CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

Student database management system is a process that facilities the detailed information of the students. And this project enables teachers or staff members to access the details of the students easily and very clearly.

I choose this topic because storing the information and the details of the student plays a major role in schools and colleges.

1.2 OBJECTIVES

In this project, we have programmed the working and functioning of a student database management.

Student database management system is specially designed for the purpose of accessing the details or information of the student like Name, Roll number and marks of the students very easily. So that, teachers or staff members can anytime check the details of the student.

And whenever there is a parent’s teachers meeting, teachers can easily access the student details and without any complexity they can show the marks of the students to their respective parents.

And this project is useful for printing the progress cards and messaging the marks of the students to the parents.

Whole project is designed in ‘c’ language, different variables have been used for development of the program. It is very easy to operate and understand by the users.

It is just a demonstration of the use of file handling in C language. This project is totally complete and it is error-free.

1.3 METHODOLOGY

1. Enter how many entries do you want to create.

2. Enter the details of the student such as name, roll no. and marks of the student.

The user must enter the details

3.Insert: 1. At the beginning, 2.at the middle, 3.at the end

4. Delete: 1.at the beginning, 2.at the middle, 3. At the end

5. Display: Here, the user must choose the above options after creating the details. If the user chose the insertion at the beginning the details will be created at the beginning. If the user chose the display the details of the students will be displayed.

1.4 EXPECTED OUT COMES

After entering the details of the students such as Name, Roll no. and marks. The user can see the options like 1. Search 2. Insert 3.display 4. Delete 5.exit.

- He can choose only one option, if the user chooses the search option,

  1.The user can search the details of the students by their roll no.
- If the user chooses the insert option,

  1. The user must again choose where they want to insert the details such as

  1.at the beginning 2. At the middle 3. At the end

  2. If the user chooses 1.at the beginning, they can insert at the beginning

  3. If the user chooses 2. At the middle, they can insert the details at the middle.

  4. If the user chooses 3. At the end, they can insert the details at the end.
- If the user chooses the display option,

  1.it will display all the details or information which he has created.

- If the user chooses delete option,
    1. The user must again choose 1. deletion at the beginning 2. At the middle 3. At the end.
    2. If the user chooses deletion at the beginning, it will delete the first students details.
    3. If the user chooses deletion at the middle, it will delete the middle student details.
    4. If the user chooses deletion at the end, it will delete the last student details.
- If the user chooses the exit option,
    1. The user will get out of the screen.

1.5 HARDWARE REQUIRMENTS

• Processor	: Any Processor above 500 MHz
• RAM	: 512Mb
• Hard Disk	: 10 GB
• Input device	: Standard Keyboard and Mouse
• Output device	: VGA and High Resolution Monitor

1.6 SOFTWARE REQUIREMENTS

• Operating system	: Windows XP
• Front End	: ASP.Net 2.0
• IDE	: Visual Studio 2008
• Data Base	: SQL Server Management Studio 2005
• Server	: Internet Information Services
• Database Connectivity	: ODBC Sources (with SQL Server)

CHAPTER 2

DATA STRUCTURES

It is the collection of data values and maintaining a relationship between the data, functions and applying it on the data.

The different data structures are:

- ARRAYS
- STACK
- QUEUE
- LINKED LIST
- TREES
- GRAPH

2.1 ARRAYS

An array is finite collection of similar elements stored in adjacent memory locations.

The number of elements to be used is mentioned as index usually ranging from 0 to N -1 where 0 is lower boundary and N -1 is upper boundary. Arrays can be 1D, 2D or Multi - Dimensional

Syntax:

Data – type   Array – Name[index];

Example:

int A[20];

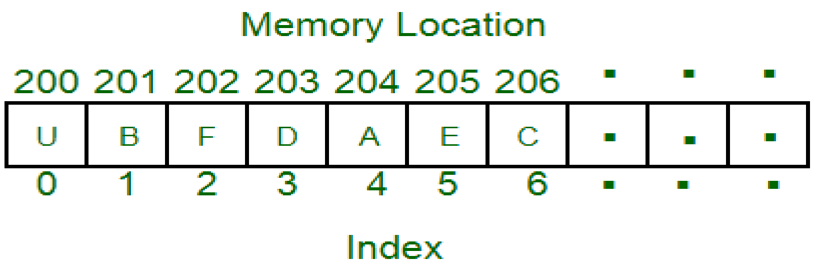


Fig 2.1: ARRAY

2.2 STACK

Stack is a linear data structure where the elements are inserted and deleted from the same end which follows last in first out (LIFO) strategy.

The basic (or) primitive operations of stack are

- Push
- Pop
- Display

Push operation: Inserting an element into to the top of the stack is called push. If we try to insert more elements than the maximum size of the stack then the condition is called stack over flow.

Pop operation: Deleting an element from the stack is called pop. If the stack is empty, but if we still try to delete elements, then it leads to stack under floe condition.

Display operation: Printing all the elements in the stack is called display.

