo->counter);y++)
27     {
28         if(ptr->roll==x)
29         {
30             return 0;
31         }
32         ptr++;
33     }
34     return 1;
35 }
```

4- add student manually

- this function to add students details manually takes pointer to Buffer controller.

-starts with checking if the buffer is existing or not then checks if the buffer is full or not.

- then starts to take student information and check if the Roll number has been taken before or not using check_roll function.

- after that it takes another data like first name and last name and GPA

- finally, it takes the student courses that should have a valid course id and the function will handle any non-valid course id

- if everything is going well, the function finally prints the max number of students that the buffer can hold and the number of empty places for new records.

```
22*int check_roll(x*fifo,int x) // to check if roll number is exist before or not
36=fifo_buffer_state add_student_manually(x*fifo)
37 {   int temp_int,y, x;
38     char temp_str[30];
39     if(!fifo->base || !fifo->head || !fifo->tail ) // check queue is exist or not
40     {
41         DPRINTF("database not exist \n");
42         return fifo_null;
43     }
44     if(fifo->counter==fifo->length) // check if full
45     {
46         DPRINTF("[ERROR] data base is full\n");
47         return fifo_full;
48     }
49     DPRINTF("-----\n");
50     DPRINTF("Add Student Details \n");
51     DPRINTF("-----\n");
52     DPRINTF("Enter the Roll Number\n");
53     gets(temp_str);
54     temp_int =atoi(temp_str);
55     if(check_roll(fifo,temp_int)==0)
56     {
57         DPRINTF("[ERROR] Roll Number is already taken before \n");
58         return fifo_error;
59     }
60     fifo->head->roll=atoi(temp_str);
61     DPRINTF("Enter First name of the student:\n");
62     gets(fifo->head->fname);
63     DPRINTF("Enter Last name of the student:\n");
64     gets(fifo->head->lname);
65     DPRINTF("Enter the GPA you obtained\n");
66     gets(temp_str);
67     fifo->head->GPA=atof(temp_str);
68     DPRINTF("Enter the course id of each course\n");
69     for(x=0;x<5;x++)
70     {
71         DPRINTF("course %d id :\n",x+1);
72         gets(temp_str);
73         y=atoi(temp_str);
74         if(y>0 && y<30) // check if course id is available id
75         {
76             fifo->head->cid[x]=y;
77             continue;
78         }
79         DPRINTF("[ERROR] course id is not correct\n");
80         x--;
81     }
82     fifo->head++;
83     fifo->counter++;
84     DPRINTF("[INFO] Student Details are added successfully \n");
85     DPRINTF("-----\n");
86     DPRINTF("[INFO] the total number of students is : %d\n",fifo->counter);
87     DPRINTF("[INFO] you can add up to %d students \n",fifo->length);
88     DPRINTF("[INFO] you can add up to %d students \n",fifo->length - fifo->counter);
89     DPRINTF("-----\n");
90     return fifo_no_error;
91 }
```

5-Show all database

- this function responsible for display all data for all students registered.
- It starts with checking if buffer is existing or not and check if the buffer is empty or not.
- It loops on the buffer and extract all data
- Finally, it informs us the total number of students

```
92=fifo_buffer_state show_s(x*fifo) // show all students information
93 { st *current_stuednt =fifo->base;
94     int x,y;
95     if(!fifo->base || !fifo->head || !fifo->tail ) // check queue is exist or not
96     {
97         DPRINTF("database not exist \n");
98         return fifo_null;
99     }
100     if(fifo->counter==0) // check if is empty
101     {
102         DPRINTF("[ERROR] database is empty \n");
103         DPRINTF("-----\n");
104         return fifo_empty;
105     }
106     for(x=0;x< fifo->counter;x++) // show students data
107     {
108         DPRINTF("-----\n");
109         DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
110         DPRINTF("Student first name : %s\n",current_stuednt->fname);
111         DPRINTF("Student last name : %s\n",current_stuednt->lname);
112         DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
113         for(y=0;y<5;y++)
114         {
115             DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
116         }
117         current_stuednt++;
118     }
119     DPRINTF("-----\n");
120     DPRINTF("total number of students : %d\n",fifo->counter);
121     return fifo_no_error;
122 }
```

6-find by roll number

- this function responsible for find the student data by roll number
- It starts with checking if buffer is existing or not and check if the buffer is empty or not.
- Then loop on all database to get student information by roll number.
- If roll number is not found function will inform us.

