Record Students Data

Abstract

Our program is to record students data and restoring. We are recording students data , names, IDs , ages , grade levels and sheets mark in an arrays of structure. In addition to structures and arrays We also use pointer variables to pass data to function in some cases.

The user is to deal with 4 main functions, add student, Search for student , Modify student name , delete student records.

First of all, user have to input to the program the password, the he can choose what he wants.

Introduction

Well, the program's idea based on internally dividing the program into a set of functions, each function is independent by its work. The user will be asked to choose one of four options then the program will read the value and put it into ***Switch* statement**, that leads to the chosen function, one of five:

- The first is to **Adding** and receiving new student data

  User is to input:

  1. ID as an integer variable.
  2. name as a string value nothing that not to use the space " " in the name because the program depends on ***scanf function*** that will stop reading when reaching space or ENTER value we could use underscore sign instead.
  3. .4 Age and grade level as integer variables

5. Sheet Mark as a float variable nothing that it will restore just 2 numbers after decimal point.

The main problem here was how to store that data and control management of each element individually, that enables us to edit , search or delete it. Also, we have to make sure we have every student data separately ... So, we managed to use **array of structures.**

We can define **structure** as a composite datatype with a collection of variables. These variables can have different data types and collectively form a structure of a composite datatype. And array of structures is a sequential collection of structures. With structures, we can store mixed record types, and with an array supporting this, we can have a list of mixed record types.

Our program here shows a structure called ***student*** that takes the ID, name, age, grade and sheet marks of a student as an input, then stores each record in an array ***ST,*** which stores the elements. Each element will hold a mixed record.

After receiving student data and restoring it the program will show up the message of *"student added"* and return automatically to **Main Menu.**

- To **Research** for a student

the user will choose of two options whether searching with id or name, program will receive the choice and restore in in temporary variable then comparing it with the stored value in if function.

If choosing **1,** the user will be asked to enter Id to search for, Id temporary variable will be sent to The second function, ***srch***, that will compare the inputting id to all array elements using *for loop* to reach the demanded student then the student details will be printed to the screen.
If choosing **2,** the user will be asked to enter a student name, then restoring it in temporary string variable then sent it to

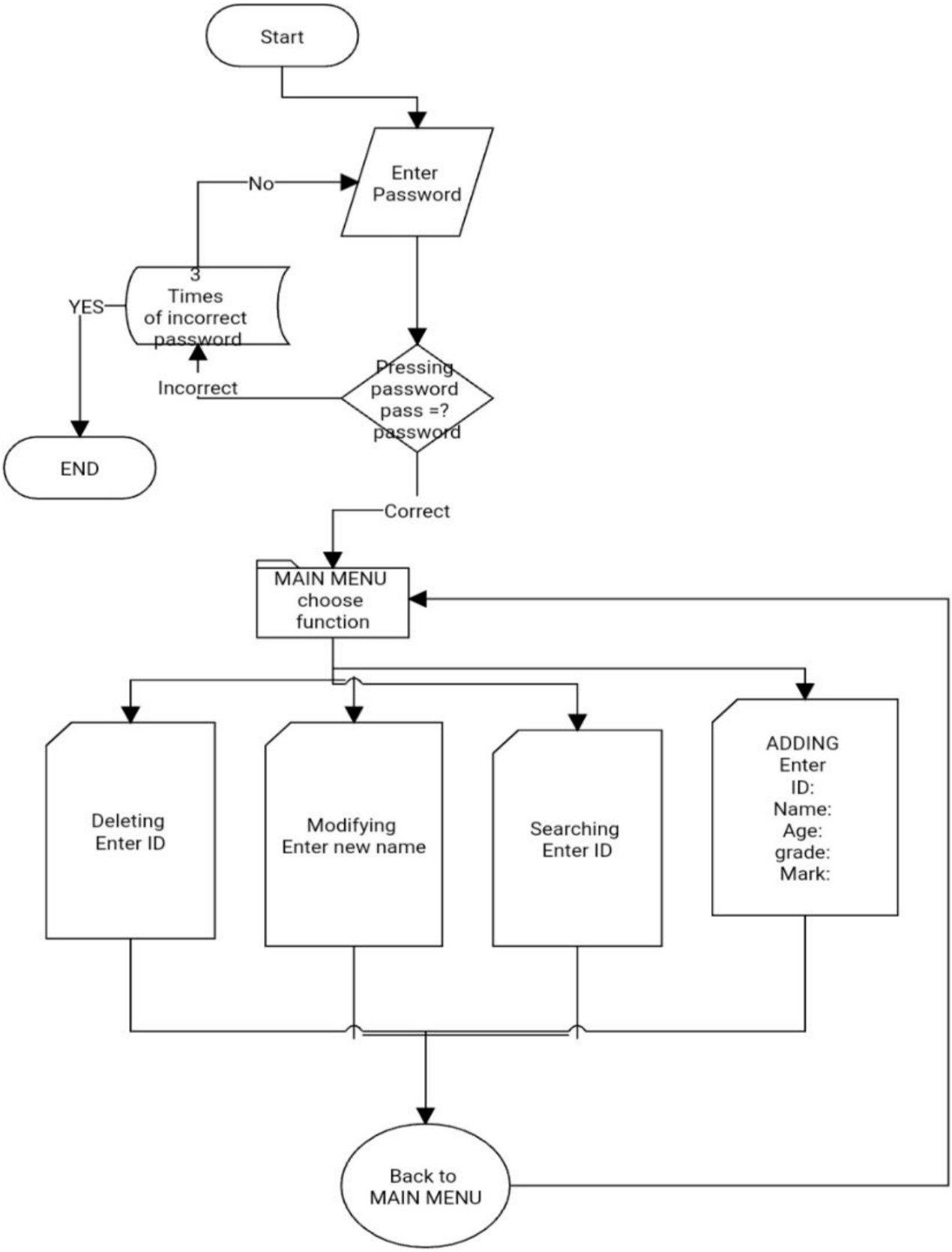
the third function ***srchn***, that compares the inputting to all array elements using *for loop* and the comparing function ***strcmp***, to reach the demanded student no. in the array then the student details will be printed to the screen.

