**Ramrao Adik Institute of Technology**

**(Department of Computer Engineering)**

****

**Mini Project Report**

**On**

­Student Details Management

**Subject-: Data Structures**

***Presented By***

|  |  |  |
| --- | --- | --- |
| Roll No | Batch-Sr. No | Name |
|  |  |  |
|  |  |  |
| **18CE2019** | ***A3/6*** | ***GUJAR PARTH SANJAY*** |
|  |  |  |

# CONTENTS OF THE REPORT

|  |  |  |
| --- | --- | --- |
| Sr no. | Content | Page number |
| 1 | Introduction | 3 |
| 2 | Project description | 4 |
| 3 | Algorithm | 5 |
| 4 | Source code | 8 |
| 5 | Outputs | 17 |
| 6 | Conclusion | 22 |

INRODUCTION

**TITLE**: -Student Details Management

**AIM**: - Write a program to enter the details of the Student.ID no, Name of the Student, Current semester of the student, Average pointer of the student, Contact number of the student, Address of the Student. Give option to edit his/her details if required, declare search () to access any student details, display () to display the student details and provide the option to delete the details of the student using delete () and export the data to text file.

**BEST SUITED DATA STRUCTURE**: - Single Linked List.

**JUSTIFICATION**: -A linked list is a complex data structure, especially used in system or applications programming. A linked list is comprised of a series of nodes, each node containing a data element and a pointer to the next node. In linked list the address of first node is important as if that address is lost the whole list is lost. In linked list the last node always contains a NULL pointer showing that list has ended. In real world applications it is mainly useful like-Image viewer, previous and next page in web browser, music player.

# Project Description

**Operations Performed in the Program**

1) void add student ()

It is the function used to add the details of the student. Here we will create a node and will store the details like student name roll no, semester etc in the node. If the list does not contain a single node then first create the node and then store the details or append the node to end of the list.

2) void display ()

This function is use to display the details of all the student in the list. If the list does not contain a single node then Display “No Records Found”.

3) void delete\_student ()

To delete the details of any student, use this function. This function takes roll no. of the student as input. If the list does not contain a single student record then display “No Records Found”. Else run a loop till we reach the last node and compare the value of ID enter by the user if it is equal then delete that student details to which temp is currently pointing. And if user enters invalid ID whose details are not present in our record then display” NO RECORDS FOUND FOR ENTERED ID”.

4) void search\_student ()

To search any particular student’s detail enters his/her ID and search using above function. This function takes the roll no. of the student as input and compares with the roll no. present in each node of the list. If the required student details are found it is displayed or if the record for the entered roll number is not present the “NO DETAILS FOUND” is displayed.

5) void modify\_student ()

In order to edit any student details, use above function. This function takes roll number of the student as input. This function provides facility to modify Student name, address, semester, average pointer. The roll no is key id therefore it cannot be modified.

6) void export()  
This function makes use of File handling in C. This function is used to export the data present in the Linked list to text file. This function provides the ability to store the data in permanent form.

# ALGORITHM

1. Start
2. Define a structure struct details. Declare the variables semester, pointer, contact number , name, address , ID of required data type. Create a pointer variable next.
3. Define a function add\_student() . Create a newnode using malloc(). Then by using gets() and scanf() take input from the user.
4. Check if it is the first node. If yes, the store the address of the newnode in the start variable.
5. Else run a loop till the end of the list and append the newnode in the list.
6. Define a function display()
7. Create a pointer variable \*temp. check (start==NULL) if true then display “NO RECORDS FOUND” else assign start to temp and keep displaying the list till the end of the list is reached.
8. Define a function delete\_student()
9. Create two structure pointer variables \*temp,\*temp1 and char id[20].
10. Check (start==NULL) if true then display “NO RECORDS FOUND”
11. Else enter the student ID,compare that entered ID with every ID stored in the linked list if equal then by using free() delete that node to which temp is pointing currently. Else if user enters invalid ID then display the message “NO RECORDS FOUND FOR ENTERED ID”
12. Define search\_student()
13. Create a structure pointer variable \*temp,

char id[10].

1. Check (start==NULL) display message “RECORDS DO NOT EXSIST”
2. Else enter the ID to be searched. Then compare the entered ID with the ID present in our linked list if match is found then display the details of that student.
3. Define modify\_student()
4. Create structure pointer variable \*temp, char id[20]
5. Check (start==NULL) if true then display “RECORDS DO NOT EXSIST”
6. Else take the input as ID from the user that needs to be modified and compare it with every ID present in the list if match found it means details of the student present at that node needs to be modified.
7. Check whether temp is not NULL, if it is NULL then display “NO RECORDS FOUND FOR ENTERED ID”
8. Else we will use switch case to take input from user. We will provide options to user to edit his/her details:- 1.name , 2.semester , 3.pointer , 4.phone number , 5.address. user has to enter his choice to edit the respective information.
9. Create a export()

* If list is empty display no records to export
* Else using functions of file handling export the data of the linked list to a text file named as Records.

1. Create a main().Give options to the user:-1.add\_new

2.display()

3.delete\_student()

4.search\_student()

5.modify\_student()

6.export()

7.exit()