```
123=fifo_buffer_state find_r1(x*fifo) // find student data using Roll number
124 { char temp_str[30];
125     int temp_roll;
126     st *current_stuednt =fifo->base;
127     int x,y;
128     if(!fifo->base || !fifo->head || !fifo->tail ) // check queue is exist or not
129     {
130         DPRINTF("database not exist \n");
131         return fifo_null;
132     }
133     if(fifo->counter==0) // check if is empty
134     {
135         DPRINTF("[ERROR] database is empty \n");
136         DPRINTF("-----\n");
137         return fifo_empty;
138     }
139     DPRINTF("Enter student roll number \n");
140     gets(temp_str);
141     temp_roll=atoi(temp_str);
142
143     for(x=0;x< fifo->counter;x++) // loop to get the roll number
144     { if(current_stuednt->roll==temp_roll)
145     {
146         DPRINTF("-----\n");
147         DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
148         DPRINTF("Student first name : %s\n",current_stuednt->fname);
149         DPRINTF("Student last name : %s\n",current_stuednt->lname);
150         DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
151         for(y=0;y<5;y++)
152         {
153             DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
154         }
155         return fifo_no_error;
156     }
157     current_stuednt++;
158 }
159 DPRINTF("-----\n");
160 DPRINTF("[ERROR] Roll number is not found\n"); // loop finished and roll not found
161 DPRINTF("-----\n");
162 return fifo_error;
163 }
```

7-Find by first name

- this function searches about all students that share the same student's first name
- It starts with checking if buffer is existing or not and check if the buffer is empty or not.
- Then loop on all database to get student information by first name.
- If roll the name is not found function will inform us.

```
164 fifo_buffer_state find_fn(x*fifo) // find student data using first name
165 { char temp_str[30];
166   int flag=0;
167   st *current_stuednt =fifo->base;
168   int x,y;
169   if(!fifo->base || !fifo->head || !fifo->tail ) // check queue is exist or not
170   {
171     DPRINTF("database not exist \n");
172     return fifo_null;
173   }
174   if(fifo->counter==0) // check if is empty
175   { DPRINTF("[ERROR] database is empty \n");
176     DPRINTF("-----\n");
177     return fifo_empty;
178   }
179   DPRINTF("Enter student first name \n");
180   gets(temp_str);
181   for(x=0;x< fifo->counter;x++) // loop to get the roll number
182   { if(strcmpi(current_stuednt->fname,temp_str)==0) //compare strings without case sensitive
183     {
184       DPRINTF("-----\n");
185       DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
186       DPRINTF("Student first name : %s\n",current_stuednt->fname);
187       DPRINTF("Student last name : %s\n",current_stuednt->lname);
188       DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
189       for(y=0;y<5;y++)
190       {
191         DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
192       }
193       DPRINTF("-----\n");
194       flag=1; // flag to know the first name found at least 1 time
195     }
196     current_stuednt++;
197   }
198   if(flag==0)
199   {
200     DPRINTF("-----\n");
201     DPRINTF("[ERROR] No first name matched this name\n"); // loop finished and roll not found
202     DPRINTF("-----\n");
203     return fifo_error;
204   }
205   return fifo_no_error;
}
```

8-find all students registered in a specific course using course id

-this function searches about all students that registered the same course

-we search using course id

-we loop on buffer and inside each student object we loop on courses if our course id matches any of student courses the function will print the student information.

- if the loop finished and there are no students registered this course the function will print message to inform us.