- The fourth is to **Modifying** student name by entering his ID then ***pointer variable*** will passed id to the function that searching in all arrays elements, using *for loop and if function*, if that student data is existed or not, if it's found the program will show up stored name and ask the user to enter the new name, that will reads by *scanf function*, and replace the pervious valve.
- The fifth is to **Delete** student record, user will be asked to enter id first then program will search for it if existed or not, if found the program will replace all variables of the array with null value that already made just if the condition is fulfilled.

In every function the program will search whether the entrance variable's value is valid or not, if not program will back to ***The Main Menu.***

Also, after preforming every function the program will back to ***The Main Menu*** to enable the user to choose the next step.

The algorithm **flow charts** we use is to be :



Results:

The code:

```
 //Included Files
#include <stdio.h <
#include <string.h<

//STRUCT Declaration

struct student
}
 char name[20];
 int id;
 int age;
 int gradelevel;
 float sheetmark;
;{
 // Fictions Declaration
void add(struct student[]);
void srch(struct student[], int);
void srchn(struct student[] , char[]);
void modn(struct student[] ,int* );
void delete(struct student[]);

 //   main function//

 int main()
}
 //declaring a variable of the type student
    struct student  ST[100];
 //Welcoming

 printf("\n\t\t\t\t\tWelcome to Our Record_Studends_Data Progrm
\n\n" );

// checking password

 int i =1 , sid;
 l3: //line used by goto statement to reture back
```

```
 printf("\n\tplease enter the password\n\t");
long pass;
scanf("%ld",&pass); //reads password

if(pass==4321)  //Verifying //4321 used as a defult
}
printf("\n Password is Valid\n\n \tSuccessful_LogIn ..\n \nchoose
wht you want");
 l5 : //line used by goto statement to reture back


printf("\n #MIAN_MENU .. \npress\n \t\t\t1 for Add_student\n
\t\t\t2 for seacrch_st\n \t\t\t3 for modify_student_name\n \t\t\t4 for
delete_st\n\n ");
// inputting choise
int chf;
        scanf("%d", &chf);
//    making a decision
       switch (chf)
}
            case 1:
         add( ST ); //go to Adding function

             goto l5; //return back to main menu                    break;
            case 2:
printf("want to search\n with Id press 1\nwith name press 2\n");
int ch2;
          scanf("%d",&ch2); //input choice
if(ch2==1) //Verifying
}
printf("please enter student ID to search ..\n" );
  scanf("%d", &sid); //inputting ID
            srch(ST, sid); //go to search by name function
               goto l5; //return back to main menu

{
else  }
  char sna[20];  //temporary variable
printf("please enter student name to search ..\n" ) ;
fflush(stdin)   ;
scanf("%s", sna) ;

         srchn(ST, sna); //go to search by name function
```

```
                goto l5; //return back to main menu

{

            break;

          case 3:

printf("enter st id\n");
int ids;
   scanf("%d",&ids); //inputting id
      int *p; //pointer value
     p = &ids ;
         modn(ST,p ); //go to modyfiy name function
          goto l5; //back to the main menu
              break;

          case 4:

         delete(ST); // go to delete function

            goto l5; //back to main menu
              break;
{

{
  else
}
printf("\n\n\tsorry!\t\b Password is  Invalid\n\t ");
if(i<3) //Verifying
}
                i; ++

goto l3 ; //Go back to enter password again
{
 else
*/failed to enter password for 3_Times
       So sadly
     program is to Exit/*

printf("\n\n\nprogram is Exited ");
   return 0 ; //Exit//
```

```
{

{
/*************************************************/
 //Function declaration
 //Adding function

void add(struct student ST[100] )
}

printf("\nEnter student data\n");
printf("\nEnter st ID : ");
int idd;
  scanf("%d", &idd); //inputting ID
 ST[idd].id = idd ;
*/note: space " " is bannedd
    you can use underscore "_" instead/*
printf("Enter st name :");
  scanf("%s",ST[idd].name); //inputting name
printf("\nEnter st age : ");
        scanf("%d", &ST[idd].age);  //inputting age

printf("Enter st gradelevel (1,2,3,4) : ");

  scanf("%d", &ST[idd].gradelevel); //inputting grade level

printf("\nEnter st sheet mark (~100) : ");
        scanf("%f", &ST[idd].sheetmark); //inputting Mark
printf("\nstudent added\n\n");
{
/*************************************************/
 //   Search Functions
 //    Search wz ID function

void srch(struct student ST[100],int sid)
}
  if (ST[sid].id == sid)  // Verifying
}
printf("ID : %d\nName : %s\nage : %d\ngradeleve : %d\nSheet
mark : %5.2f\n" , ST[sid].id,ST[sid].name,
ST[sid].age,ST[sid].gradelevel,ST[sid].sheetmark);
{
```

