****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Faculty of Sciences and Engineering**

**Semester:(Summer,Year:2022),B.Sc. in CSE (Day)**

**Course Title: Data Structure Lab**

**Course Code: CSE 106 Section: 213 DB**

**Lab Project Name: Student Record system**

**Student Details**

|  |  |  |
| --- | --- | --- |
| **Name** | | **ID** |
| **1.** | Rabiul Hasan | 213902072 |

**Submission Date: \_ \_ \_10-09-2022 \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_**

**Course Teacher’s Name: \_** **Farhana Akter Sunny\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_**

**[For Teachers use only: Don’t Write Anything inside this box]**

|  |
| --- |
| **Lab Project Status**  **Marks: ………………………………… Signature: .....................**  **Comments: .............................................. Date: ..............................** |

# Chapter 1 Introduction

## Introduction

The Student Record System is a C mini-project that can help understand the basic concepts of functions, file handling,sorting and data structure. This program will teach how to add, list, change or edit, search and remove data from/to a file. Adding new records, editing and updating them, looking for saved contacts, and removing Student records are simple Functions that make up the main menu of this Student Record program.

Student information, such as name, sub-mark, total, persent, is requested when add a record to your Student Record system . These records can then be updated, entered, searched, and deleted. I’ve used a lot of functions in this mini-project. These functions are easy to understand since their name means just their respective operations.

## Design Goals/Objective

The goal of the project is to design Student Record system :

Create a “Student Record System” framework using C programming. This software is very useful nowadays to store atudent information . The software also has options for removing and changing the data entered.

# Chapter 2

# Implementation of the Project

**Designing modules of Student Record System**

The software architecture consists of the following modules: Preprocessor commands, Functions, Variables, Statements & Expressions.

**Module -1: (Header files)**

* #include<stdio.h>
* #include<stdlib.h>

**Module-2:(declaring functions):**

This function displays the user to select his choice of operations.

**Module-3(main function):**

A function declared in a class named **Student Record System**.

**Module -4(APPEND):**

This module is used for add menu details.

**Module-5(NO OF RECORD**):

This part is to showhow number of recorded file.

**Module -6 (SEARCH) :**

This part is search from saving file

**Module-7 (DELETE) :**

This choice deletes a student information.

**Module-8 (UPDATE) :**

This choice is used to update or alter the information of the record.

**Module-9 (DISPLAY) :**

This choice display the record data.

**Module-10 (SORT TOTAL DESC) :**

This choice total is descending by sort.

1. **Implementation**

C source code

#include<stdio.h>

#include<stdlib.h>

typedef struct student

{

int rno;

char name[20];

struct subject{

int scode;

char name[20];

int mark;

}sub[3];

int total;

float per;

}student;

void create(){

student \*s;

FILE \*fp;

int n, i, j;

printf("Enter rhow many student you want: ");

scanf("%d", &n);

s=(student\*)calloc(n, sizeof(student));

fp = fopen("mystudents.txt","w");

for(i=0; i<n; i++)

{

s[i].total=0;

s[i].per=0;

printf("enter roll number: ");

scanf("%d", &s[i]);

fflush(stdin);

printf("enter name: ");

scanf("%[^\n]s",s[i].name);

for(j=0; j<3; j++)

{

printf("enter marks of subject%d: ", j+1);

scanf("%d",&s[i].sub[j].mark);

s[i].total+=s[i].sub[j].mark;

}

s[i].per=s[i].total/3.0;

fwrite(&s[i], sizeof(student),1,fp);

}

fclose(fp);

}

void display(){

student s1;

FILE \*fp;

int j;

fp = fopen("mystudents.txt","r");

while(fread(&s1,sizeof(student),1,fp))

{

printf("\n%-5d%-20s",s1.rno,s1.name);

for(j=0;j<3;j++)

{

printf("%4d",s1.sub[j].mark);

}

printf("%5d%7.2f",s1.total,s1.per);

}

fclose(fp);

}

void append(){

student \*s;

FILE \*fp;

int n, i, j;

printf("Enter rhow many student you want: ");

scanf("%d", &n);

s=(student\*)calloc(n, sizeof(student));

fp = fopen("mystudents.txt","a");

for(i=0; i<n; i++)

{

s[i].total=0;

s[i].per=0;

printf("enter roll number: ");

scanf("%d", &s[i]);

fflush(stdin);

printf("enter name: ");

scanf("%[^\n]s",s[i].name);

for(j=0; j<3; j++)

{

printf("enter marks of subject%d: ", j+1);

scanf("%d",&s[i].sub[j].mark);

s[i].total+=s[i].sub[j].mark;

}

s[i].per=s[i].total/3.0;

fwrite(&s[i], sizeof(student),1,fp);

}

fclose(fp);

}

void noofrec()

{

student s1;

FILE \*fp;

fp = fopen("mystudents.txt", "r");

fseek(fp, 0 , SEEK\_END);

int n=ftell(fp)/sizeof(student);

printf("\n\n NO OF RECORDS = %d", n);

fclose(fp);

}

void search()