```
207=fifo_buffer_state find_c(x*fifo) // display students info registered by course id
208 {
209     char temp_str[30];
210     int temp_course_id;
211     st *current_stuednt =fifo->base;
212     int x,y,z,flag=0;
213     if(!fifo->base || !fifo->head || !fifo->tail ) // check queue is exist or not
214     {
215         DPRINTF("database not exist \n");
216         return fifo_null;
217     }
218     if(fifo->counter==0) // check if is empty
219     {
220         DPRINTF("[ERROR] database is empty \n");
221         DPRINTF("-----\n");
222         return fifo_empty;
223     }
224     DPRINTF("Enter course id number \n");
225     gets(temp_str);
226     temp_course_id=atoi(temp_str);
227     for(x=0;x< fifo->counter;x++) // loop to get the course id
228     {
229         for(z=0;z<5;z++)
230         {
231             if(current_stuednt->cid[z]==temp_course_id) // if course id matches
232             {
233                 DPRINTF("-----\n");
234                 DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
235                 DPRINTF("Student first name : %s\n",current_stuednt->fname);
236                 DPRINTF("Student last name : %s\n",current_stuednt->lname);
237                 DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
238                 for(y=0;y<5;y++)
239                 {
240                     DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
241                     flag=1; // found at least one student
242                 }
243             }
244             current_stuednt++;
245         }
246         if(flag==0)
247         {
248             DPRINTF("-----\n");
249             DPRINTF("[ERROR] no student registered \n");
250             DPRINTF("-----\n");
251             return fifo_error;
252         }
253     }
254 }
```

9- Total number of students

This function gets the total number of students in the database

```
253=fifo_buffer_state tot_s(x*fifo) // total number of student
254 {
255     if(!fifo->base || !fifo->head || !fifo->tail ) // check queue is exist or not
256     {
257         DPRINTF("database not exist \n");
258         return fifo_null;
259     }
260     if(fifo->counter==0) // check if is empty
261     {
262         DPRINTF("[ERROR] database is empty \n");
263         DPRINTF("-----\n");
264         return fifo_empty;
265     }
266     DPRINTF("-----\n");
267     DPRINTF("[INFO] the total number of students is : %d\n",fifo->counter);
268     DPRINTF("[INFO] you can add up to %d students \n",fifo->length);
269     DPRINTF("[INFO] you can add more about %d students \n",fifo->length - fifo->counter);
270     DPRINTF("-----\n");
271     return fifo_no_error;
272 }
```

10- Delete student using roll number

- this function deletes student registration by roll number.
- First it loops on database to find the roll number
- It displays student information before delete
- Then asks us to confirm deleting process
- If roll number is not found it will inform us
- If we enter wrong option It will inform and back to main menu
- This function uses another function called `shift_buffer` we will discuss it later.

```

275 @fifo_buffer_state del_s(x*fifo) // delete student by roll num
276 { char temp_str[30];
277 int x,y,temp_roll,index=0;
278 st *current_stuednt=fifo->base;
279 if(!fifo->base || !fifo->head || !fifo->tail ) // check queue is exist or not
280 {
281     DPRINTF("database not exist \n");
282     return fifo_null;
283 }
284 if(fifo->counter==0) // check if is empty
285 {
286     DPRINTF("[ERROR] database is empty \n");
287     DPRINTF("-----\n");
288     return fifo_empty;
289 }
290 DPRINTF("Enter student roll number \n");
291 gets(temp_str);
292 temp_roll=atoi(temp_str);
293
294 for(x=0;x< fifo->counter;x++) // loop to get the roll number
295 { if(current_stuednt->roll==temp_roll)
296 {
297     DPRINTF("-----\n");
298     DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
299     DPRINTF("Student first name : %s\n",current_stuednt->fname);
300     DPRINTF("Student last name : %s\n",current_stuednt->lname);
301     DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
302     for(y=0;y<5;y++)
303     {
304         DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
305     }
306     DPRINTF("-----\n");
307     DPRINTF("Delete student 1-yes 2-No \n");
308     DPRINTF("-----\n");
309     gets(temp_str);
310     temp_roll=atoi(temp_str);
311     if(temp_roll==1){
312         shift_buffer(index,fifo); // to shift buffer and override on location wanted to be deleted
313         fifo->head--;
314         fifo->counter--;
315         return fifo_no_error;
316     }
317     else if(temp_roll==0)
318     {
319         DPRINTF("-----\n");
320         DPRINTF("-----Process canceled-----\n");
321         return fifo_no_error;
322     }
323     else
324     {
325         DPRINTF("-----\n");
326         DPRINTF("[ERROR]wrong choice ..\n Uncompleted process...\n back to main menu ..... \n");
327         return fifo_no_error;
328     }
329 }
330 current_stuednt++;
331 index++; //to find location in the buffer
332 }
333 DPRINTF("-----\n");
334 DPRINTF("[ERROR] Roll number is not found\n"); // loop finished and roll not found
335 DPRINTF("-----\n");
336 return fifo_error;

```

11 – shift buffer

This function responsible for shifting the deleted object location and all objects after it left to fill the free location of deleted object inside the buffer.