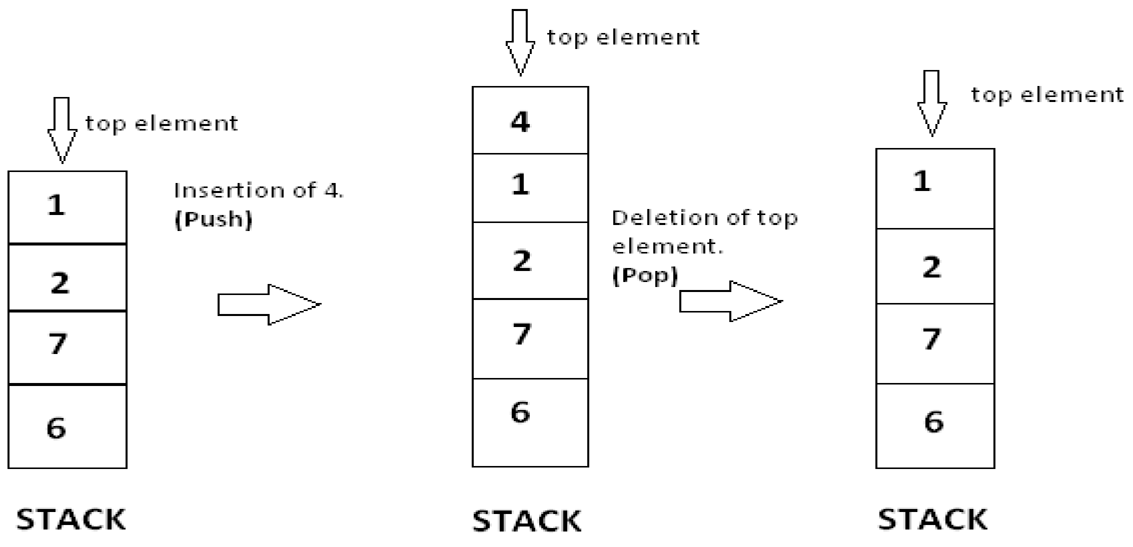


Fig 2.2: STACK

2.3 QUEUE

A QUEUE is an ordered collection of data such that data is inserted at one end and deleted from another end. Queue follows first in first out (FIFO) order.

The main queue operations are:

- Insertion
- Deletion
- Display

Insertion: The addition of new element to the queue is called insertion and every time before inserting a new item rear is incremented first then the item is inserted.

The condition to be checked in insertion is queue overflow where rear reaches maximum size.

Deletion: The removal of front element from the queue is called deletion and after deleting the element we need to increment the front pointer.

The condition to be checked in deletion is queue under flow i.e. , no items are present in the queue.

Display:

Printing all the elements from front to rear is called display

FIGURE 1.3

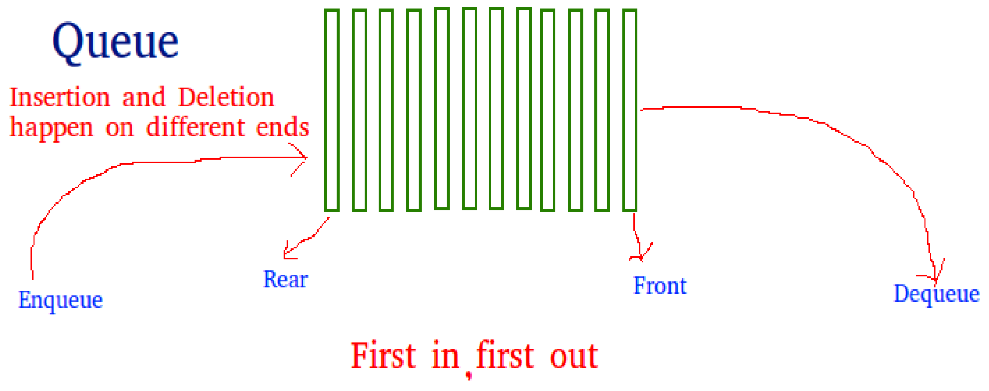


Fig 2.3: QUEUE

There are different types of queues:

- Linear queue
- Circular queue
- Double ended queue
- Priority queue

Linear queue:

It is a type of queue in which the data is added at one end called rear end and deleted at other end called front end.

Circular queue:

It is a type of queue in which all locations are treated as circular such that the first location follows the last location.

Double ended queue:

It is a type of queue in which insertion and deletion can happen at both ends of the queue

Priority queue: It is type of queue which follows FIFO order but un priority elements are moved to last.

2.4 LINKED LIST

A linked list is a collection of nodes where each node as its data segment and pointers pointing to other nodes in sequence.

TYPES OF LINKED LIST:

1. Single linked list:

It is the collection of nodes where each node as data part and a pointer pointing to its immediate adjacent node. It can be traversed only in one direction.

```
// A linked list node

struct Node {

  int data;

4   struct Node* next;

};
```

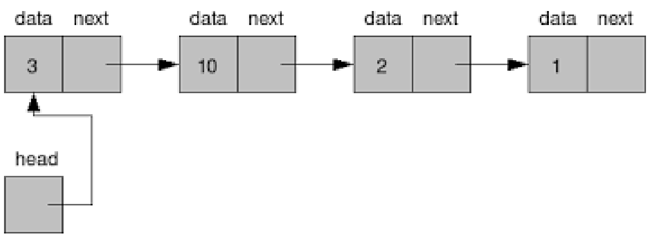


Fig 2.4(a): LINKED LIST

2. Double linked list:

It is the collection of nodes where each node as data part and one pointer pointing to its immediate next adjacent node and other pointer pointing to previous node .it can be traversed both forward and backward direction.

```
/* Node of a doubly linked list */

struct Node {

   int data;

   struct Node* next; // Pointer to next node in DLL

    struct Node* prev; // Pointer to previous node in DLL };
```