{

student s1;

FILE \*fp;

int j,rno;

fp = fopen("mystudents.txt","r");

printf("enter roll no search: ");

scanf("%d",&rno);

while(fread(&s1,sizeof(student),1,fp))

{

if(s1.rno == rno){

printf("\n%-5d%-20s",s1.rno,s1.name);

for(j=0;j<3;j++)

{

printf("%4d",s1.sub[j].mark);

}

printf("%5d%7.2f",s1.total,s1.per);

}

}

fclose(fp);

}

void update()

{

student s1;

FILE \*fp, \*fp1;

int j,rno, found=0;

fp = fopen("mystudents.txt","r");

fp1 = fopen("temp.txt","w");

printf("enter roll no update: ");

scanf("%d",&rno);

while(fread(&s1,sizeof(student),1,fp))

{

if(s1.rno == rno){

s1.total=0;

s1.per =0;

found =1;

fflush(stdin);

printf("enter new name: ");

scanf("%[^\n]s",s1.name);

for(j=0; j<3; j++)

{

printf("enter new marks of subject%d: ", j+1);

scanf("%d",&s1.sub[j].mark);

s1.total+=s1.sub[j].mark;

}

s1.per=s1.total/3.0;

}

fwrite(&s1,sizeof(student),1,fp1);

}

fclose(fp);

fclose(fp1);

if(found){

fp1 = fopen("temp.txt","r");

fp = fopen("mystudents.txt","w");

while(fread(&s1,sizeof(student),1,fp1)){

fwrite(&s1,sizeof(student),1,fp);

}

fclose(fp);

fclose(fp1);

}

else

printf("\n not found\n");

fclose(fp);

}

void delete\_rec()

{

student s1;

FILE \*fp, \*fp1;

int j,rno, found=0;

fp = fopen("mystudents.txt","r");

fp1 = fopen("temp.txt","w");

printf("enter roll no to delete: ");

scanf("%d",&rno);

while(fread(&s1,sizeof(student),1,fp))

{

if(s1.rno == rno){

found =1;

}

else

fwrite(&s1,sizeof(student),1,fp1);

}

fclose(fp);

fclose(fp1);

if(found){

fp1 = fopen("temp.txt","r");

fp = fopen("mystudents.txt","w");

while(fread(&s1,sizeof(student),1,fp1)){

fwrite(&s1,sizeof(student),1,fp);

}

fclose(fp);

fclose(fp1);

}

else

printf("\n not found\n");

fclose(fp);

}

void sort\_total\_on\_screen(){

student \*s, s1;

FILE \*fp;

int i,j;

fp = fopen("mystudents.txt","r");

fseek(fp,0,SEEK\_END);

int n = ftell(fp)/sizeof(student);

s = (student\*)calloc(n,sizeof(student));

rewind(fp);

for(i=0;i<n;i++)

fread(&s[i],sizeof(student),1,fp);

for(i=0;i<n;i++){

for(j=i+1;j<n;j++){

if(s[i].total<s[j].total){

s1 = s[i];

s[i]=s[j];

s[j]=s1;

}

}

}

for(i=0; i<n; i++){

printf("\n%-5d%-20s",s[i].rno,s[i].name);

for(j=0;j<3;j++){

printf("%4d",s[i].sub[j].mark);

}

printf("%5d%7.2f",s[i].total,s[i].per);

}

fclose(fp);

}

int main()

{

system("color 5f");

int ch;

do{

printf("\n1. CREATE");

printf("\n2. DISPLAY");

printf("\n3. APPEND");

printf("\n4. NO OF RECORDS");

printf("\n5. SEARCH");

printf("\n6. UPDATE");

printf("\n7. DELETE");

printf("\n8. SORT TOTTAL ON SCREEN");

printf("\n0. EXIT");

printf("\nenter your choice: ");

scanf("%d", &ch);

switch(ch)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

append();

break;

case 4:

noofrec();

break;

case 5:

search();

break;

case 6:

update();

break;

case 7:

delete\_rec();

break;

case 8:

sort\_total\_on\_screen();

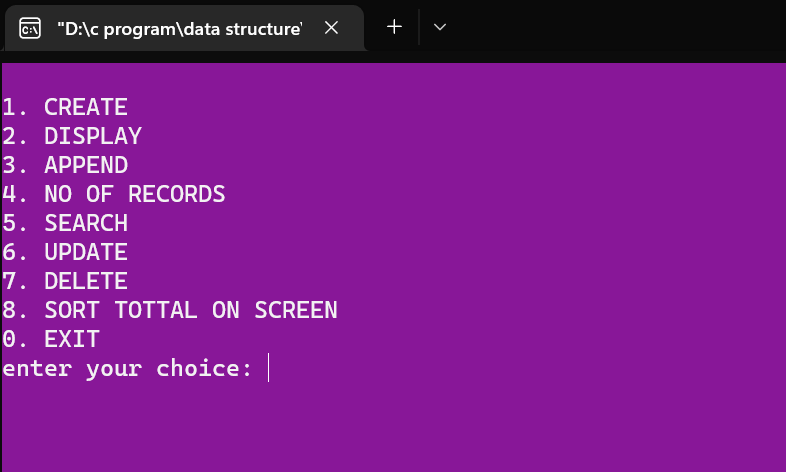
break;