```

@fifo_buffer_state shift_buffer(int index,x*fifo) // to shift buffer and override
{
    int x;
    for(x=index;x<fifo->counter;x++)
    {
        buffer[x]=buffer[x+1];
    }
    DPRINTF("-----\n");
    DPRINTF("student deleted successfully\n");
    DPRINTF("-----\n");
    return fifo_no_error;
}

```


12-update student information

-this function updates a specific data in a previous registered student's data

- this data could be first name or second name or GPA or courses.

- it navigates on database based on student roll number.

- first it displays the data of student and allow us to choose the data wanted to be updated.

- if you entered wrong choice the function will return to the main menu with a message that is wrong choice.

-if you update data of student the function will display the student's information after updating.

-if you entered a wrong roll number it will tell you that is wrong roll number and back to main menu.

```
349= fifo_buffer_state up_s(x*fifo) // update student by roll num
350 {
351     char temp_str[30];
352     int x,y,i,j,temp_option, temp_roll;
353     st *current_stuednt =fifo->base;
354     if(!fifo->base || !fifo->head || !fifo->tail ) // check queue is exist or not
355     {
356         DPRINTF("database not exist \n");
357         return fifo_null;
358     }
359     if(fifo->counter==0) // check if is empty
360     {
361         DPRINTF("[ERROR] database is empty \n");
362         DPRINTF("-----\n");
363         return fifo_empty;
364     }
365     DPRINTF("Enter student roll number \n");
366     gets(temp_str);
367     temp_roll=atoi(temp_str);
368
369     for(x=0;x< fifo->counter;x++) // loop to get the roll number
370     { if(current_stuednt->roll==temp_roll)
371     {
372         DPRINTF("-----\n");
373         DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
374         DPRINTF("Student first name : %s\n",current_stuednt->fname);
375         DPRINTF("Student last name : %s\n",current_stuednt->lname);
376         DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
377         for(y=0;y<5;y++)
378         {
379             DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
380         }
381         DPRINTF("-----\n");
382         DPRINTF("Enter option to update data\n");
383         DPRINTF("1- first name\n");
384         DPRINTF("2- last name \n");
385         DPRINTF("3- GPA\n");
386         DPRINTF("4- courses \n");
387         DPRINTF("-----\n");
388         gets(temp_str);
389         temp_option=atoi(temp_str);
390         switch(temp_option)
391         {
392             case 1 :
393             {
394                 DPRINTF("Enter New first name\n");
395                 gets(current_stuednt->fname);
396                 break;
397             }
398             case 2 :
399             {
400                 DPRINTF("Enter New second name\n");
401                 gets(current_stuednt->lname);
402                 break;
403             }
404             case 3 :
405             {
406                 DPRINTF("Enter New GPA \n");
407                 gets(temp_str);
408                 current_stuednt->GPA=atof(temp_str);
409                 break;
410             }
411             case 4 :
412             {
413                 DPRINTF("Enter the course id of each course\n");
414                 for(i=0;i<5;i++)
415                 {
416                     DPRINTF("course %d id : \n",i+1);
417                     gets(temp_str);
418                     j=atoi(temp_str);
419                     if(j>0 && j<30) // check if course id is available id
420                     {
421                         current_stuednt->cid[i]=j;
422                         continue;
423                     }
424                     DPRINTF("[ERROR] course id is not correct\n");
425                     i--;
426                 }
427                 break;
428             }
429             default :
430             {
431                 DPRINTF("[ERROR] wrong choice \n");
432                 return fifo_error;
433             }
434         }
435     }
436     // print student information after update
437     DPRINTF("-----\n");
438     DPRINTF("information updated successfully \n");
439     DPRINTF("Student information after update \n");
440     DPRINTF("-----\n");
441     DPRINTF("Student Roll number : %d\n",current_stuednt->roll);
442     DPRINTF("Student first name : %s\n",current_stuednt->fname);
443     DPRINTF("Student last name : %s\n",current_stuednt->lname);
444     DPRINTF("Student GPA : %.2f\n",current_stuednt->GPA);
445     for(y=0;y<5;y++)
446     {
447         DPRINTF("course %d id : %d \n",y+1,current_stuednt->cid[y]);
448     }
449     return fifo_no_error;
450 }
451 current_stuednt++;
452 }
453 DPRINTF("-----\n");
454 DPRINTF("[ERROR] Roll number is not found\n"); // loop finished an
455 DPRINTF("-----\n");
456 return fifo_error;
457 }
```

13- Add student from text File

-this function reads from text file and save students information into data base.

-after checking buffer exist or not and is full or not the function stars connection with the text file

- it checks if roll number is taken or not and if this condition happened you get message with line error in text file and skip this record.

- if any student has non-valid course id, a message gets printed and this student is skipped.

-if buffer size reaches the full size

The function will stop adding and

A message gets printed with the number of students added and the remaining students.

-If all things are ok a message is printed with the number of students added and the number of error in students information

