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: 213DC

Lab Project Name: HOSPITAL MANAGEMENT SYSTEM

Student Details

Name		ID
1.	Rayhan Chowdhury	213902078

Submission Date: 9/11/2022_ _

Course Teacher's Name: Md. Sultanul Islam Ovi_ _

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

Lab Project Status

Marks: ………………………………… **Signature: ....**

Comments: .... **Date: ....**

Table of Contents

Chapter 1

Introduction

1.1 Introduction

2 The project Hospital Management system include registration of patients, storing their details into the system. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search details of a patient using the id. Patients detail are also sorted.

1.2 Design Goals/Objective

The goal of the project is to design a Hospital Management Systems:

1. Recording information about the Patients that come.

2. Recording information related to diagnosis given to Patients.

3. Keeping record of the Immunization provided to patients.

4. To allow staff to easily update information in the system, and have the system inform everyone who needs to know.

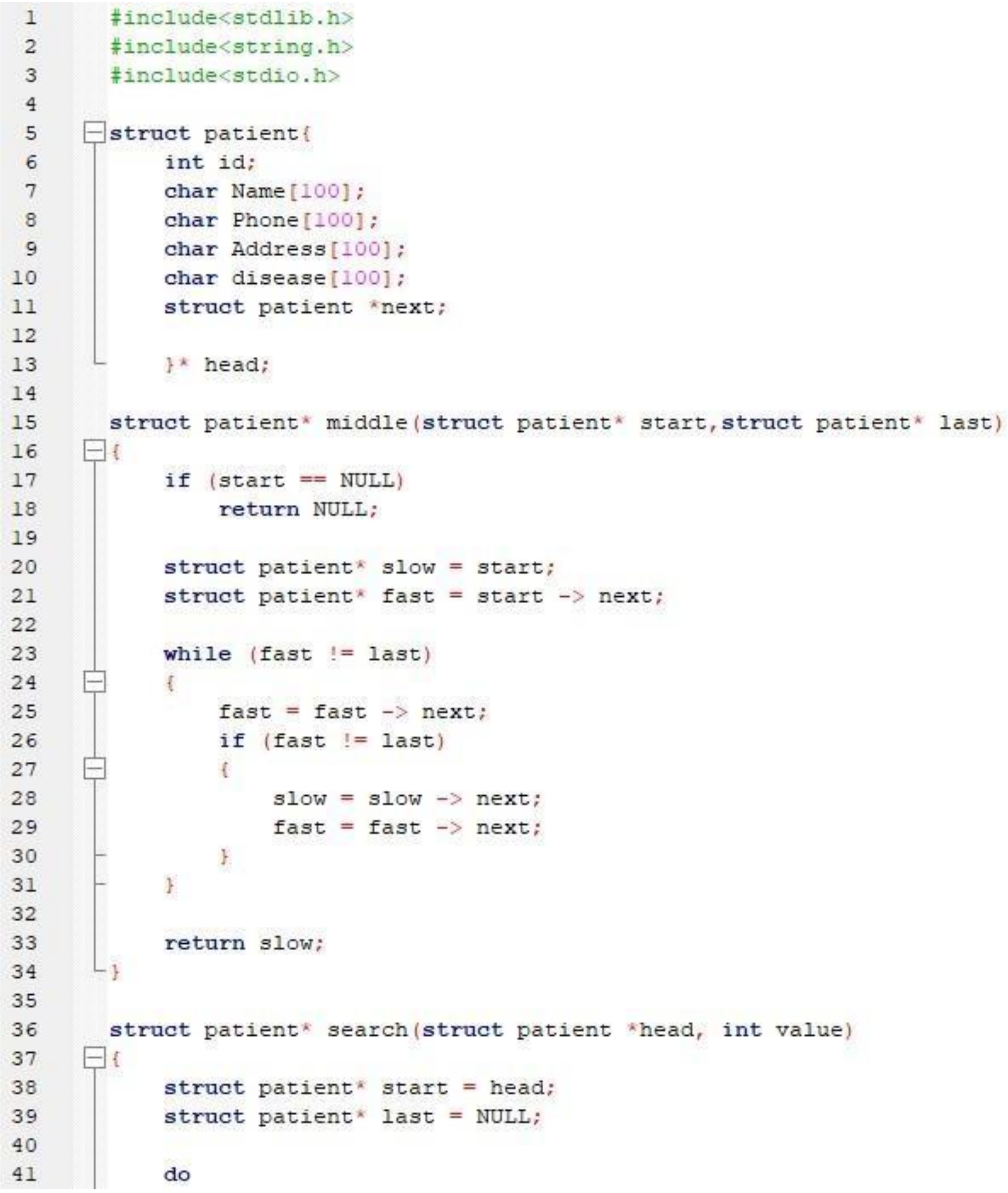
5. And also patients details are sorted so that doesn't creat any conflict.