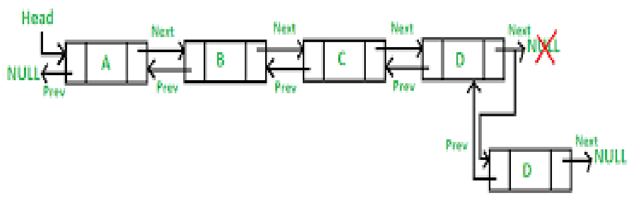


Fig 2.4(b): DOULE LINKED LIST

3. Circular single linked list:

It is the collection of nodes where each node as data part and a pointer pointing to its immediate adjacent node. The last node points back to the head node .it can be traversed only in one direction.

```
typedef struct Node


{

        int info;

        struct Node *next;
}node;


node *front=NULL,*rear=NULL,*temp;
```

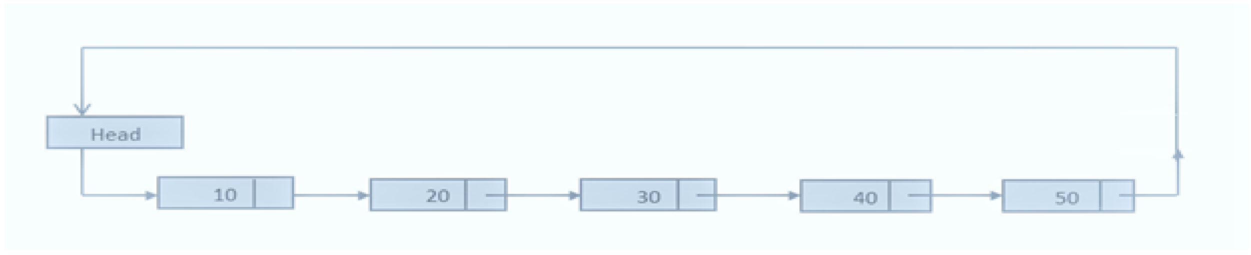
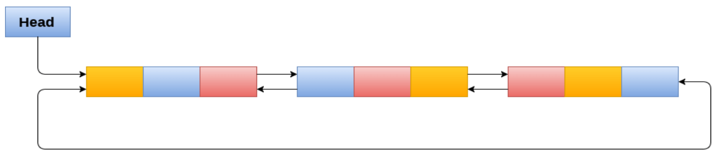


fig 2.4(c): CIRCULAR LINKD LIST

4. Circular double linked list:

It is the collection of nodes where each node as data part and one pointer pointing to its immediate next adjacent node and other pointer pointing to previous node .The last nodes next pointer is pointed back to the head node. It can be traversed both forward and backward direction.

```
struct node

{

   struct node *prev;

   struct node *next;

   int data;

};
```



Circular Doubly Linked List

Fig 2.4(d): CIRCULAR DOUBLE LINKED LIST

5. Header linked list:

The header linked list consists of header which contains special information like number of nodes, address of last node.

```
struct node

{

int data;

struct node *next;

};
```

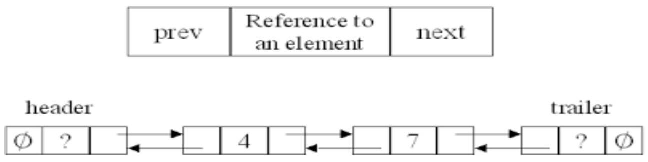


Fig 2.4(e): HEADER LINKED LIST

2.5 TREES

It is a collection of nodes in a hierarchical manner .It can be represented using linked list and arrays .Its types include

1e. EXPRESSION TREE

2. HEAP TREE

3. BINARY SEARCH TREE

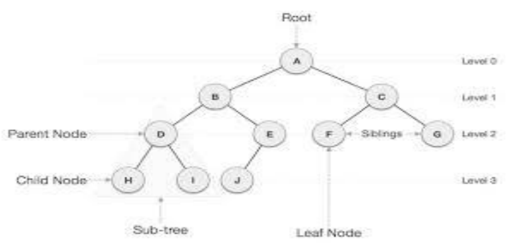


Fig 2.5: TREE

2.6 GRAPH

Graph is a data structure which two components nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph. More formally a Graph can be defined as Graph consists of a finite set of vertices (or nodes) and set of Edges which connect a pair of nodes.

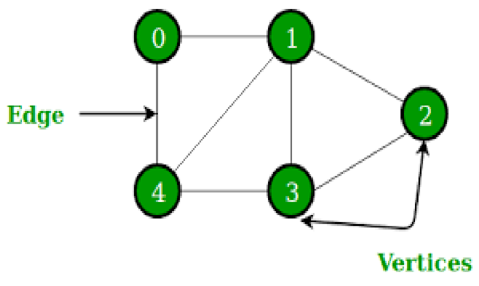


Fig 2.6: GRAPHS

CHAPTER 3

ALGORITHM

Step 1: insert and declare all header files which are required for this project

Step 2: and declare all variables which were required for project

Step 3: and then declare the data types

Step 4: Next we use display function for displaying the Roll number, Name and marks of the students in respective subjects and internals.

Step 5: And then by using create function we are going to enter the Roll number, Name, and marks of the students in respective subjects and internals.

Step 6: In the same way, by using insert function we are going to insert the Roll number, Name and Marks of the students in respective subjects and internals.

Step 7:similarly, by using delete function we are going to delete the details or information of the student by using the roll number.