}

}while(ch!=0);

}

**Screenshots**



*Figure 1: Main Menu interface*

When the program is executed, the user will be directed to the main menu interface. The program is introduced with a few lines of texts. Then 9 selections are made for the user as the user can choose to CREATE

2. DISPLAY

3. APPEND

4. NO OF RECORDS

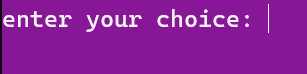
5. SEARCH

6. UPDATE

7. DELETE

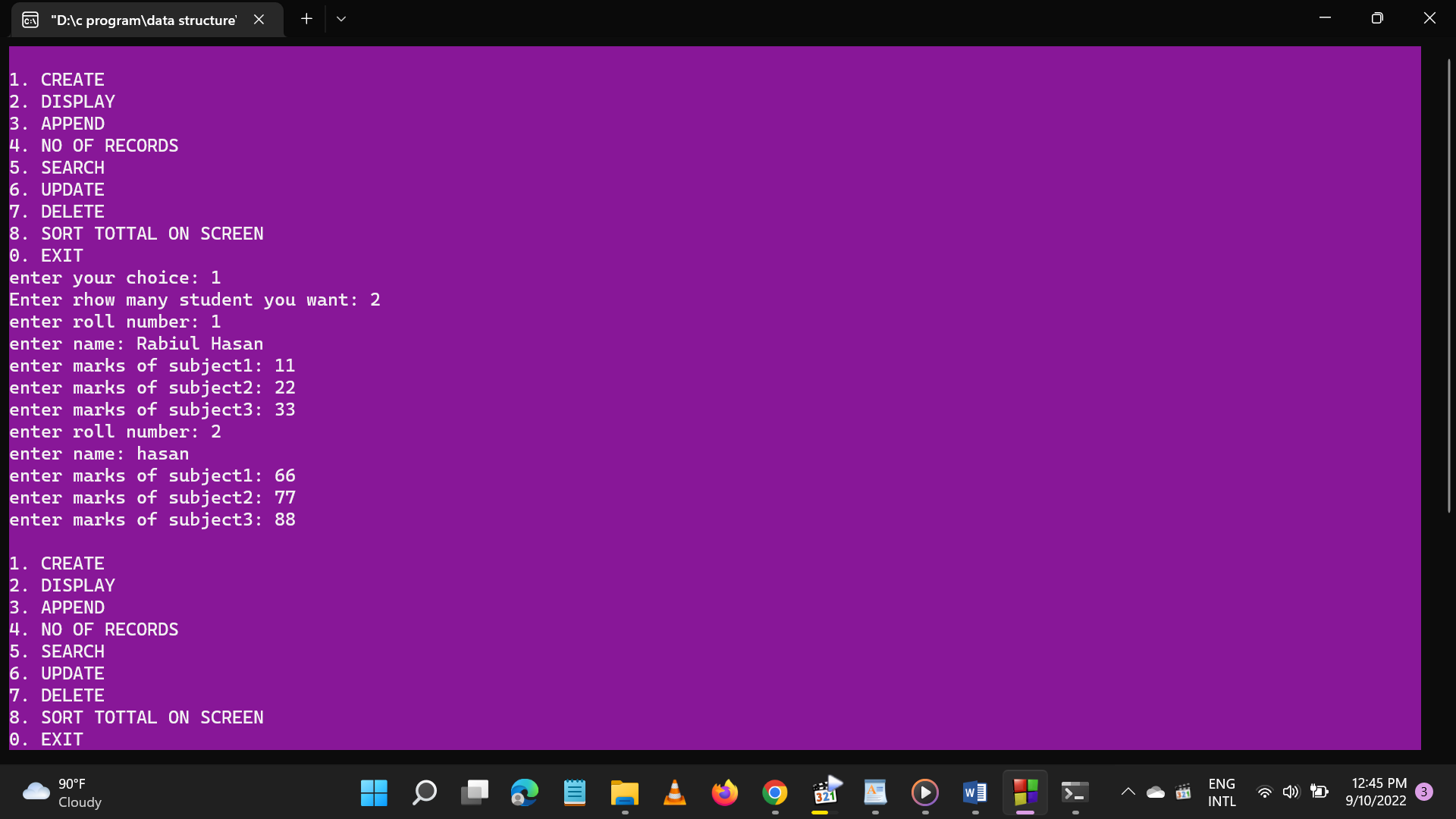
8. SORT TOTTAL ON SCREEN **and**

0. EXIT the program.



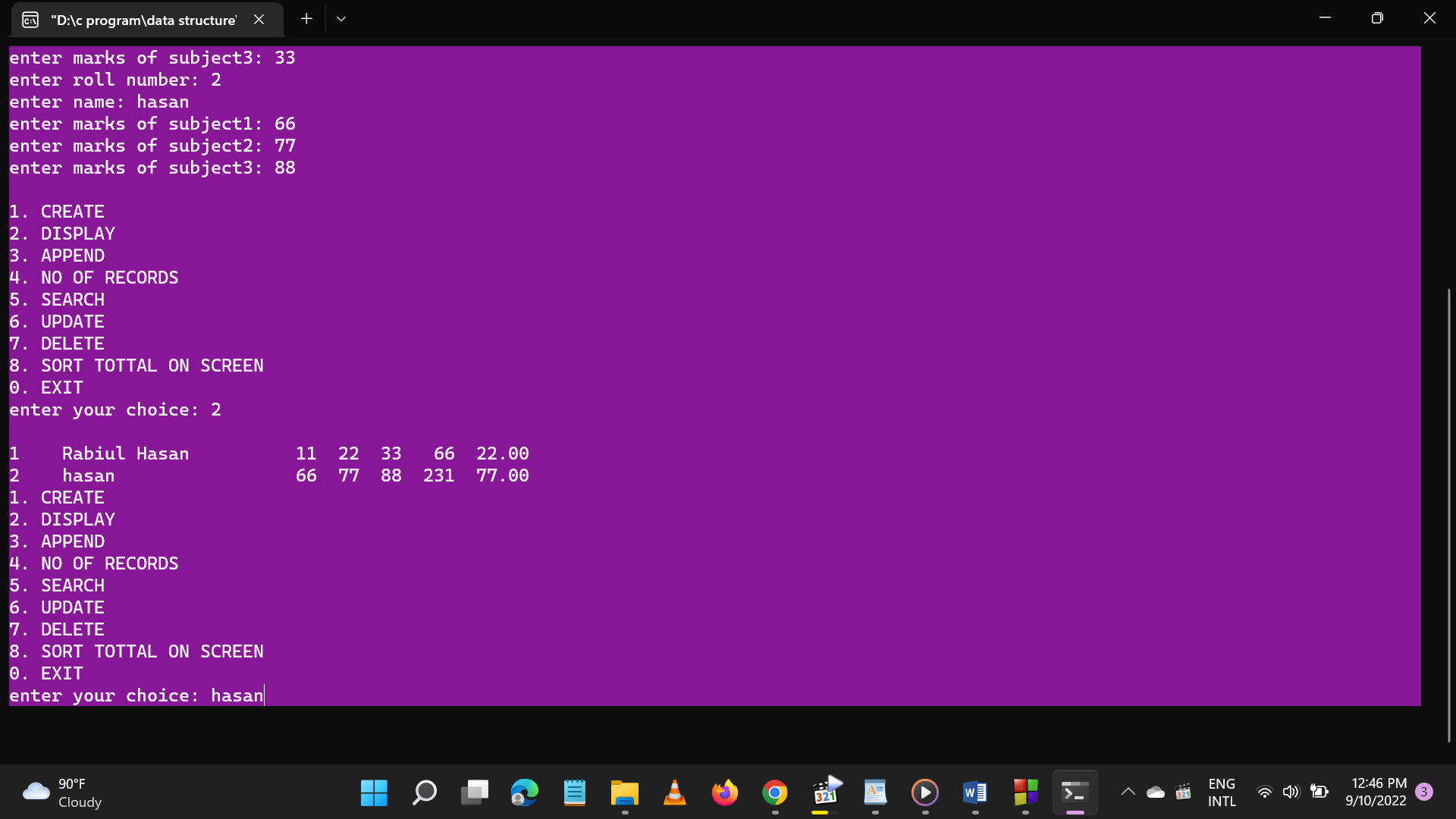
*Figure 2: valid value entered (main menu)*

. If the user accidentally enters an invalid input, an interface will be shown to notify the user to choose again and it notify the user again to enter from 0-8.