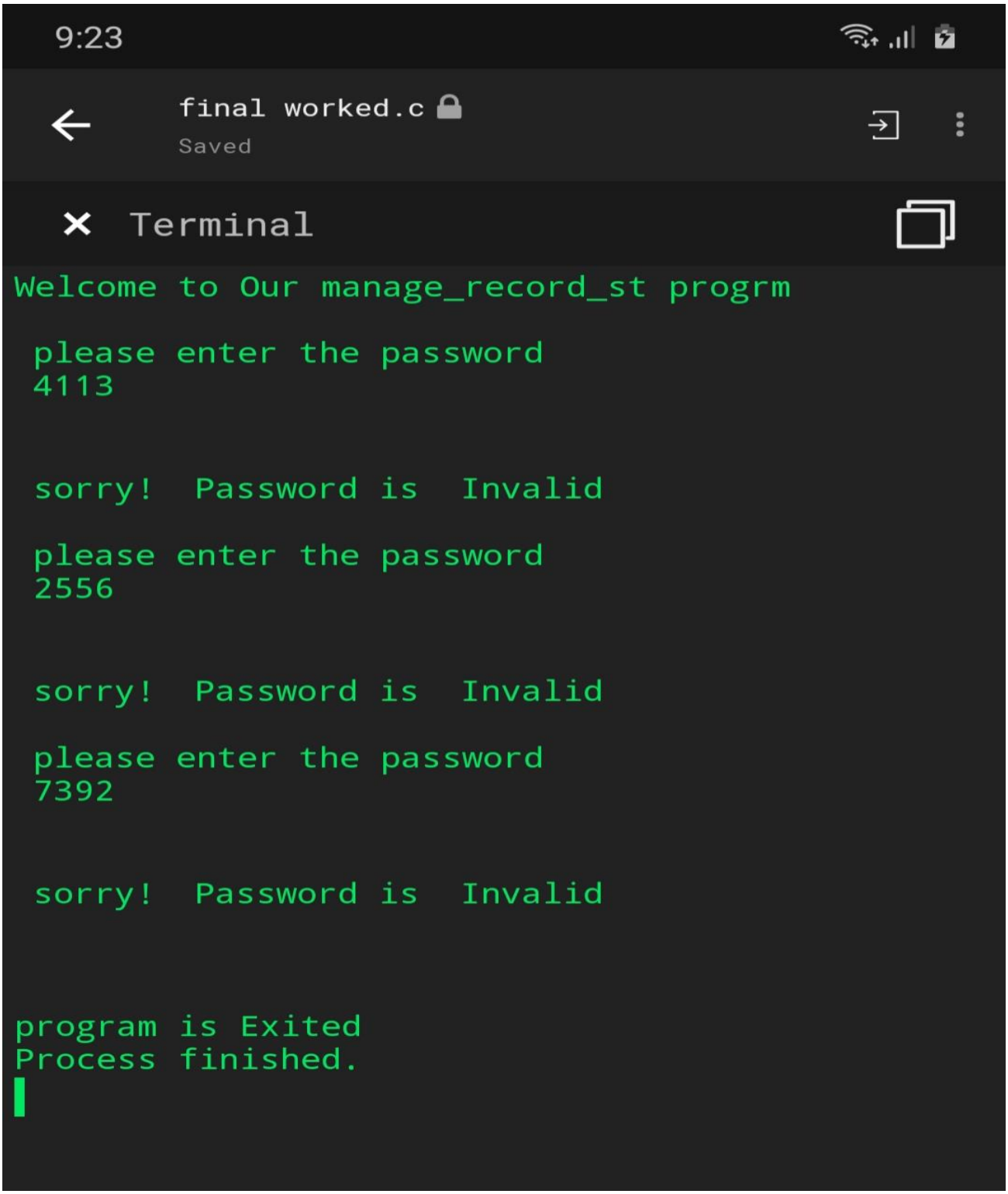
```
    else { printf("Record not Found\n");}
{

 // Search wz name

void srchn(struct student ST[100] , char sna[20])
}

int i; int k=1 ;
 //  search in all array elements
for (i=0;i<100;i++)}
//  comparing to the stored data
if (strcmp(ST[i].name,sna)==0 ) //Verifying
}
printf("ID : %d\nName : %s\nage : %d\ngradeleve : %d\nSheet
mark : %5.2f\n" , ST[i].id,ST[i].name,
ST[i].age,ST[i].gradelevel,ST[i].sheetmark);
 k++; // to tell us if the order made or NOT
{
{
 //if the order did made then k will reminds 1//
if(k==1){printf("\n Record Not Found");}
{

/*****************************************************/

//Modifying name using ID
void modn(struct student ST[100] ,int *p)
 }
 int k=1 ;
// searching in all array elements
int i; for (i=0;i<100;i++)
 }
 int a  = *p;
   if (ST[i].id == a) // found the demanded student
}

printf("Current name: %s\n " ,ST[i].name);

printf("enter the new name\n>>> ");
```

```
 //     to make scanf read properly
   fflush(stdin) ;
  scanf("%s", ST[i].name); // receive new name
          printf("\n\n Updated\n\n");

  k; ++
{
{
 // k will remain 1 if the previous condition didn't made

 if(k==1){printf("\n Record Not Found\n\n");}
{

/*****************************************************/

 //  delete_ function
void delete(struct student ST[100])
}
int i;
printf("enter st id\n you want to delete his record\n");
int ids; // just temporary variable

scanf("%d",&ids); //inputting Id
char emp[20] = { '\0' }; //just an empty string

  if(ST[ids].id==ids) //Verifying
}
//emptying..
     strcpy(ST[ids].name,emp); //copy empty string into name string
to empty it
     ST[ids].id=0;   //null value//
     ST[ids].age=0;  //null value //
     ST[ids].age=0;  //null value//
     ST[ids].gradelevel=0;  //null value//
     ST[ids].sheetmark=0.0; //null value//
 //     show up the updates
printf("ID : %d\nName : %s\nage : %d\ngradeleve : %d\nSheet
mark : %5.2f\n" , ST[ids].id,ST[ids].name,
ST[ids].age,ST[ids].gradelevel,ST[ids].sheetmark);

printf("\n\t\tRecord DELETED\n\n")           ;
{
else
```

```
printf("Record Not found");
{

*****************************************************/
/**************************
*****************************************************/
*****************************************************
/**********************
#//   Thanks   //

*****************************************************/
*****************************************************
/**********************

 /*****************************************************
                    ***************************/
```