```
460 *fifo_buffer_state add_student_file(x*fifo) // add students into using text file
461 {
462     char f_name[50];
463     char l_name[50];
464     int roll_num,cid[5],x,file_count=0,flag=0;
465     float GPA;
466     int line =0;
467     if(!fifo->base || !fifo->head || !fifo->tail ) // check queue is exist or not
468     {
469         DPRINTF("database not exist \n");
470         return fifo_null;
471     }
472     if(fifo->counter==fifo->length) // check if full
473     {
474         DPRINTF("[ERROR] data base is full\n");
475         return fifo_full;
476     }
477     FILE *p_file = fopen("text.txt","r");
478     if(p_file==NULL)
479     {
480         DPRINTF("----- \n");
481         DPRINTF("[ERROR] File not found \n");
482         return fifo_error;
483     }
484
485     // reading from file
486
487     while( fscanf(p_file,"%d %s %s %f %d %d %d [%^n]", &roll_num, f_name,l_name,&GPA,&cid[0],&cid[1],&cid[2],&cid[3],&cid[4])!=EOF)
488     {
489         if(fifo->counter==fifo->length)
490         { DPRINTF("----- \n");
491           DPRINTF("[ERROR] data base is full\n");
492           DPRINTF("[INFO] students added : %d\n",file_count);
493           DPRINTF("[INFO] remaining students due to size or errors are :%d\n",line-file_count);
494
495           return fifo_full;
496         }
497         if(check_roll(fifo,roll_num)==0)
498         {
499             DPRINTF("[ERROR] IN line %d : Roll Number is already taken before \n",line);
500             line++;
501             continue; // to skip this student
502         }
503         fifo->head->roll=roll_num;
504         fifo->head->GPA=GPA;
505         strcpy(fifo->head->fname,f_name);
506         strcpy(fifo->head->lname,l_name);
507         for(x=0;x<5;x++)
508         {
509             if(cid[x]<0 || cid[x]>30)
510             {
511                 flag=1; // that there is non-valid course id
512                 break;
513             }
514             fifo->head->cid[x]=cid[x];
515         }
516         if(flag==1) // non valid course id
517         {
518             DPRINTF("[ERROR] IN line %d : non valid course id we will skip this student \n",line);
519             line++;
520             continue; // to skip this student
521         }
522         fifo->head++;
523         fifo->counter++;
524         line++;
525         file_count++; // to record successful records
526     }
527     DPRINTF("\nEnd of file.\n");
528
529     // close connection
530     fclose(p_file);
531     DPRINTF("[INFO] students added : %d\n",file_count);
532     DPRINTF("[INFO] remaining students due to errors are :%d\n",line-file_count);
533     return fifo_error;
534 }
535 }
```

14- main function