Step 8: Lastly, by using search function we are going to search the details or information of the student by using roll number

CHAPTER-4

IMPLIMENTATION

Display Here, the display function displays the roll no. , Name and marks of the student.

```
void Display(Node *head)
{

        Node *p;
        int i;

        if(head==NULL)
        {
                printf("There is no records in database");

        }

        else
        {
                p=head;
                for(i=0;i<80;i++)
                {

                        printf("-");

                }

                printf("Updated Student Database");
                for(i=0;i<80;i++)
                {
```

```
                printf("-");

        }
        printf("");
        printf("Roll No.\n");
        printf("name\n");
        printf("internal1 and iternal2 marks of subject1 and subject2 \n");
        for(i=0;i<80;i++)
        {

                printf("-");

        }
        printf("");
        while(p!=NULL)
        {

                printf("%d                    %s          %0.2f%0.2f%0.2f %0.2f",p->roll,p-
>name,p->subject1, p->subject2,p->internal1,p->internal2);
                printf("");
                for(i=0;i<80;i++)
                {

                        printf("-");

                }
                printf("");

                flushall();
                p=p->next;
            }

        }

}
```

CREATE : Here, the create function creates the details of the students. As how many entries the user wants to create.

```
Node* Create(Node *head)
{
        int n,i;
        Node *nn,*p;

        printf("How many Entries to Create Database???");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
            if(head==NULL)
            {

                nn=(Node*)malloc(sizeof(Node));

        printf("Enter Roll No:\n Name\n internal1 and internal2 marks of subject1 and subject2\n");
                scanf("%d",&(nn->roll));
                flushall();

                gets(nn->name);

                scanf("%f",&(nn->subject1));
                scanf("%f", &(nn->subject2));
                scanf("%f",&(nn->internal1));
                scanf("%f",&(nn->internal2));

                nn->next=NULL;

                head=nn;

        }
        else
        {
                p=nn=head;

                while(nn->next!=NULL)
                {
                        nn=nn->next;
                }
                nn->next=(Node*)malloc(sizeof(Node));
                nn=nn->next;

printf("Enter Roll NO:\n Name\n internal1 and internal2 marks of Subject1 & Subject2\n");
                scanf("%d",&(nn->roll));
                flushall();

                gets(nn->name);
                scanf("%f",&(nn->subject1));
                scanf("%f", &(nn->subject2));
                scanf("%f",&(nn->internal1));
                scanf("%f",&(nn->internal2));

                nn->next=NULL;
```

```
                    nn=p;
                }

        }

                return head;

}
```

INSERT: here, the insert function can insert the details of the student at the beginning, at the middle, at the end. As where the user wants to insert.

```
Node* Insert(Node *head)
{
        int ch,r;
        char ans;
        Node *p,*nn,*q;
        do
        {

                printf("Where do you want to enter new entry???");

                printf("1.At the Begining2.At the middle3.At the end");
                printf("Enter your choice:");
                scanf("%d",&ch);
                switch(ch)
                {
                case 1:

                    p=head;

                    nn=(Node*)malloc(sizeof(Node));
```

```
printf("Enter Roll NO., Name,internal1 and internal2marks of subject1 & subject2");
                scanf("%d",&(nn->roll));
                flushall();

                gets(nn->name);
                scanf("%f",&(nn->subject1));
                scanf("%f", &(nn->subject2));
                scanf("%f",&(nn->internal1));
                scanf("%f",&(nn->internal2));

                nn->next=NULL;
         nn->next=p;
         head=nn;

                printf("Entry is Created successfully");
                Display(head);

         break;

                case 2:


                        if(head==NULL)
                        {

                                printf("Yet database is not created.");
                                printf("Database is empty.");
                                printf("First Create Database.");
```

```
            }

            else
            {

                    printf("After which Roll NO. You want to insert new Data???");
                    scanf("%d",&r);

                    p=head;

                    while(p->roll!=r && p->next!=NULL)
                    {
                            p=p->next;

                    }

                    if(p->roll!=r)
                    {
                            printf("There is no such entry");

                    }

                    else
                    {

                            nn=(Node*)malloc(sizeof(Node));


    printf("Enter Roll NO.,Name,internal1 and internal 2 marks subject1 & subject2");
                            scanf("%d",&(nn->roll));
                            flushall();
                            gets(nn->name);

                            scanf("%f",&(nn->subject1));
                            scanf("%f", &(nn->subject2));
                            scanf("%f",&(nn->internal1));
                            scanf("%f",&(nn->internal2));

                            nn->next=NULL;
                            q=p->next;
                            p->next=nn;
                            nn->next=q;

                            printf("Entry is Created successfully.");
                            Display(head);
                    }
            }

        break;

        case 3:

            if(head==NULL)
            {

                    printf("Yet database is not created.");
                    printf("Database is empty.");
                    printf("First Create Database.");


            }
```

```
                else
                {

                        p=head;

                        nn=(Node*)malloc(sizeof(Node));


    printf("Enter Roll NO.,Name, internal1 and internal2 marks of subject1 & subject2");
                        scanf("%d",&(nn->roll));
                        flushall();
                        gets(nn->name);
                        scanf("%f",&(nn->subject1));
                        scanf("%f", &(nn->subject2));
                        scanf("%f",&(nn->internal1));
                        scanf("%f",&(nn->internal2));

                        nn->next=NULL;
                        while(p->next!=NULL)
                        {
                                p=p->next;

                        }

                        p->next=nn;

                        printf("Entry is Created successfully.");
                        Display(head);
                }
```

```
                    break;
                }
                printf("Do you want to Insert more data(Y/N)???");
                flushall();
                scanf("%c",&ans);

        }while(ans=='y' || ans=='Y');

        return head;
}
```