Chapter 2

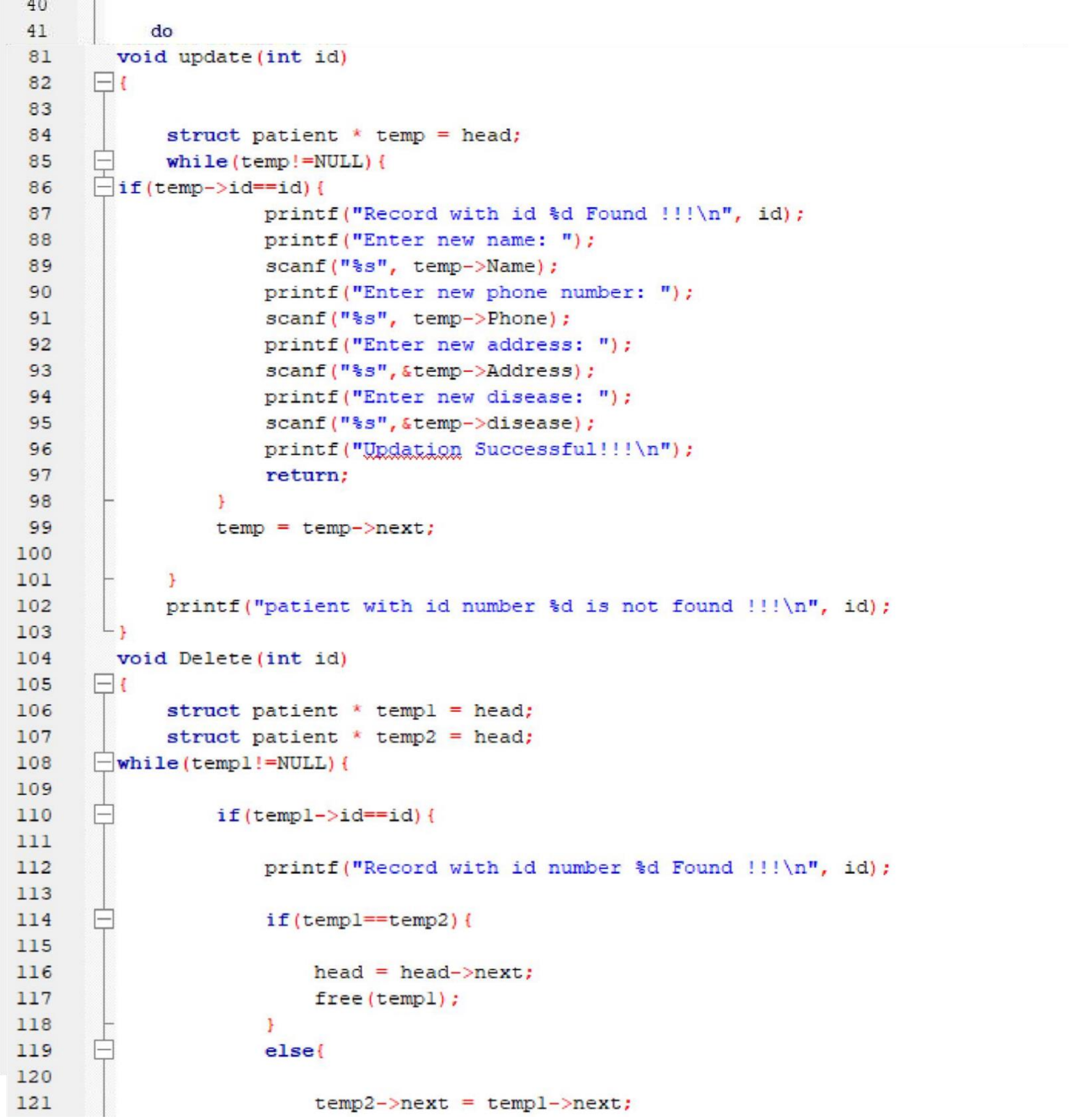
Implementation of the Project

1. Implementation

C sou

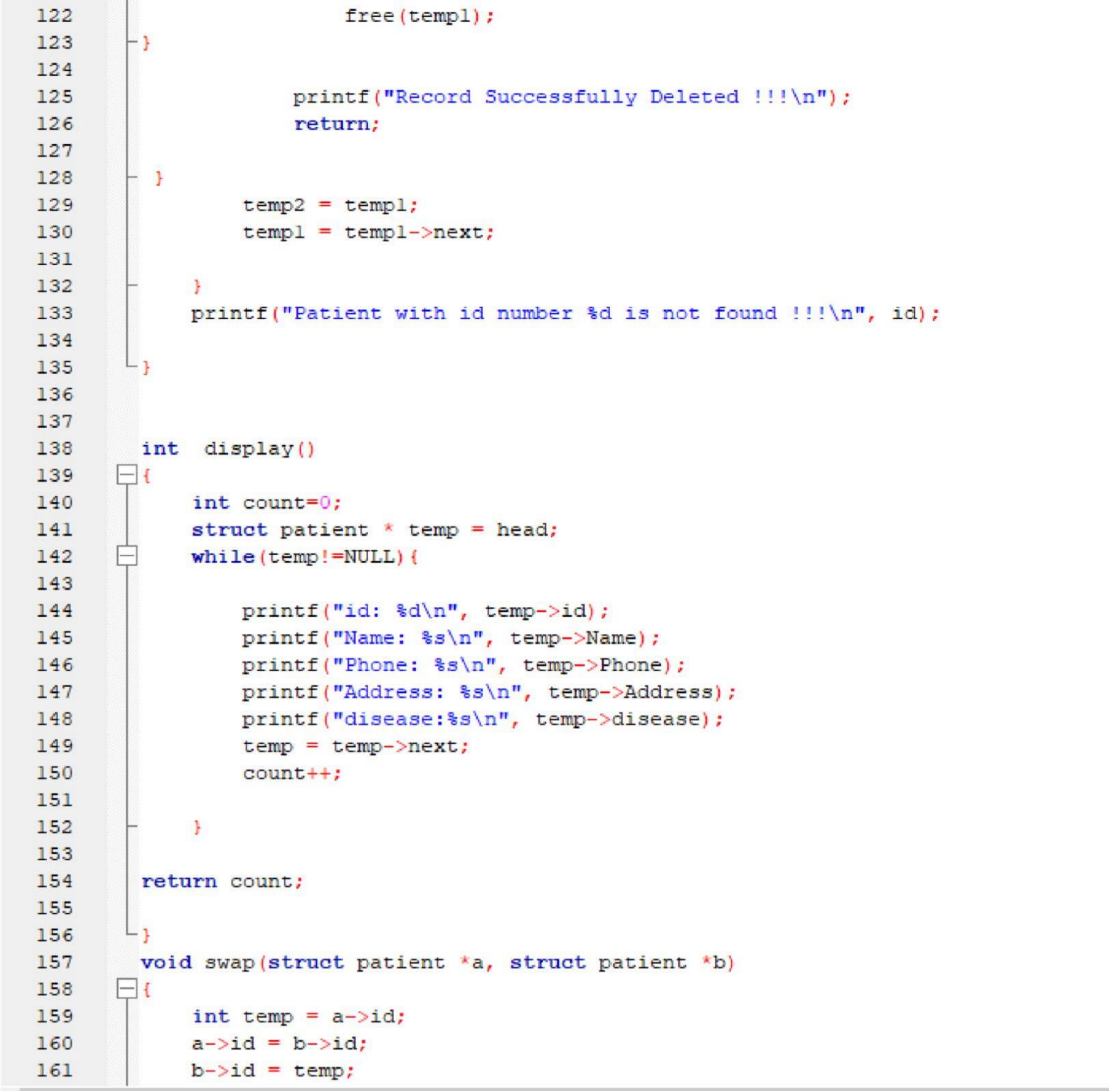


```c
#include<stdlib.h>
#include<string.h>
#include<stdio.h>

struct patient{
    int id;
    char Name[100];
    char Phone[100];
    char Address[100];
    char disease[100];
    struct patient *next;

    }* head;

struct patient* middle(struct patient* start,struct patient* last)
{
    if (start == NULL)
        return NULL;

    struct patient* slow = start;
    struct patient* fast = start -> next;

    while (fast != last)
    {
        fast = fast -> next;
        if (fast != last)
        {
            slow = slow -> next;
            fast = fast -> next;
        }
    }

    return slow;
}

struct patient* search(struct patient *head, int value)
{
    struct patient* start = head;
    struct patient* last = NULL;

    do
```