10. Now by using switch case we will take users choice as input and will perform respective operation.

# Source code

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<string.h>**

**struct details**

**{**

**int semester;**

**float pointer;**

**long int contact\_number;**

**char name[30],address[200],id[10];**

**struct details \*next;**

**}\*start=NULL;**

**void add\_student()**

**{**

**struct details \*newnode, \*temp;**

**int i;**

**newnode=(struct details \*)malloc(sizeof(struct details));**

**fflush(stdin);**

**printf("\nEnter the id of the student:");**

**gets(newnode->id);**

**fflush(stdin);**

**printf("\nEnter the Name of the student:");**

**gets(newnode->name);**

**fflush(stdin);**

**printf("\nEnter the Semester of the student:");**

**scanf("%d",&newnode->semester);**

**fflush(stdin);**

**printf("\nEnter the average pointer of the student:");**

**scanf("%f",&newnode->pointer);**

**fflush(stdin);**

**printf("\nEnter the contact number of the student:");**

**scanf("%10lld",&newnode->contact\_number);**

**fflush(stdin);**

**printf("\nEnter the address of the student:");**

**gets(newnode->address);**

**newnode->next=NULL;**

**if(start==NULL)**

**{**

**start=newnode;**

**}**

**else**

**{**

**temp=start;**

**while(temp->next !=NULL)**

**{**

**temp=temp->next;**

**}**

**temp->next=newnode;**

**}**

**}**

**void display()**

**{**

**struct details \*temp;**

**int i;**

**if(start==NULL)**

**{**

**printf("No records found");**

**}**

**else**

**{**

**temp=start;**

**printf("\nID \t \t NAME \t \t SEMESTER \t \t average POINTER \t \t CONTACT NUMEBER \t ADDRESS\n");**

**printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");**

**while(temp !=NULL)**

**{**

**printf("%s",temp->id);**

**printf(" \t");**

**printf("%s",temp->name);**

**printf(" \t");**

**printf(" %d \t \t \t",temp->semester);**

**printf(" %.02f \t \t",temp->pointer);**

**printf("\t %lld",temp->contact\_number);**

**printf("\t \t ");**

**puts(temp->address);**

**temp=temp->next;**

**}**

**}**

**}**

**void delete\_student()**

**{**

**struct details \*temp,\*temp1;**

**char id[20];**

**if(start==NULL)**

**{**

**printf("RECORDS DONOT EXIT");**