DELETE :here, delete function deletes the details of the students at the begging, at the middle or at the end. As the user wants to delete ehere.

```
Node* Delete(Node* head)
{
        Node *p,*q,*r;
        char ans;
        int ch,n;

        do{
                printf("Which Entry you want to Delete???");
                printf("1.First2.Middle3.End");
                scanf("%d",&ch);

                                if(head==NULL)
                                {

                                        printf("Yet database is not created.");
                                        printf("Database is empty.");
                                        printf("First Create Database.");
```

```
}

else
{
        switch(ch)
        {
        case 1:

                p=head;

                head=head->next;
                free(p);

                printf("First entry is deleted.");

                Display(head);

        break;

        case 2:
                        /*Delete middle Node*/

                p=head;
                printf("Enter roll no. which you want to delete:");
                scanf("%d",&n);

                while((p->next)->roll!=n && p->next->next!=NULL)
                {
```

```
                                        p=p->next;

                                }

                                if(p->next->next==NULL)
                                {
                                        printf("There is no such entry.");

                                }

                                else
                                {
                                        q=p->next;
                                        r=q->next;
                                        p->next=r;
                                        free(q);

                                        printf("Entry is deleted.");
                                        Display(head);
                                }
                                break;

                        case 3:

                                p=head;
                                while(p->next->next!=NULL)
                                {
                                        p=p->next;

                                }
                                        q=p->next;
                                        free(q);

                                        p->next=NULL;
                                        printf("Last entry is deleted.");
                                        Display(head);

                                   break;

                              }

                         }

                    printf("Do you want to delete more data(Y/N)???");
                    flushall();
                    scanf("%c",&ans);

               }while(ans=='y'||ans=='Y');


     return head;

}
```

SEARCH :Here, the search function searches the student as the user wants by their respective roll no.

```
void Search(Node *head)
{

     Node *p;
     int r,cnt=0;
     if(head==NULL)
```

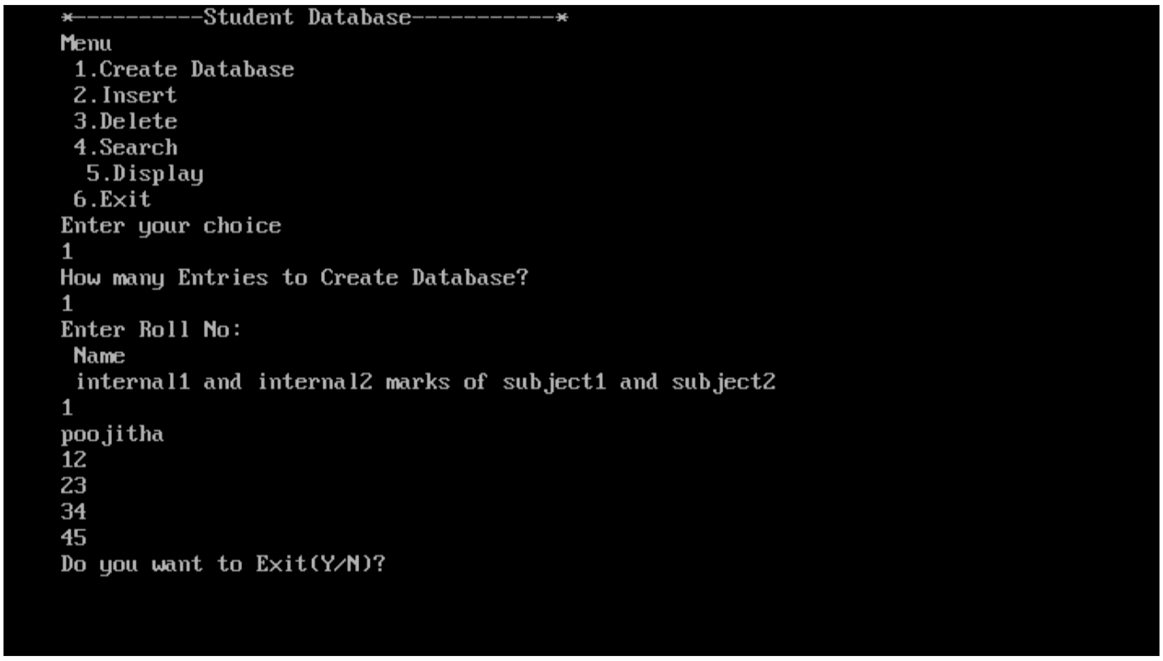
```
        {
                printf("Yet database is not created.");
                printf("Database is empty.");
                printf("First Create Database.");
        }

        else
        {
                p=head;
                printf("Enter roll no. which you want to Search:");
                scanf("%d",&r);

                while(p->roll!=r && p->next!=NULL)
                {
                        p=p->next;
                        cnt++;
                }

        if(p->roll!=r)
        {
                printf("There is no such entry.");
                printf("There are no records.");
        }

        else
        {
                printf("Roll NO. %d is at %d th Position.",r,(cnt+1));
                printf("Roll No.   Name      internal1 and internal2 marks of
        Subject1   subject2");

       printf("%d                    %s                          %0.2f%0.2f%0.2f  %0.2f",p->roll,p->name,p-
>internal1,p->internal2,p->subject1,p->subject2);
```

```
        }
        }


}
```