```
12 #include "queue.h"
13
14 int main(void) {
15     int temp;
16     x buffer_controller; // that controls student buffer
17     fifo_init(&buffer_controller,buffer,100);
18
19     DPRINTF("Welcome to the Student Management System\n");
20     while(1)
21     {
22         DPRINTF("-----\n");
23         DPRINTF("Choose The Task that you want to perform\n");
24         DPRINTF("1. Add the Student Details Manually\n");
25         DPRINTF("2. Add the Student Details From Text File\n");
26         DPRINTF("3. Find the Student Details by Roll Number\n");
27         DPRINTF("4. Find the Student Details by First Name\n");
28         DPRINTF("5. Find the Student Details by Course ID\n");
29         DPRINTF("6. Find the Total number of Students\n");
30         DPRINTF("7. Delete the Student Details by Roll Number \n");
31         DPRINTF("8. Update the Student Details by Roll Number \n");
32         DPRINTF("9. Show all information\n");
33         DPRINTF("Enter your choice to perform the task\n");
34         scanf("%d",&temp);
35         switch(temp)
36         {
37             case 1 :
38             {
39                 add_student_manually(&buffer_controller);
40                 break;
41             }
42             case 2 :
43             {
44                 add_student_file(&buffer_controller);
45                 break;
46             }
47             case 3 :
48             {
49                 find_r1(&buffer_controller);
50                 break;
51             }
52             case 4 :
53             {
54                 find_fn(&buffer_controller);
55                 break;
56             }
57             case 5 :
58             {
59                 find_c(&buffer_controller);
60                 break;
61             }
62             case 6 :
63             {
64                 tot_s(&buffer_controller);
65                 break;
66             }
67             case 7 :
68             {
69                 del_s(&buffer_controller);
70                 break;
71             }
72             case 8 :
73             {
74                 up_s(&buffer_controller);
75                 break;
76             }
77             case 9 :
78             {
79                 show_s(&buffer_controller);
80                 break;
81             }
82             default :
83             {
84                 DPRINTF("Wrong choice\n");
85             }
86         }
87     }
88 }
```

Testing code logic

1-Adding students manually

We will add 4 students with the same way.

```
Welcome to the Student Management System
-----
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
Enter your choice to perform the task
1
-----
Add Student Details
-----
Enter the Roll Number
1
Enter First name of the student:
shady
Enter Last name of the student:
mamdouh
Enter the GPA you obtained
2.5
Enter the course id of each course
course 1 id :
1
course 2 id :
2
course 3 id :
3
course 4 id :
4
course 5 id :
5
[INFO] Student Details are added successfully
-----
[INFO] the total number of students is : 1
[INFO] you can add up to 100 students
[INFO] you can add more about 99 students
```

```
9. Show all information
Enter your choice to perform the task
```

```
9
-----
Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
```

```
-----
Student Roll number : 2
Student first name : keroles
Student last name : shenouda
Student GPA : 3.80
course 1 id : 10
course 2 id : 5
course 3 id : 3
course 4 id : 2
course 5 id : 7
```

```
-----
Student Roll number : 3
Student first name : ahmed
Student last name : tawfek
Student GPA : 2.00
course 1 id : 20
course 2 id : 7
course 3 id : 2
course 4 id : 5
course 5 id : 1
```

```
-----
Student Roll number : 4
Student first name : shady
Student last name : essam
Student GPA : 2.50
course 1 id : 3
course 2 id : 5
```

2-Adding from text file

text.txt

```
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
Enter your choice to perform the task
2
[ERROR] IN line 0 : Roll Number is already taken before
[ERROR] IN line 2 : non valid course id we will skip this student
End of file.
[INFO] students added : 2
[INFO] remaining students due to errors are :2
-----
```

```
Student_management_system.c  queue.c  queue.h  text.txt x
11 shady mamdouh 3.5 1 2 3 4 5
28 mohamed hassan 3.5 1 11 9 6 5
36 ahmed ali 2.5 1 2 3 4 50
49 diaa saad 3.1 5 7 8 6 3
```

We have 2 students will added successfully, and 2 students will not be added

One of them has roll number (roll number 1) which is taken before and another entered a non-valid course id(50) as we know the valid course id range from 1 to 29

so look at text file

shady and **ahmed** will not be added.

Mohamed and **diaa** will be added successfully.

Before add