**}**

**else**

**{**

**printf("Enter student ID:");**

**gets(id);**

**if(strcmpi(start->id,id)==0)**

**{**

**temp=start->next;**

**free(start);**

**start=temp;**

**}**

**else**

**{**

**temp=start;**

**while (temp != NULL && (strcmpi(temp->id,id)!=0))**

**{**

**temp1=temp;**

**temp=temp->next;**

**}**

**if(temp==NULL)**

**{**

**printf("\nNO RECORDS FOUND FOR ENTERD ID.");**

**return;**

**}**

**else**

**{**

**temp1->next=temp->next;**

**free(temp);**

**}**

**}**

**}**

**}**

**void search\_student()**

**{**

**struct details \*temp;**

**char id[10];**

**int i;**

**if(start==NULL)**

**{**

**printf("\nRECODRS DO NOT EXSIST");**

**return;**

**}**

**else**

**{**

**printf("\nEnter the id to be searched");**

**gets(id);**

**temp=start;**

**while(temp !=NULL && (strcmpi(id,temp->id) !=0))**

**{**

**temp=temp->next;**

**}**

**if(temp ==NULL)**

**{**

**printf("\nNO RECORDS FOUND FOR ENTERED ID.");**

**return;**

**}**

**else**

**{**

**printf("\nID \t \t NAME \t \t SEMESTER \t \t average POINTER \t \t CONTACT NUMEBER \t ADDRESS\n");**

**printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");**

**printf("%s",temp->id);**

**printf(" \t");**

**printf("%s",temp->name);**

**printf(" \t");**

**printf(" %d \t \t \t",temp->semester);**

**printf(" %.02f \t \t",temp->pointer);**

**printf("\t %lld",temp->contact\_number);**

**printf("\t \t ");**

**puts(temp->address);**

**}**

**}**

**}**

**void modify\_student()**

**{**

**struct details \*temp;**

**char id[20];**

**int i,n;**

**if(start==NULL)**

**{**

**printf("\nRECORDS DONOT EXSIST");**

**return;**

**}**

**else**

**{**

**printf("\nEnter the id to be modified");**

**gets(id);**

**temp=start;**

**while(temp !=NULL && (strcmpi(id,temp->id) !=0))**

**{**

**temp=temp->next;**

**}**

**if(temp ==NULL)**

**{**

**printf("\nNO RECORDS FOUND FOR ENTERED ID.");**

**return;**

**}**

**else**

**{**

**printf("\nEnter 1 to change name.");**

**printf("\nEnter 2 to change semester.");**

**printf("\nEnter 3 to change pointer.");**

**printf("\nEnter 4 to change phone number.");**

**printf("\nEnter 5 to change address.");**

**printf("\nEnter your choice");**

**scanf("%d",&n);**