CHAPTER 5

RESULT

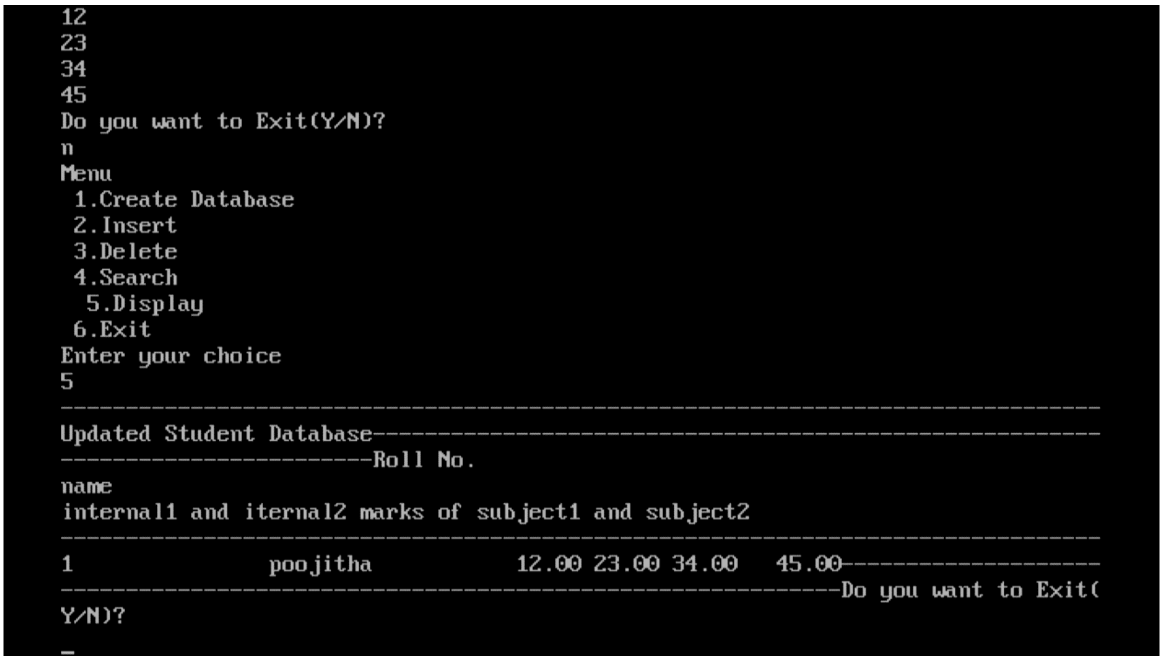
Here, by using create function we created the details of the student such as name, roll no. and marks.



```
*----------Student Database-----------*
Menu
 1.Create Database
 2.Insert
 3.Delete
 4.Search
  5.Display
 6.Exit
Enter your choice
1
How many Entries to Create Database?
1
Enter Roll No:
 Name
 internal1 and internal2 marks of subject1 and subject2
1
poojitha
12
23
34
45
Do you want to Exit(Y/N)?
```

Fig 5.1: CREATION

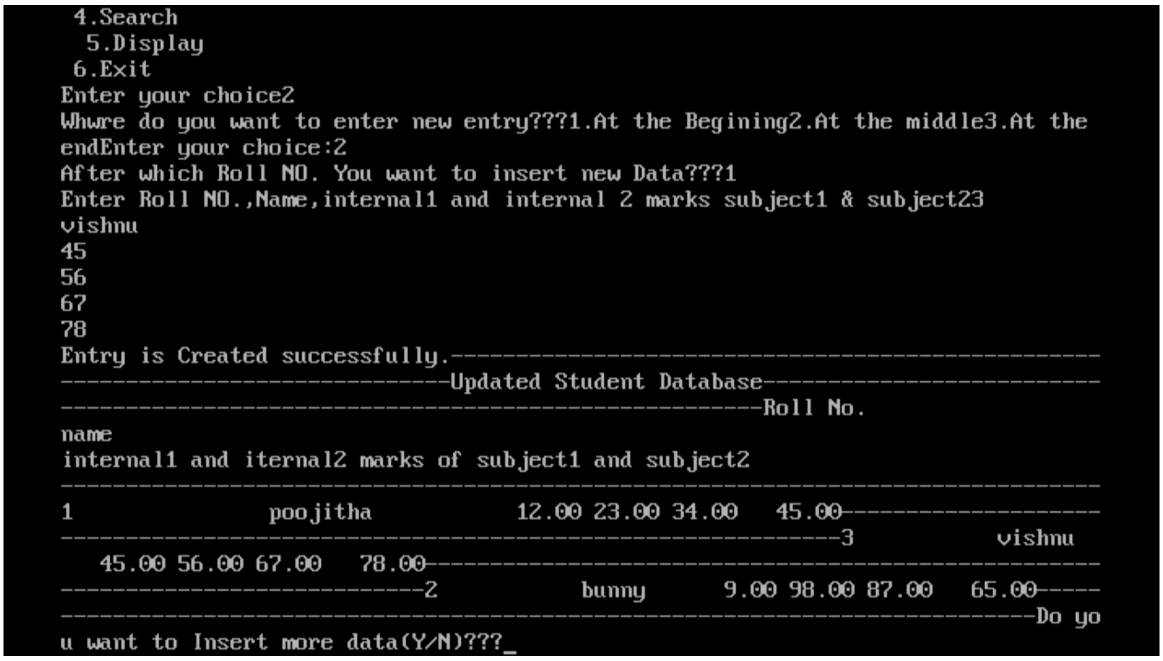
Here, by using display function we have displayed the details of the students.



```
12
23
34
45
Do you want to Exit(Y/N)?
n
Menu
 1.Create Database
 2.Insert
 3.Delete
 4.Search
  5.Display
 6.Exit
Enter your choice
5
--------------------------------------------------------------------------------
Updated Student Database--------------------------------------------------------
------------------------Roll No.
name
internal1 and iternal2 marks of subject1 and subject2
--------------------------------------------------------------------------------
1               poojitha           12.00 23.00 34.00   45.00--------------------
------------------------------------------------------------Do you want to Exit(
Y/N)?
_
```