```
9. Show all information
Enter your choice to perform the task
9
-----
Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-----
Student Roll number : 2
Student first name : keroles
Student last name : shenouda
Student GPA : 3.80
course 1 id : 10
course 2 id : 5
course 3 id : 3
course 4 id : 2
course 5 id : 7
-----
Student Roll number : 3
Student first name : ahmed
Student last name : tawfek
Student GPA : 2.00
course 1 id : 20
course 2 id : 7
course 3 id : 2
course 4 id : 5
course 5 id : 1
-----
Student Roll number : 4
Student first name : shady
Student last name : essam
Student GPA : 2.50
course 1 id : 3
course 2 id : 5
course 3 id : 8
course 4 id : 12
course 5 id : 6
-----
Student Roll number : 8
Student first name : mohamed
Student last name : hassan
Student GPA : 3.50
course 1 id : 1
course 2 id : 11
course 3 id : 9
course 4 id : 6
course 5 id : 5
-----
Student Roll number : 9
Student first name : diaa
Student last name : saad
Student GPA : 3.10
course 1 id : 5
course 2 id : 7
course 3 id : 8
course 4 id : 6
course 5 id : 3
-----
total number of students : 6
-----
```

after add

```
9. Show all information
Enter your choice to perform the task
9
-----
Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-----
Student Roll number : 2
Student first name : keroles
Student last name : shenouda
Student GPA : 3.80
course 1 id : 10
course 2 id : 5
course 3 id : 3
course 4 id : 2
course 5 id : 7
-----
Student Roll number : 3
Student first name : ahmed
Student last name : tawfek
Student GPA : 2.00
course 1 id : 20
course 2 id : 7
course 3 id : 2
course 4 id : 5
course 5 id : 1
-----
Student Roll number : 4
Student first name : shady
Student last name : essam
Student GPA : 2.50
course 1 id : 3
course 2 id : 5
course 3 id : 8
course 4 id : 12
course 5 id : 6
-----
Student Roll number : 8
Student first name : mohamed
Student last name : hassan
Student GPA : 3.50
course 1 id : 1
course 2 id : 11
course 3 id : 9
course 4 id : 6
course 5 id : 5
-----
Student Roll number : 9
Student first name : diaa
Student last name : saad
Student GPA : 3.10
course 1 id : 5
course 2 id : 7
course 3 id : 8
course 4 id : 6
course 5 id : 3
-----
total number of students : 6
-----
```

3-get by roll number

```
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
Enter your choice to perform the task
3
Enter student roll number
8
[-----
Student Roll number : 8
Student first name : mohamed
Student last name : hassan
Student GPA : 3.50
course 1 id : 1
course 2 id : 11
course 3 id : 9
course 4 id : 6
course 5 id : 5
-----]
```

4-get all students share the same first name

We have 2 students share first name

Shady

```
-----
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
Enter your choice to perform the task
4
Enter student first name
shady
-----

Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-----

Student Roll number : 4
Student first name : shady
Student last name : essam
Student GPA : 2.50
course 1 id : 3
course 2 id : 5
course 3 id : 8
course 4 id : 12
course 5 id : 6
-----
```

5-Get all students registered in a specific course id

For ex course id: 2

we have 3 students from 6 registered in course id 2

```
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
Enter your choice to perform the task
5
Enter course id number
2
-----
Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-----
Student Roll number : 2
Student first name : keroles
Student last name : shenouda
Student GPA : 3.80
course 1 id : 10
course 2 id : 5
course 3 id : 3
course 4 id : 2
course 5 id : 7
-----
Student Roll number : 3
Student first name : ahmed
Student last name : tawfek
Student GPA : 2.00
course 1 id : 20
course 2 id : 7
course 3 id : 2
course 4 id : 5
course 5 id : 1
-----
```

6-Get total number of students

we have 6 students

```
-----
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
Enter your choice to perform the task
6
[-----
[INFO] the total number of students is : 6
[INFO] you can add up to 100 students
[INFO] you can add more about 94 students
-----
```


7-Delete student by roll number

If you enter the roll number to be delete the function displays the student information and asks you to confirm deleting process.