**switch (n)**

**{**

**case 1:**

**printf("Enter new name:");**

**gets(temp->name);**

**break;**

**case 2:**

**printf("Enter updated semester:");**

**scanf("%d",&temp->semester);**

**break;**

**case 3:**

**printf("Enter new pointer:");**

**scanf("%f",&temp->pointer);**

**break;**

**case 4:**

**printf("Enter updated phone number:");**

**scanf("%10lld",&temp->contact\_number);**

**break;**

**case 5:**

**printf("Enter new address:");**

**gets(temp->address);**

**break;**

**default:printf("INVALID INPUT");**

**break;**

**}**

**}**

**}**

**}**

**void export()**

**{**

**struct details \*temp;**

**FILE \*fp;**

**if(start==NULL)**

**{**

**printf("\nNO RECORDS EXSIST");**

**return;**

**}**

**else**

**{**

**fp=fopen("record.txt","w+");**

**temp=start;**

**fprintf(fp,"\nID \t \t NAME \t \t SEMESTER \t \t average POINTER \t \t CONTACT NUMEBER \t ADDRESS\n");**

**fprintf(fp,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");**

**while(temp !=NULL)**

**{**

**fprintf(fp,"%s",temp->id);**

**fprintf(fp," \t");**

**fprintf(fp,"%s",temp->name);**

**fprintf(fp," \t");**

**fprintf(fp," %d \t \t \t",temp->semester);**

**fprintf(fp," %.02f \t \t",temp->pointer);**

**fprintf(fp,"\t %lld",temp->contact\_number);**

**fprintf(fp,"\t \t ");**

**fputs(temp->address,fp);**

**frpintf(fp,"\n");**

**temp=temp->next;**

**}**

**}**

**}**

**void menu()**

**{**

**int ch;**

**do**

**{**

**printf("\nEnter 1 to Add new student details");**

**printf("\nEnter 2 to Show the list");**

**printf("\nEnter 3 to Delete student details");**

**printf("\nEnter 4 to Search student details");**

**printf("\nEnter 5 to Modify student details");**

**printf("\nEnter 6 to Export details to file");**

**printf("\nEnter 7 to Exit");**

**printf("\nEnter your choice");**

**scanf("%d",&ch);**

**switch (ch)**

**{**

**case 1:**

**add\_student();**

**break;**

**case 2:**

**display();**

**break;**

**case 3:**

**fflush(stdin);**

**delete\_student();**

**break;**

**case 4:**

**fflush(stdin);**

**search\_student();**

**break;**

**case 5:**

**fflush(stdin);**

**modify\_student();**

**break;**

**case 6:**

**fflush(stdin);**

**export();**

**break;**

**case 