5 Screenshots showing up the performance of the program :

1. Entering the password 3 times incorrectly leads to Exit the program



2. Adding function

```
9:24

final worked.c
Saved

Terminal

Welcome to Our manage_record_st progrm

 please enter the password
 4321

 Password is Valid

  Successful_LogIn ..

choose wht you want
 #MIAN_MENU ..
press
    1 for Add_student
    2 for seacrch_st
    3 for modify_student_name
    4 for delete_st

 1

Enter student data

Enter st ID : 101
Enter st name :mona

Enter st age : 23
Enter st gradelevel (1,2,3,4) : 3

Enter st sheet mark (~100) : 78

student added


 #MIAN_MENU ..
press
    1 for Add_student
    2 for seacrch_st
    3 for modify_student_name
    4 for delete_st
```

3. Searching function

```
9:26
final worked.c
Saved
Terminal
 #MIAN_MENU ..
press
    1 for Add_student
    2 for seacrch_st
    3 for modify_student_name
    4 for delete_st

 2
want to search
 with Id press 1
with name press 2
1
please enter student ID to search ..
101
ID : 101
Name : mona
age : 23
gradeleve : 3
Sheet mark : 78.00

 #MIAN_MENU ..
press
    1 for Add_student
    2 for seacrch_st
    3 for modify_student_name
    4 for delete_st

 2
want to search
 with Id press 1
with name press 2
2
please enter student name to search ..
heba
Record not Found

 #MIAN_MENU ..
press
    1 for Add_student
```

4.Modifying function

```
9:28

final worked.c
Saved

Terminal

 #MIAN_MENU ..
press
    1 for Add_student
    2 for seacrch_st
    3 for modify_student_name
    4 for delete_st

 3
enter st id
101
Present Name : mona
 enter the new name
>>> Mona_ali

 #MIAN_MENU ..
press
    1 for Add_student
    2 for seacrch_st
    3 for modify_student_name
    4 for delete_st

 2
want to search
 with Id press 1
with name press 2
1
please enter student ID to search ..
101
ID : 101
Name : Mona_ali
age : 23
gradeleve : 3
Sheet mark : 78.00

 #MIAN_MENU ..
```

5.Delete function

```
9:30

final worked.c
Saved

Terminal

 #MIAN_MENU ..
press
    1 for Add_student
    2 for seacrch_st
    3 for modify_student_name
    4 for delete_st

 4
enter st id
 you want to delete his record
101
ID : 0
Name :
age : 0
gradeleve : 0
Sheet mark :  0.00

  Record DELETED


 #MIAN_MENU ..
press
    1 for Add_student
    2 for seacrch_st
    3 for modify_student_name
    4 for delete_st

 2
want to search
 with Id press 1
with name press 2
1
please enter student ID to search ..
101
Record not Found
```

Discussion

The point that is demanded to be discussed is the password's performance

So how the password is working?

> Well the first interface that the user will face is that the program ask him to enter password that already stored while programming. There was no data about what should the password be as a default, so simply we used [***4321***] as a password, the user couldn't change it just if wanted to modify one have to edit the source code .
> So the user will input a value that maybe is true or not , here the program will read it using *scanf* function and restoring it in an integer variable that will be compared with the value that program already knows using *if* function. If *if Condition* is fulfilled, then the program will lead the user to *Main Menu* to make a decision to the next step.
> If the condition is **Not** fulfilled, then the program will display " *sorry! The password is invalid"* and use ***goto line,*** to return to the function of reading the password and re –ask the user to enter the password, here the program will make a counter to calculate how many invalid times the user is made. If the counter is **3** the program will let the MAIN function to *return **0,*** that means the program will be exited.

Conclusion

The main purpose we discuss in this program is simply to make friendly interface using *C Programming language* that makes it easy for the user to enter the data wanted to store and enable him easily to process and modify the students' data.
Using coding to simplify, define a real-life problem and introduce a good example of solving that problem as simplest we can.

References

Array of structures : Structures in c - studytonight

// Used function:
strcmp(str, str); // to compare two string variables.
strump() - programiz

strcpy(str, str); // to copy a string into another one.
strcpy() - programiz

fflush(stdin); //to clear the input buffer
A Question in stackoverflow