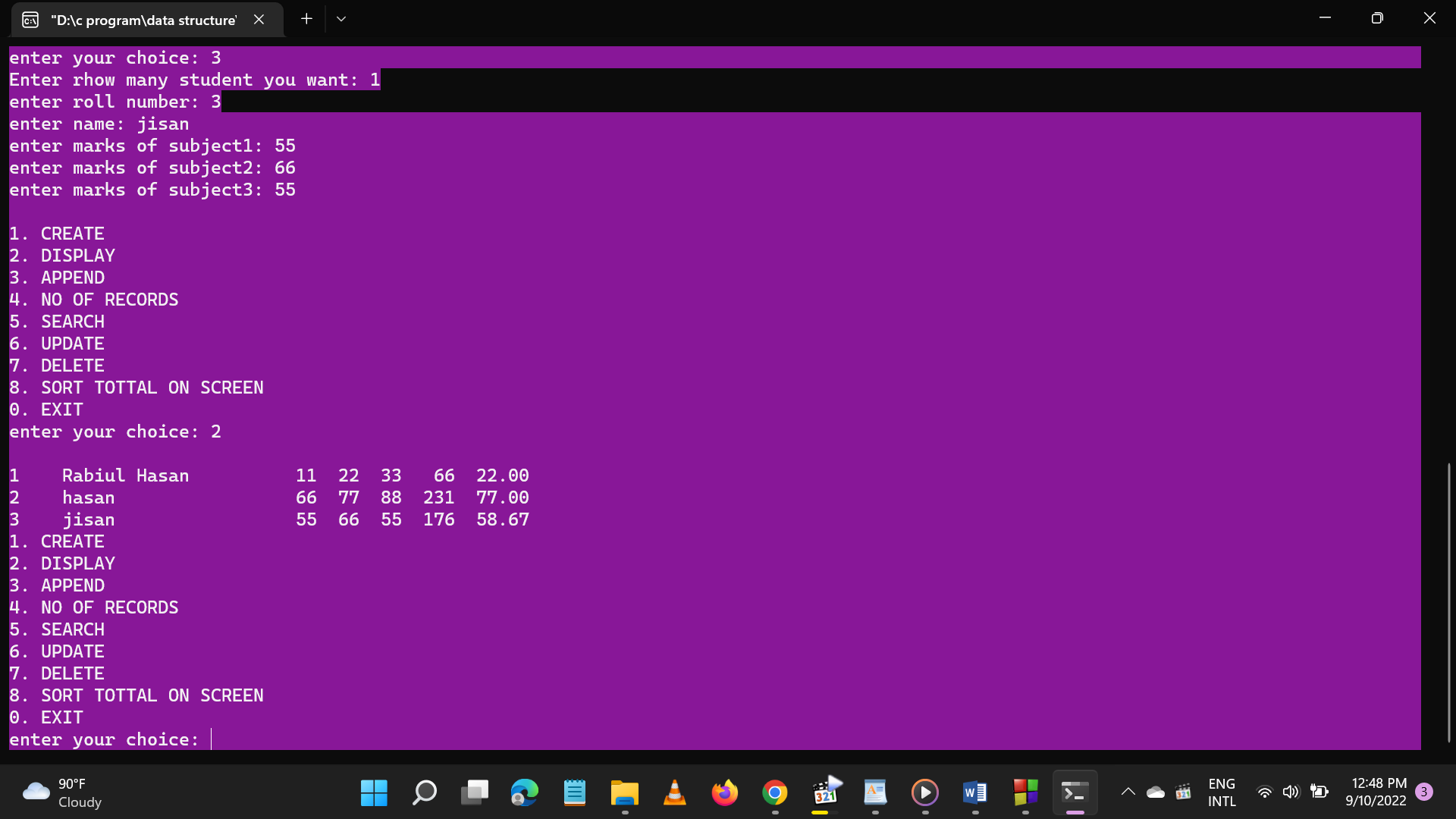
*Figure 3: Create function*

When enter two it display recorded file.



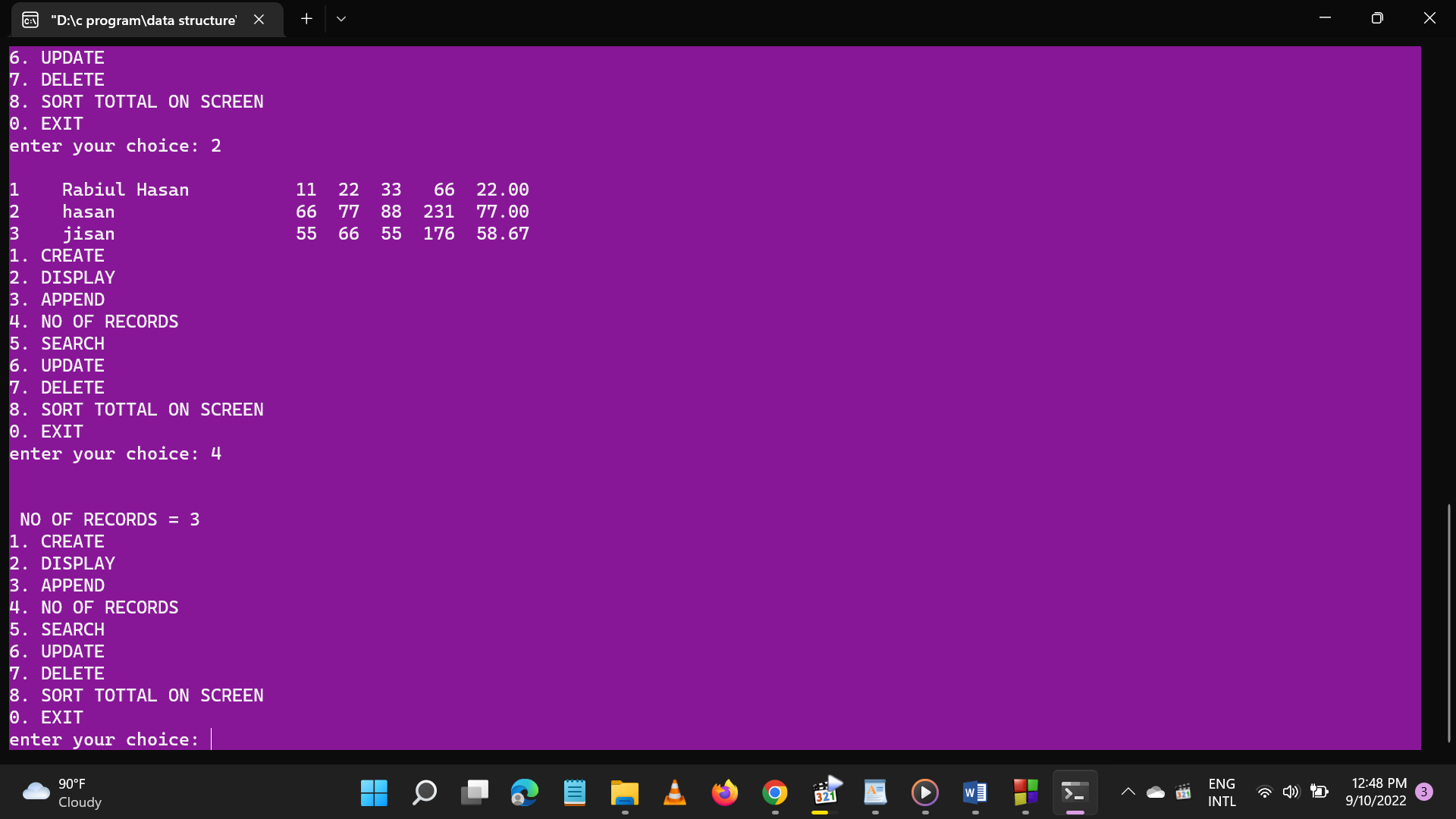
*Figure 4: the seat has successfully recorded*