7:**

**printf("THANK YOU :)");**

**break;**

**default: printf("INALID INPUT");**

**break;**

**}**

**} while (ch != 7);**

**}**

**int main()**

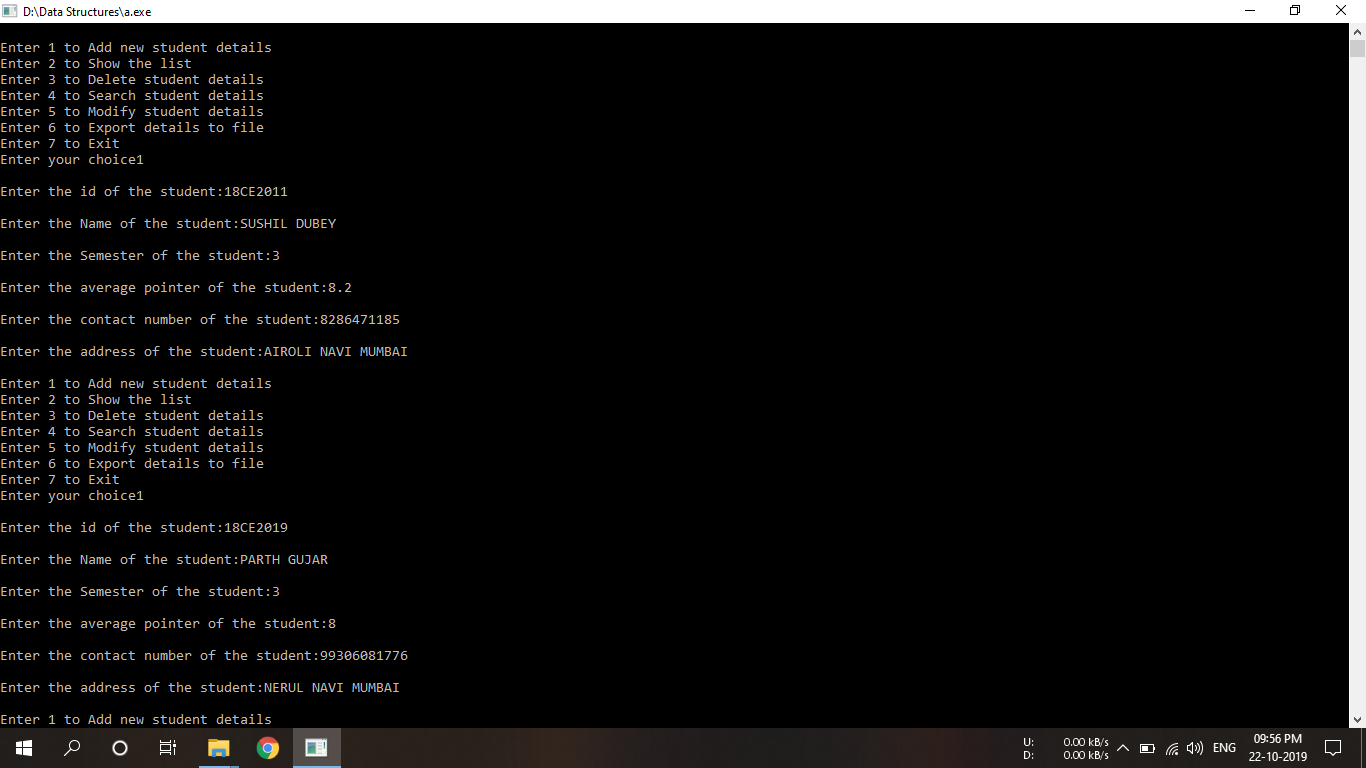
**{**

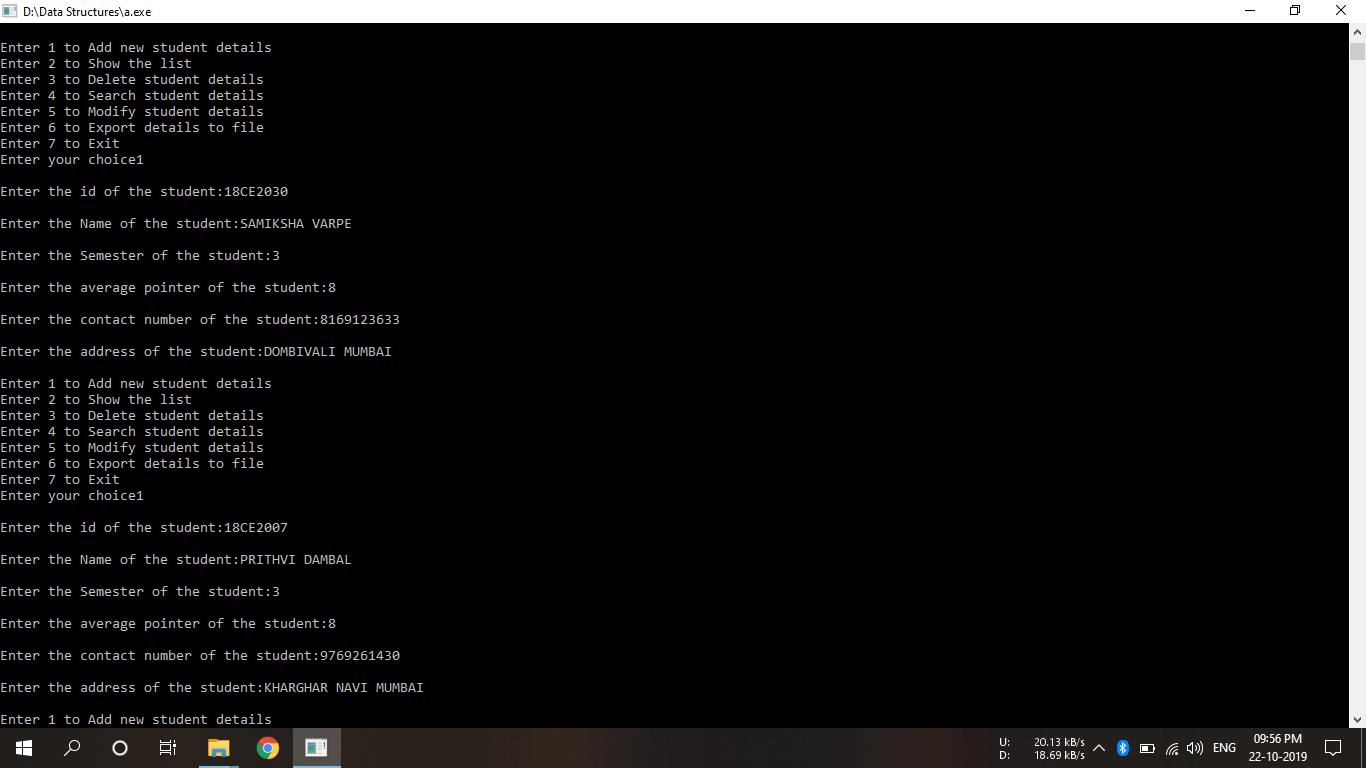
**menu();**

**}**

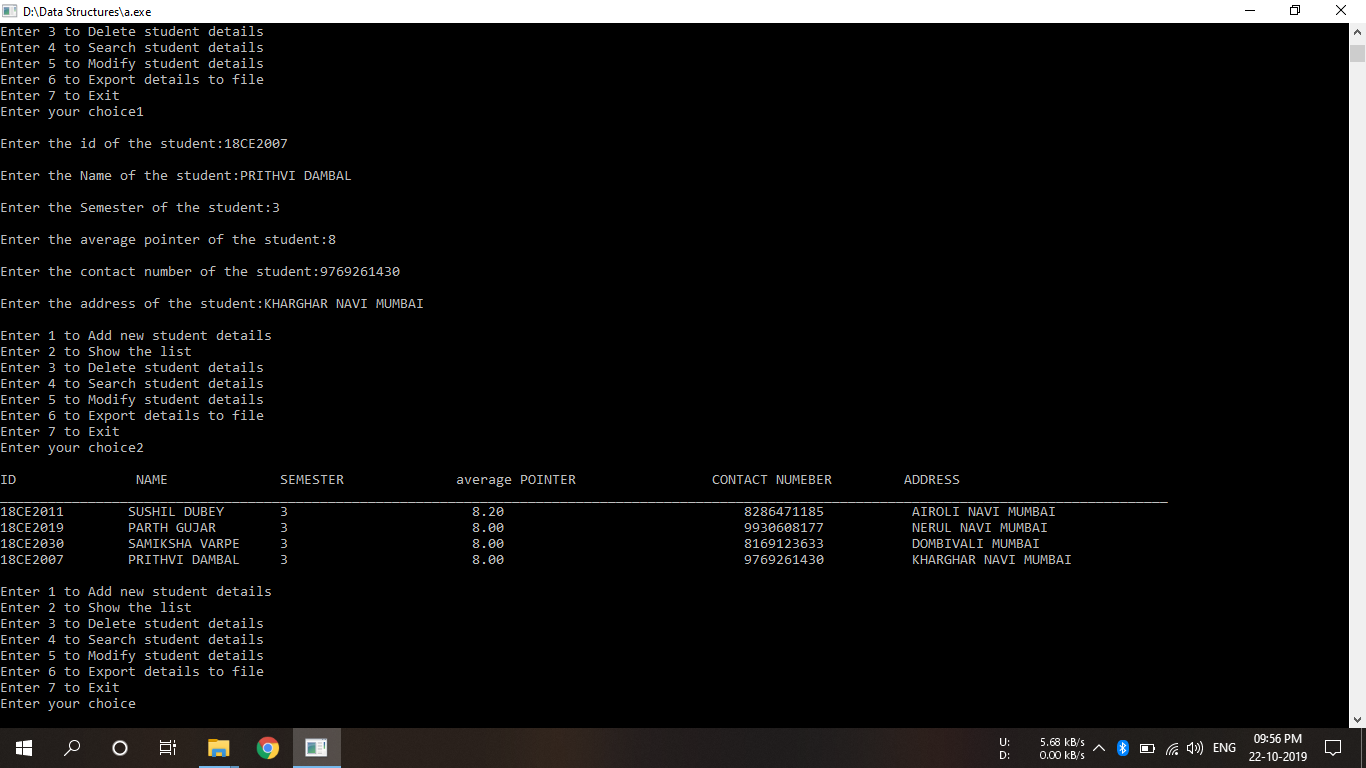
# OUPUTS

# Adding student data

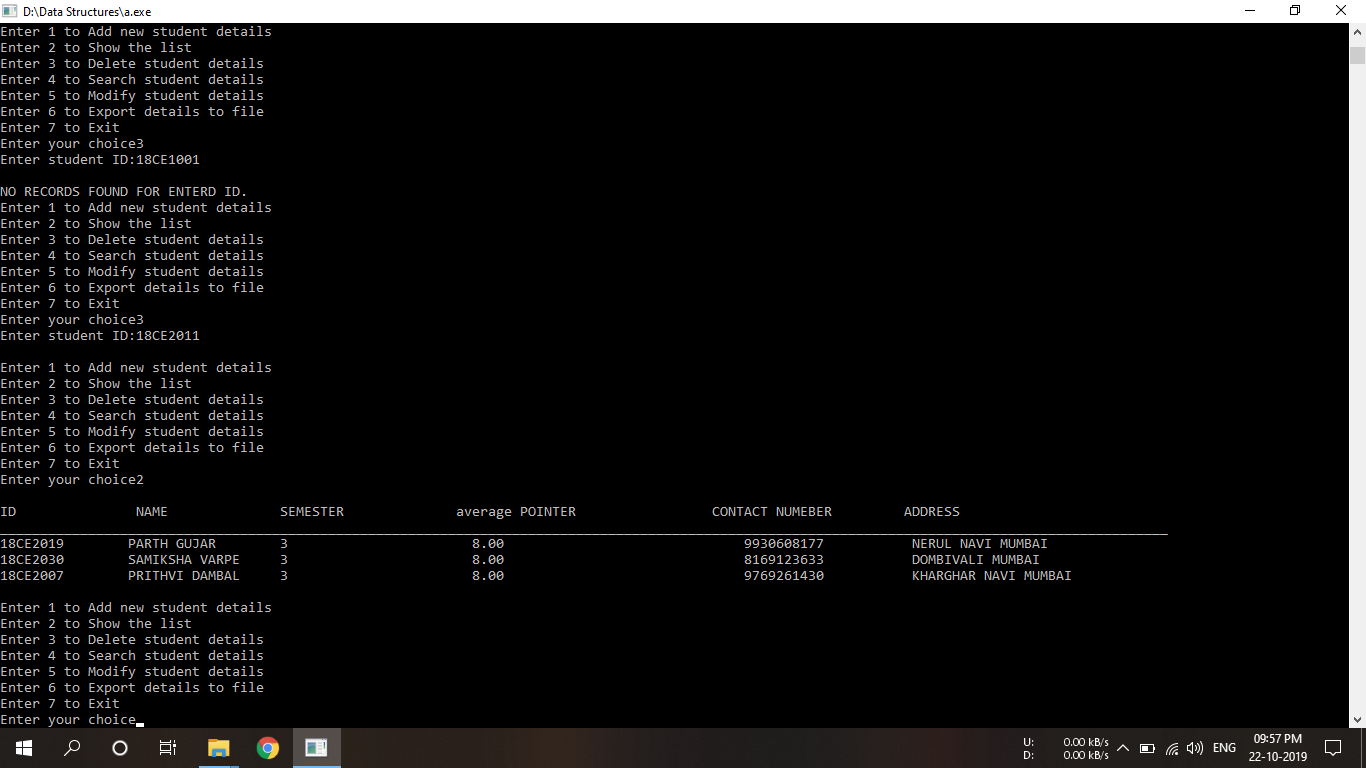


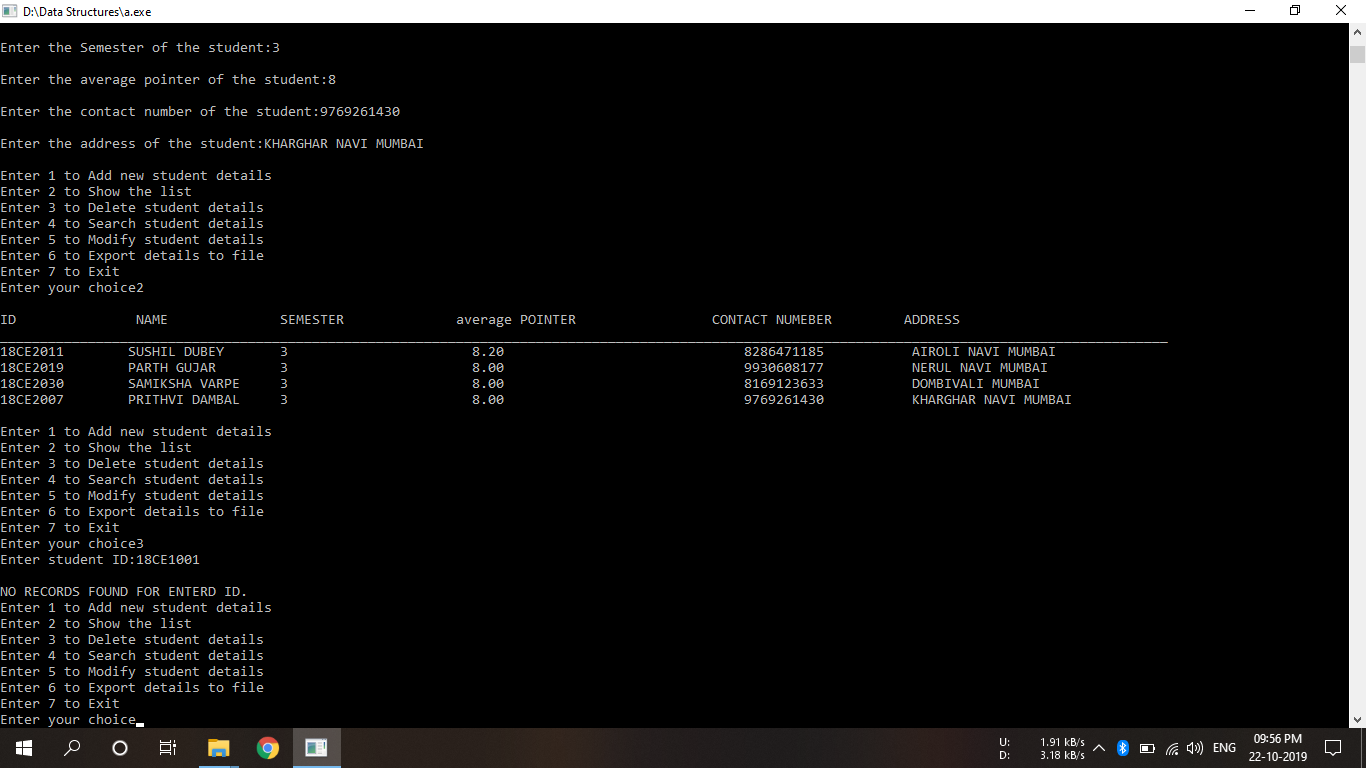


# DISPLAYING STUDENT DATA

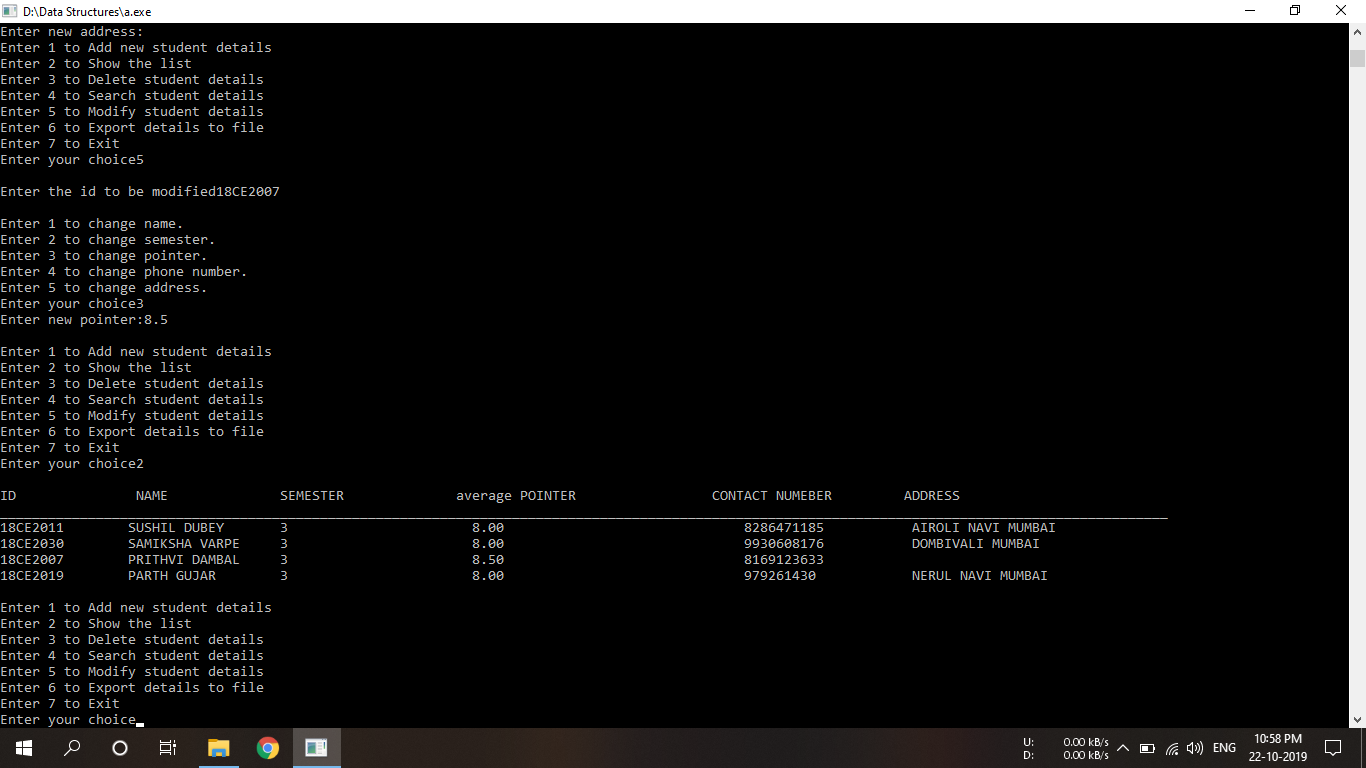


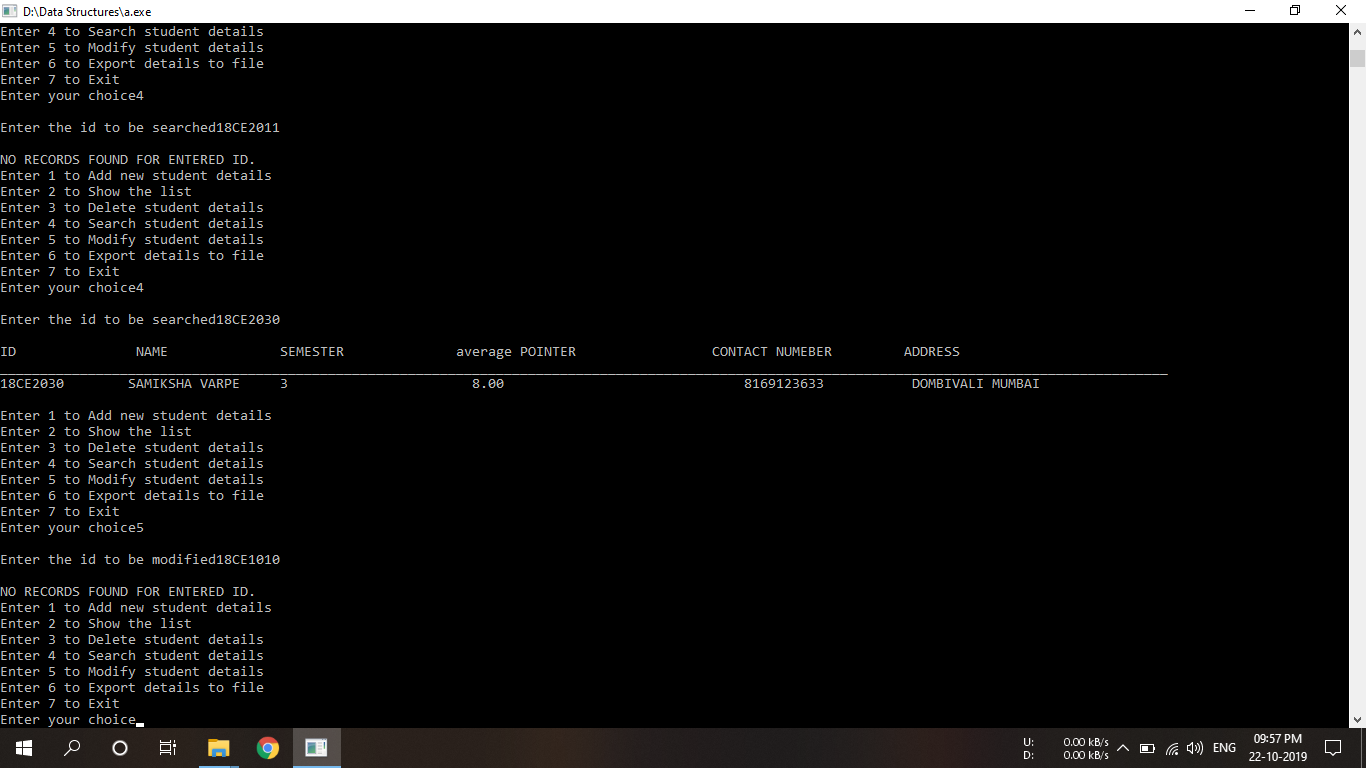
# DELTEING STUDENT DATA



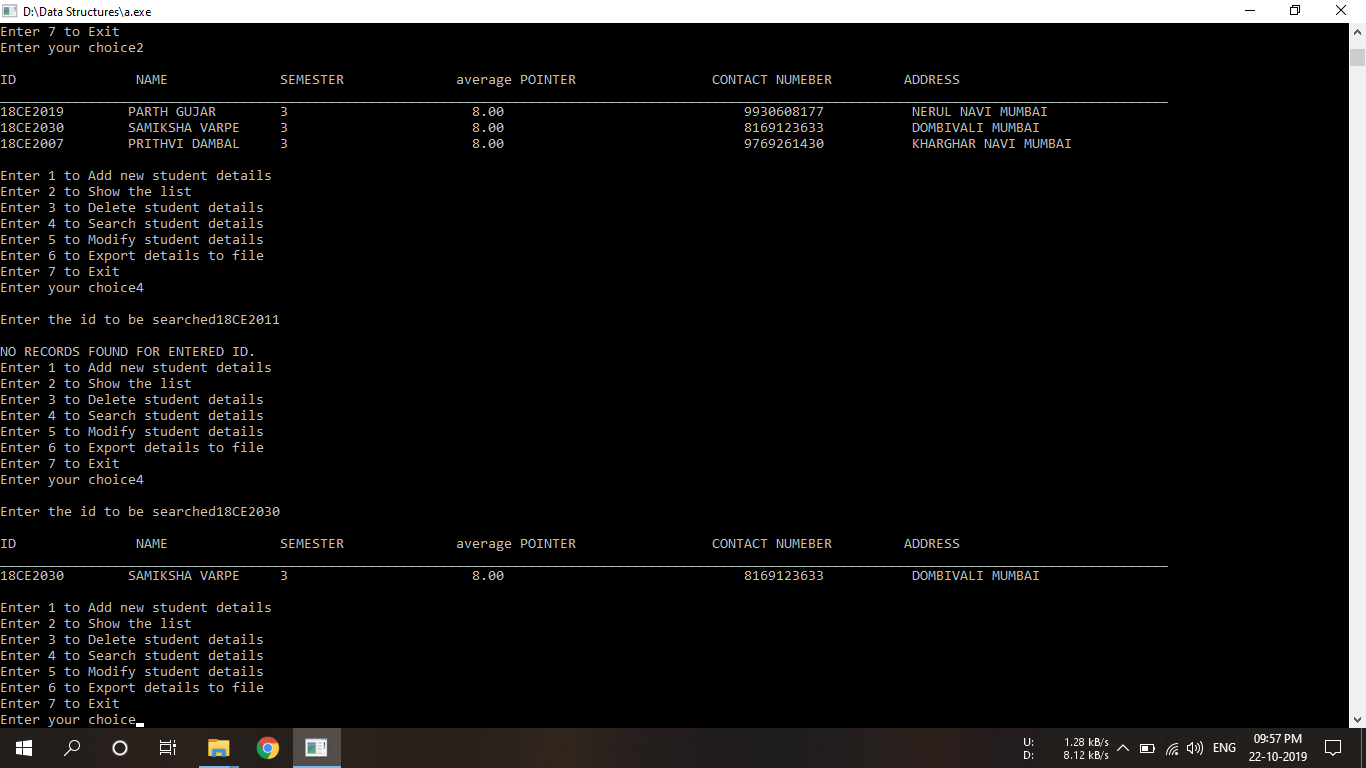


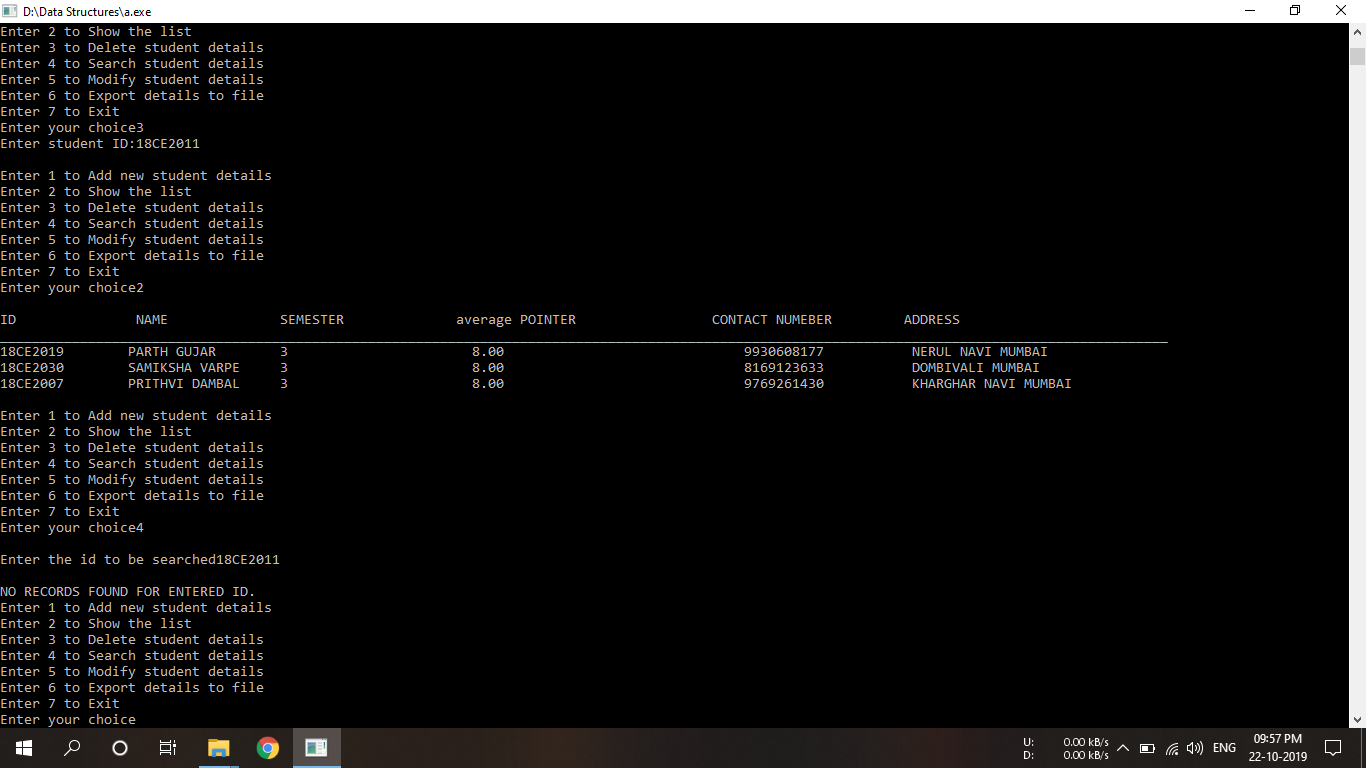
# MODIFY STUDENT DETAILS



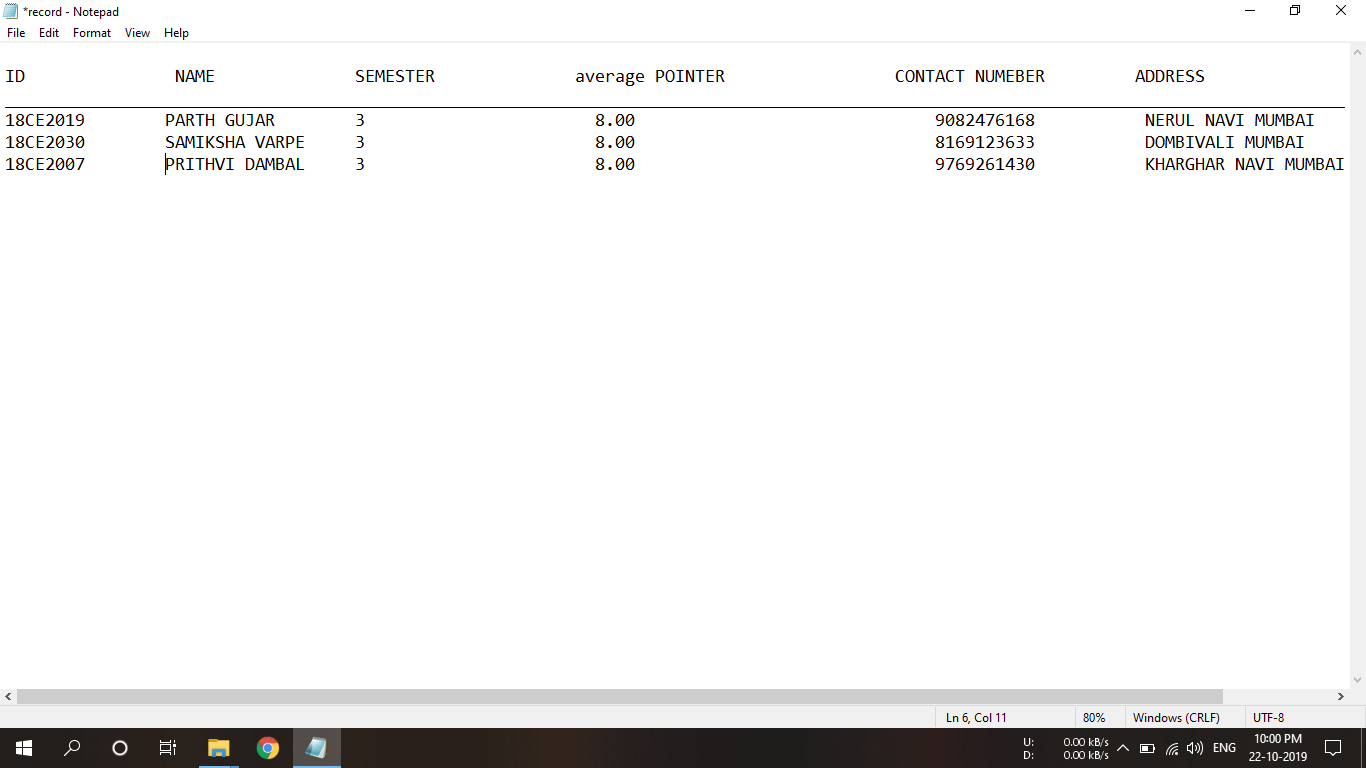


# SEARCH STUDENT





# EXPORT DATA TO TEXT FILE



**CONCLUSION**: - We successfully taken data form user about the students. We have performed various operations like Adding student to the list, displaying the list, deleting a student’s detail form the list, searching for student’s detail in the list, modifying a student data using roll number and successfully exported the data to a text file.