```
-----
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
Enter your choice to perform the task
7
Enter student roll number
3
-----
Student Roll number : 3
Student first name : ahmed
Student last name : tawfek
Student GPA : 2.00
course 1 id : 20
course 2 id : 7
course 3 id : 2
course 4 id : 5
course 5 id : 1
-----
Delete student 1-yes 2-No
-----
1
[-----
student deleted successfully
-----
```

Student “ahmed tawfek” has been

Deleted and total number of students decreased by 1

-this function uses another function to shift the deleted location from the buffer

Show all students after delete

```
Choose The Task that you want to perform
1. Add the Student Details Manually
2. Add the Student Details From Text File
3. Find the Student Details by Roll Number
4. Find the Student Details by First Name
5. Find the Student Details by Course ID
6. Find the Total number of Students
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
Enter your choice to perform the task
9
[-----
Student Roll number : 1
Student first name : shady
Student last name : mamdouh
Student GPA : 2.50
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-----
Student Roll number : 2
Student first name : keroles
Student last name : shenouda
Student GPA : 3.80
course 1 id : 10
course 2 id : 5
course 3 id : 3
course 4 id : 2
course 5 id : 7
-----
Student Roll number : 4
Student first name : shady
Student last name : essam
Student GPA : 2.50
course 1 id : 3
course 2 id : 5
course 3 id : 8
course 4 id : 12
course 5 id : 6
-----
Student Roll number : 8
Student first name : mohamed
Student last name : hassan
Student GPA : 3.50
course 1 id : 1
course 2 id : 11
course 3 id : 9
course 4 id : 6
course 5 id : 5
-----
Student Roll number : 9
Student first name : diaa
Student last name : saad
Student GPA : 3.10
course 1 id : 5
course 2 id : 7
course 3 id : 8
course 4 id : 6
course 5 id : 3
-----
total number of students : 5
-----
```


8-Update specific data of a student

For example:

update Diaa's GPA

```
7. Delete the Student Details by Roll Number
8. Update the Student Details by Roll Number
9. Show all information
Enter your choice to perform the task
8
Enter student roll number
9
-----
Student Roll number : 9
Student first name : diaa
Student last name : saad
Student GPA : 3.10
course 1 id : 5
course 2 id : 7
course 3 id : 8
course 4 id : 6
course 5 id : 3
-----
Enter option to update data
1- first name
2- last name
3- GPA
4- courses
-----
3
Enter New GPA
2.8
[-----
information updated successfully
Student information after update
-----
Student Roll number : 9
Student first name : diaa
Student last name : saad
Student GPA : 2.80
course 1 id : 5
course 2 id : 7
course 3 id : 8
course 4 id : 6
course 5 id : 3
-----
```

update Diaa courses id

note: if you try to enter a non-valid course id an error message will be printed as shown below:

```
Student GPA : 2.80
course 1 id : 5
course 2 id : 7
course 3 id : 8
course 4 id : 6
course 5 id : 3
-----
Enter option to update data
1- first name
2- last name
3- GPA
4- courses
-----
4
Enter the course id of each course
course 1 id :
1
course 2 id :
90
[ERROR] course id is not correct
course 2 id :
2
course 3 id :
3
course 4 id :
4
course 5 id :
5
[-----
information updated successfully
Student information after update
-----
Student Roll number : 9
Student first name : diaa
Student last name : saad
Student GPA : 2.80
course 1 id : 1
course 2 id : 2
course 3 id : 3
course 4 id : 4
course 5 id : 5
-----
```

9-show all database

```
-----  
Student Roll number : 1  
Student first name : shady  
Student last name : mamdouh  
Student GPA : 2.50  
course 1 id : 1  
course 2 id : 2  
course 3 id : 3  
course 4 id : 4  
course 5 id : 5  
-----  
Student Roll number : 2  
Student first name : keroles  
Student last name : shenouda  
Student GPA : 3.80  
course 1 id : 10  
course 2 id : 5  
course 3 id : 3  
course 4 id : 2  
course 5 id : 7  
-----  
Student Roll number : 4  
Student first name : shady  
Student last name : essam  
Student GPA : 2.50  
course 1 id : 3  
course 2 id : 5  
course 3 id : 8  
course 4 id : 12  
course 5 id : 6  
-----  
Student Roll number : 8  
Student first name : mohamed  
Student last name : hassan  
Student GPA : 3.50  
course 1 id : 1  
course 2 id : 11  
course 3 id : 9  
course 4 id : 6  
course 5 id : 5  
-----  
Student Roll number : 9  
Student first name : diaa  
Student last name : saad  
Student GPA : 2.80  
course 1 id : 1  
course 2 id : 2  
course 3 id : 3  
course 4 id : 4  
course 5 id : 5  
-----  
total number of students : 5  
-----
```

Good luck