As shown in the interface the seat information has been record successfully, after the user has entered the details.



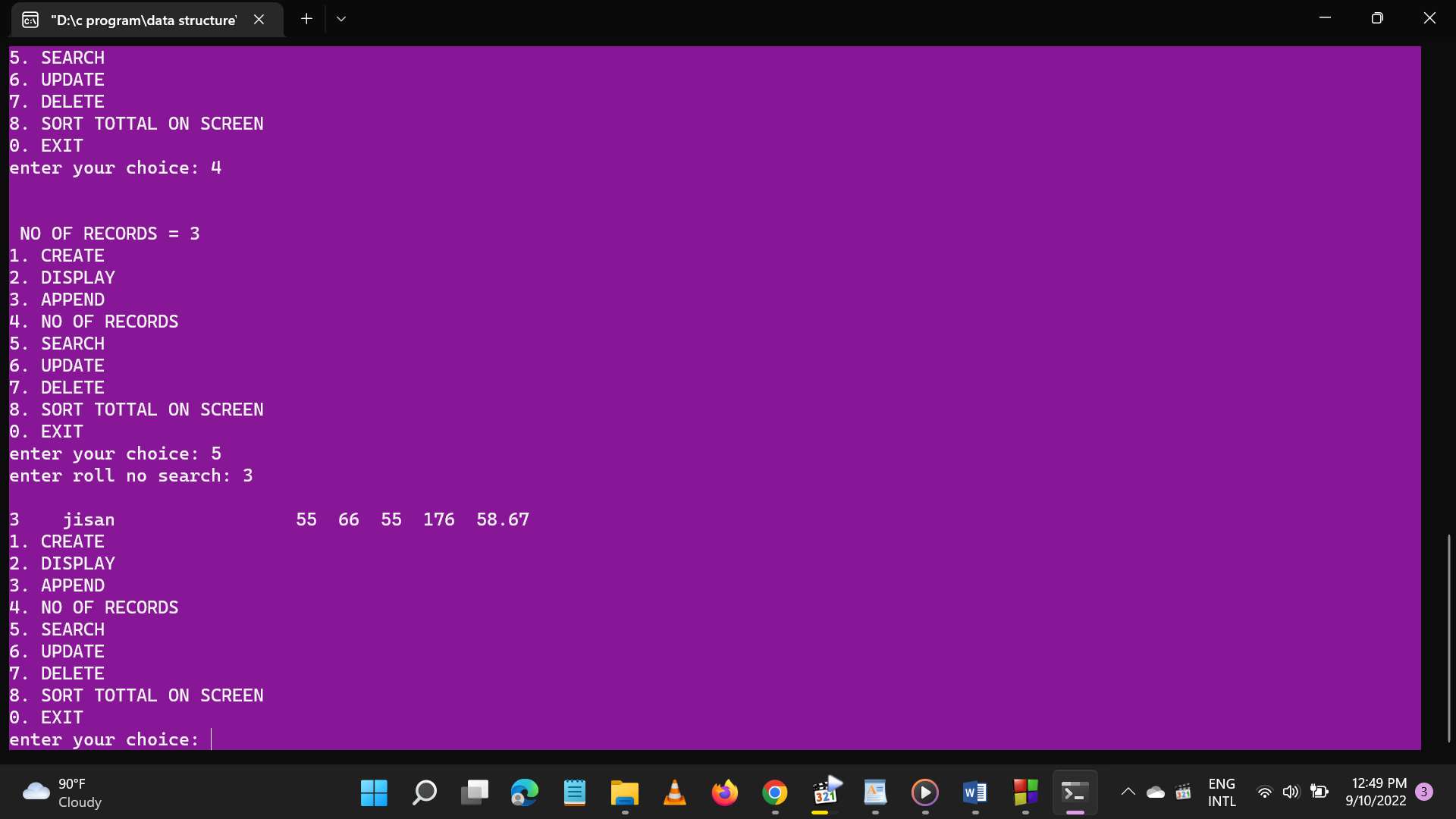
*Figure 5: add new student information*

Search number of records how information records you.