Fig 5.2: DISPLAY

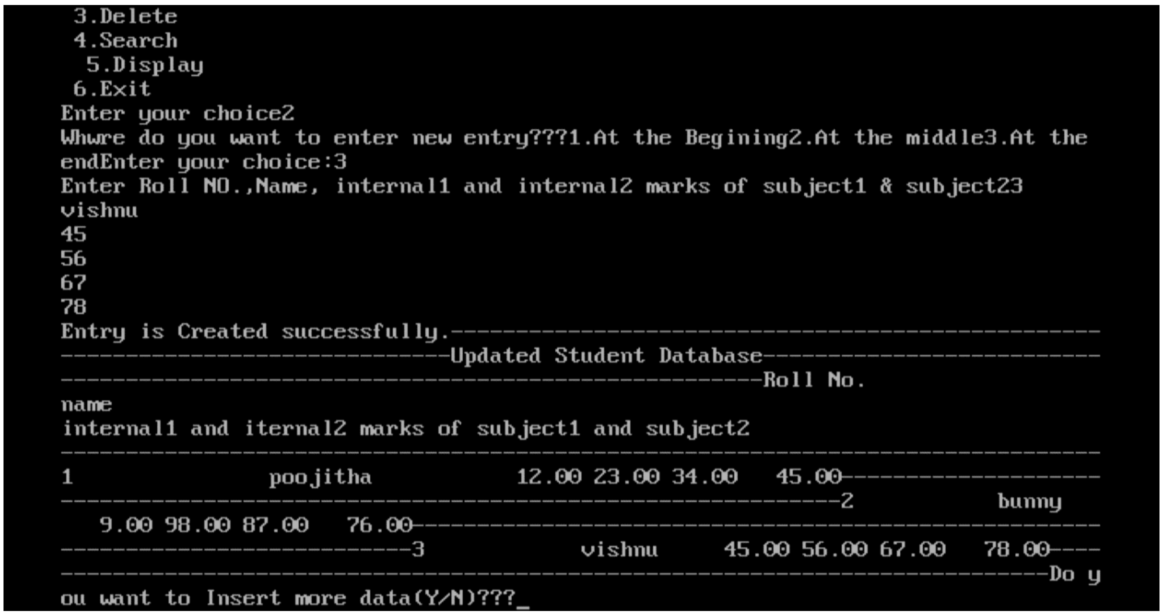
Here, we have inserted the detail of the student at the beginning.



```
     4.Search
      5.Display
     6.Exit
    Enter your choice2
    Whwre do you want to enter new entry???1.At the Begining2.At the middle3.At the
    endEnter your choice:2
    After which Roll NO. You want to insert new Data???1
    Enter Roll NO.,Name,internal1 and internal 2 marks subject1 & subject23
    vishnu
    45
    56
    67
    78
    Entry is Created successfully.-----------------------------------------------------
    ------------------------------Updated Student Database---------------------------
    ------------------------------------------------------Roll No.
    name
    internal1 and iternal2 marks of subject1 and subject2
    --------------------------------------------------------------------------------
    1               poojitha          12.00 23.00 34.00    45.00--------------------
    ------------------------------------------------------------3           vishnu
       45.00 56.00 67.00    78.00---------------------------------------------------
    ----------------------------2           bunny      9.00 98.00 87.00    65.00-----
    ---------------------------------------------------------------------------Do yo
    u want to Insert more data(Y/N)???_
```

Fig5.3: INSERTION AT BEGINNING

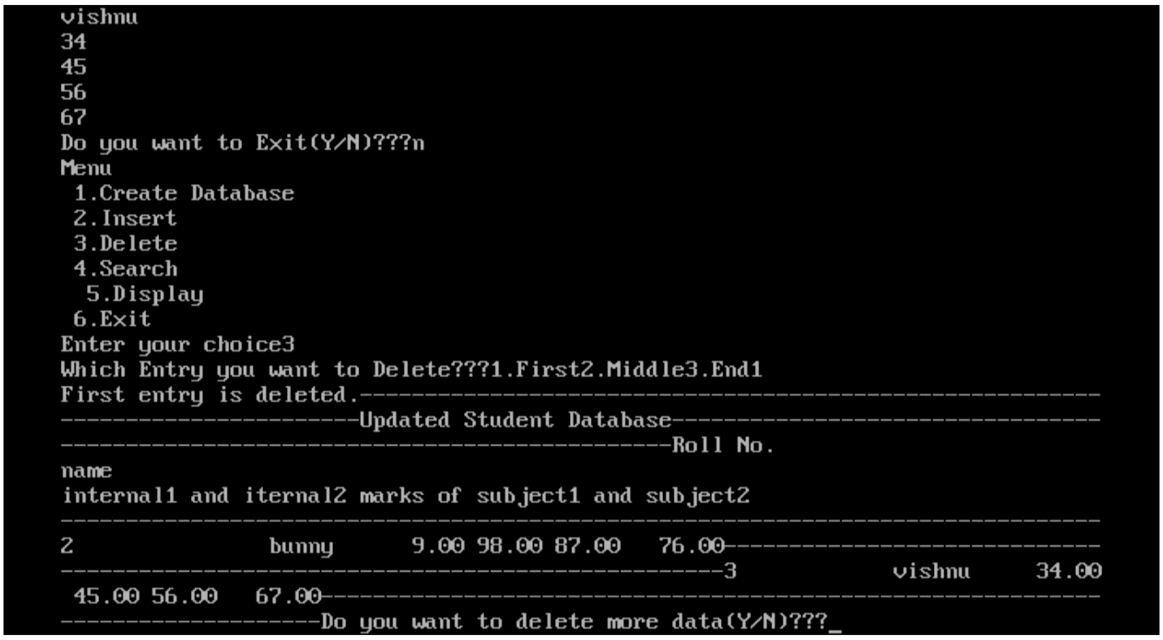
Here, we have inserted the details of the student in the middle.



```
  3.Delete
  4.Search
   5.Display
  6.Exit
Enter your choice2
Whwre do you want to enter new entry???1.At the Begining2.At the middle3.At the
endEnter your choice:3
Enter Roll NO.,Name, internal1 and internal2 marks of subject1 & subject23
vishnu
45
56
67
78
Entry is Created successfully.------------------------------------------------
------------------------------Updated Student Database--------------------------
-------------------------------------------------------Roll No.
name
internal1 and iternal2 marks of subject1 and subject2
--------------------------------------------------------------------------------
1               poojitha           12.00 23.00 34.00   45.00--------------------
-------------------------------------------------------------2           bunny
   9.00 98.00 87.00   76.00----------------------------------------------------
---------------------------3            vishnu     45.00 56.00 67.00   78.00----
----------------------------------------------------------------------------Do y
ou want to Insert more data(Y/N)???_
```