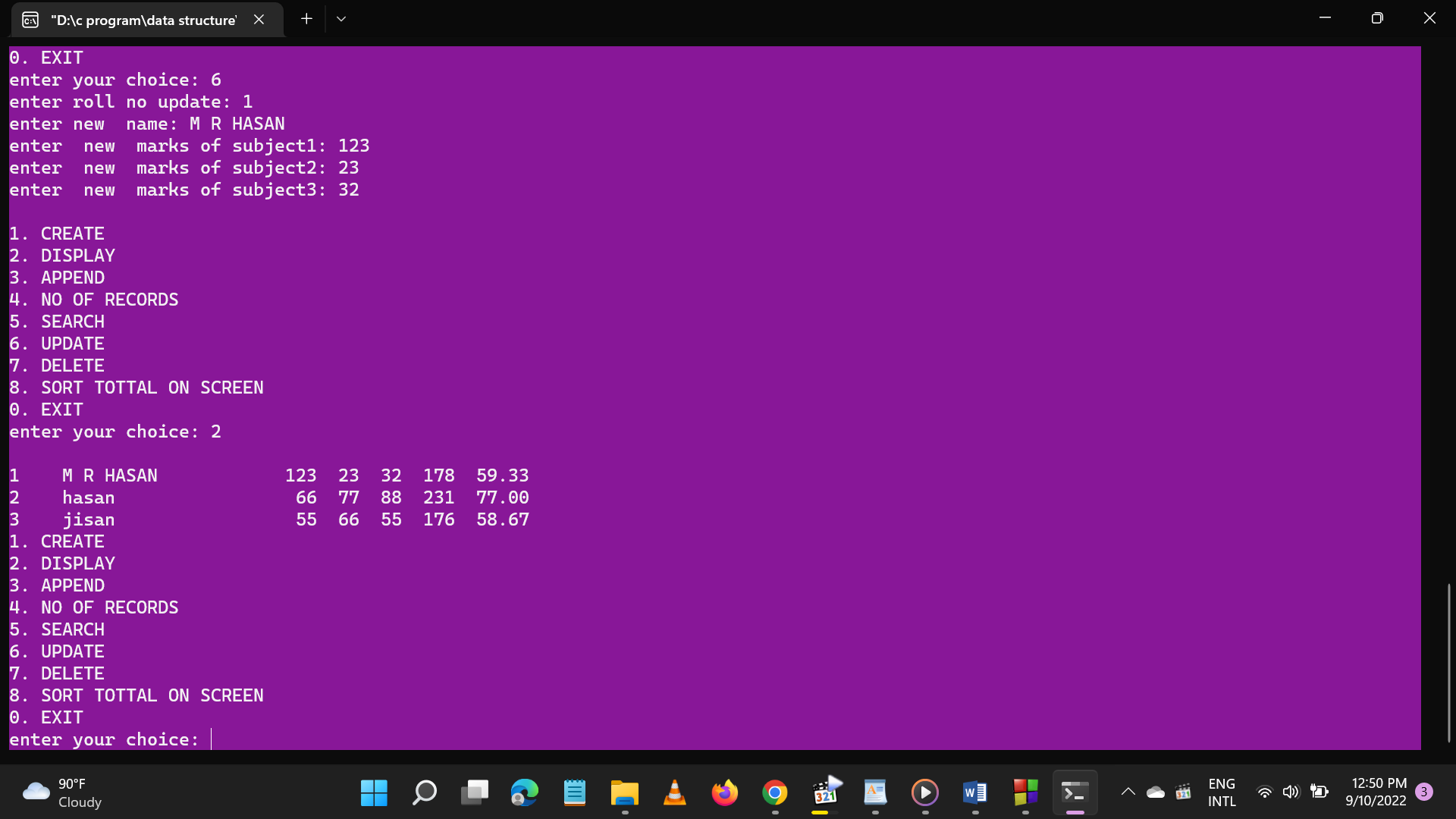


*Figure 6: find number of records*

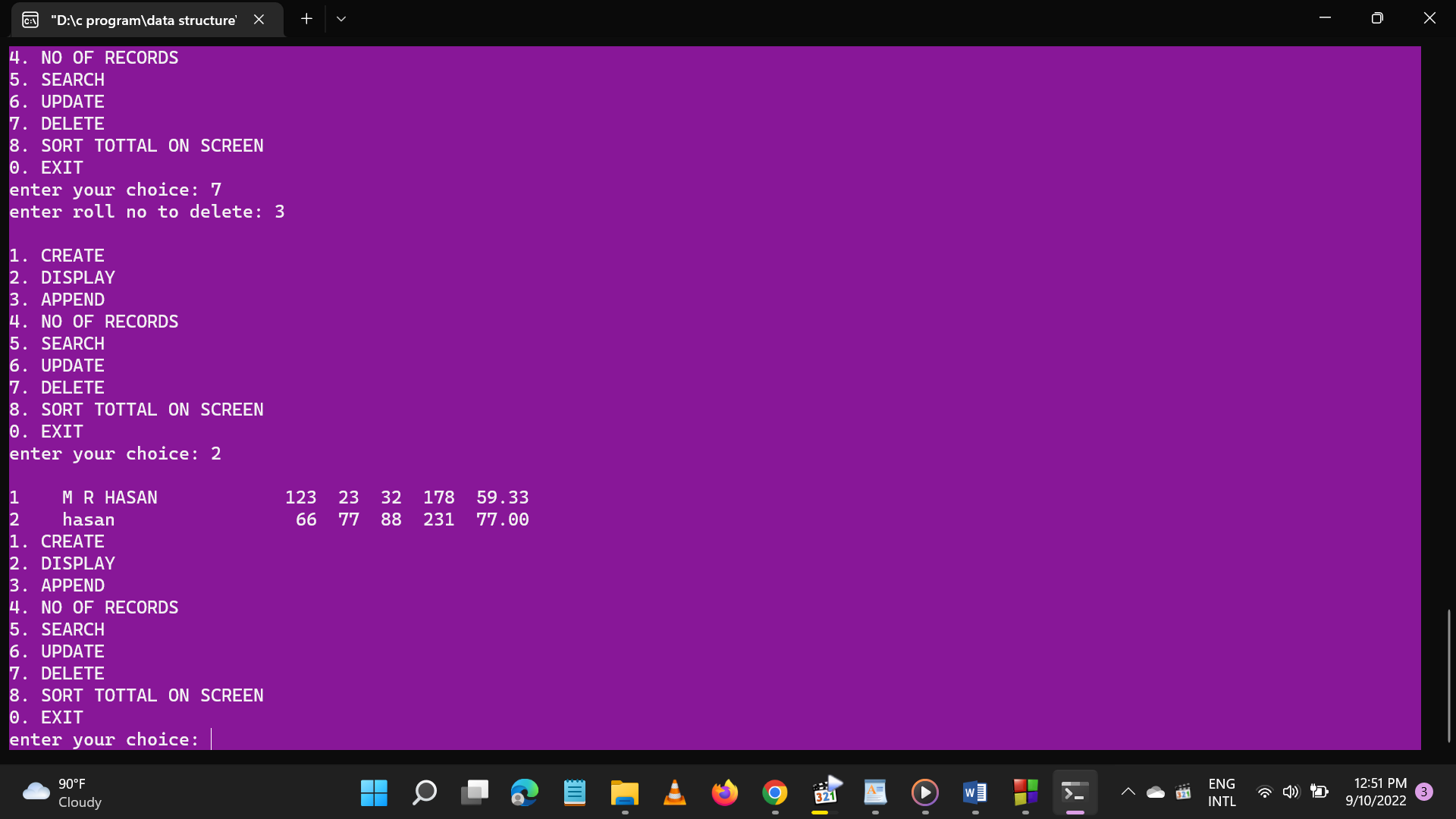
Search whose student information you want



*Figure 7: search*



*Figure 8: update information*



*Figure 9: delete from the information*



*Figure 10: sort on total*

# Chapter 3 Conclusion

**Learning Outcome**

**The Student Record System project** is developing a program which deals with the combination of structures, arrays, File pointers and other functions. This program could do someoperations on arrays such as insertion, deletion, sorting, searching, update, retrieve,merging, append, exit.By implementing this program, we can execute the inserted contact data, deletion ofthe data, searching, updating, append, exit with numbers by using arrays and filepointers. This program is implemented for only numbers that can enter into an array.To do this analysis manually it takes a lot of time and patience but by implementingthis program using a high-level language like C it becomes much easier. But beforegoing to make final solution for the problem, the problem must be analysed.First of all, the basic information regarding the program which consists of complexnumbers. This program is solved by using several methods like one can solve thisprogram using user defined functions concept, loops conditions, go to statements. Inthis abstract we used the concept of functions, while loop, for loop, switch case and ifcondition’s which helps to execute the problem much easier. The following steps arefollowed while implementing the given program using if and while loop.The input is entered i.e., the value of choice (the menu no) select the particularmenu.Next it goes to particular menu and then go to the particular function.It prints the resultant value which came from the execution.The outcome of the work is one can get the required changes like inserting or deletion sorting or merging or append or retrieve or update or exit for a given array.

**Future Scope**

* We can add every data to another person.
* We can also add information two or more person.

# References

* + - 1. Online tutorial [YOUTUBE]