Fig 5.4: INSERTION AT MIDDLE

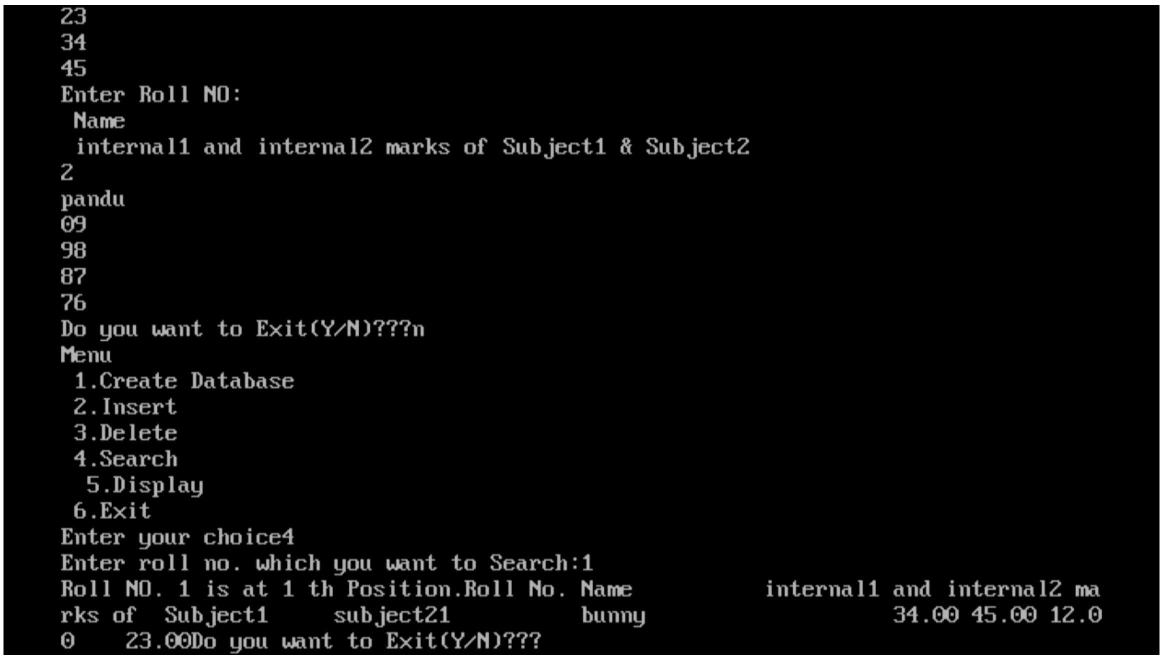
Here, we have deleted the details of the student.



```
vishnu
34
45
56
67
Do you want to Exit(Y/N)???n
Menu
 1.Create Database
 2.Insert
 3.Delete
 4.Search
  5.Display
 6.Exit
Enter your choice3
Which Entry you want to Delete???1.First2.Middle3.End1
First entry is deleted.--------------------------------------------------------
-----------------------Updated Student Database---------------------------------
-----------------------------------------------Roll No.
name
internal1 and iternal2 marks of subject1 and subject2
--------------------------------------------------------------------------------
2               bunny      9.00 98.00 87.00   76.00-----------------------------
---------------------------------------------------3            vishnu     34.00
 45.00 56.00   67.00-----------------------------------------------------------
--------------------Do you want to delete more data(Y/N)???_
```

Fig 5.5: DELETION

Here, we have searched the details of the particular student by their roll no.



```
23
34
45
Enter Roll NO:
 Name
 internal1 and internal2 marks of Subject1 & Subject2
2
pandu
09
98
87
76
Do you want to Exit(Y/N)???n
Menu
 1.Create Database
 2.Insert
 3.Delete
 4.Search
  5.Display
 6.Exit
Enter your choice4
Enter roll no. which you want to Search:1
Roll NO. 1 is at 1 th Position.Roll No. Name          internal1 and internal2 ma
rks of  Subject1     subject21          bunny                  34.00 45.00 12.0
0    23.00Do you want to Exit(Y/N)???
```

Fig 5.6: SEARCH

Chapter 6

CONCLUSION

The purpose of this project **“STUDENT DATABASE”** is to create a student database and the user can easily access the details of the student. The user can easily search for the details of the student, insert the details of the student, display the details of the student.

REFERENCES

- Introduction to programming with C padma reddy
- Data structures and algorithm with C-Mark Allen Weiss
- Lets C-Yashwanth Ketnekar