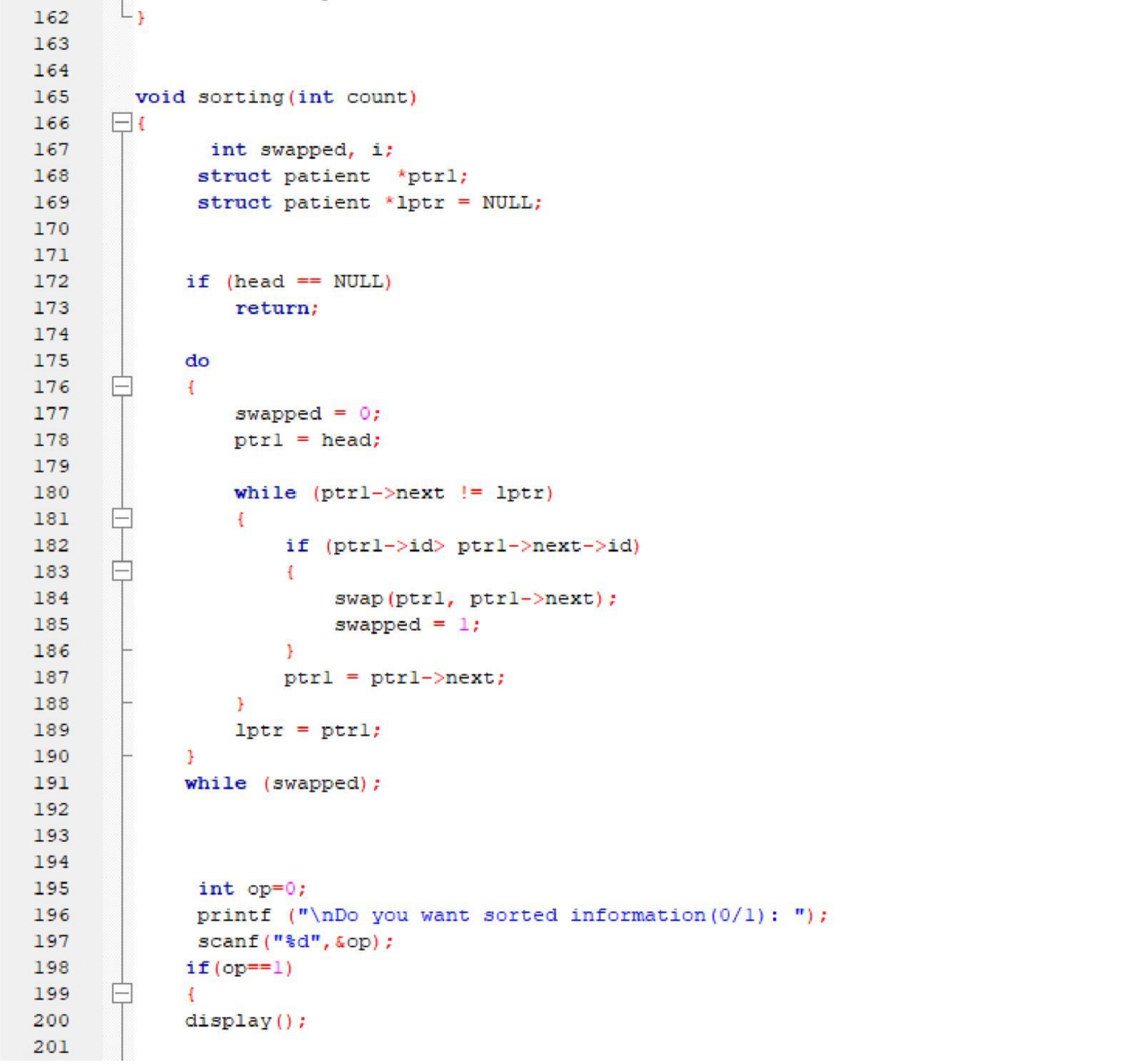


```
    do
```

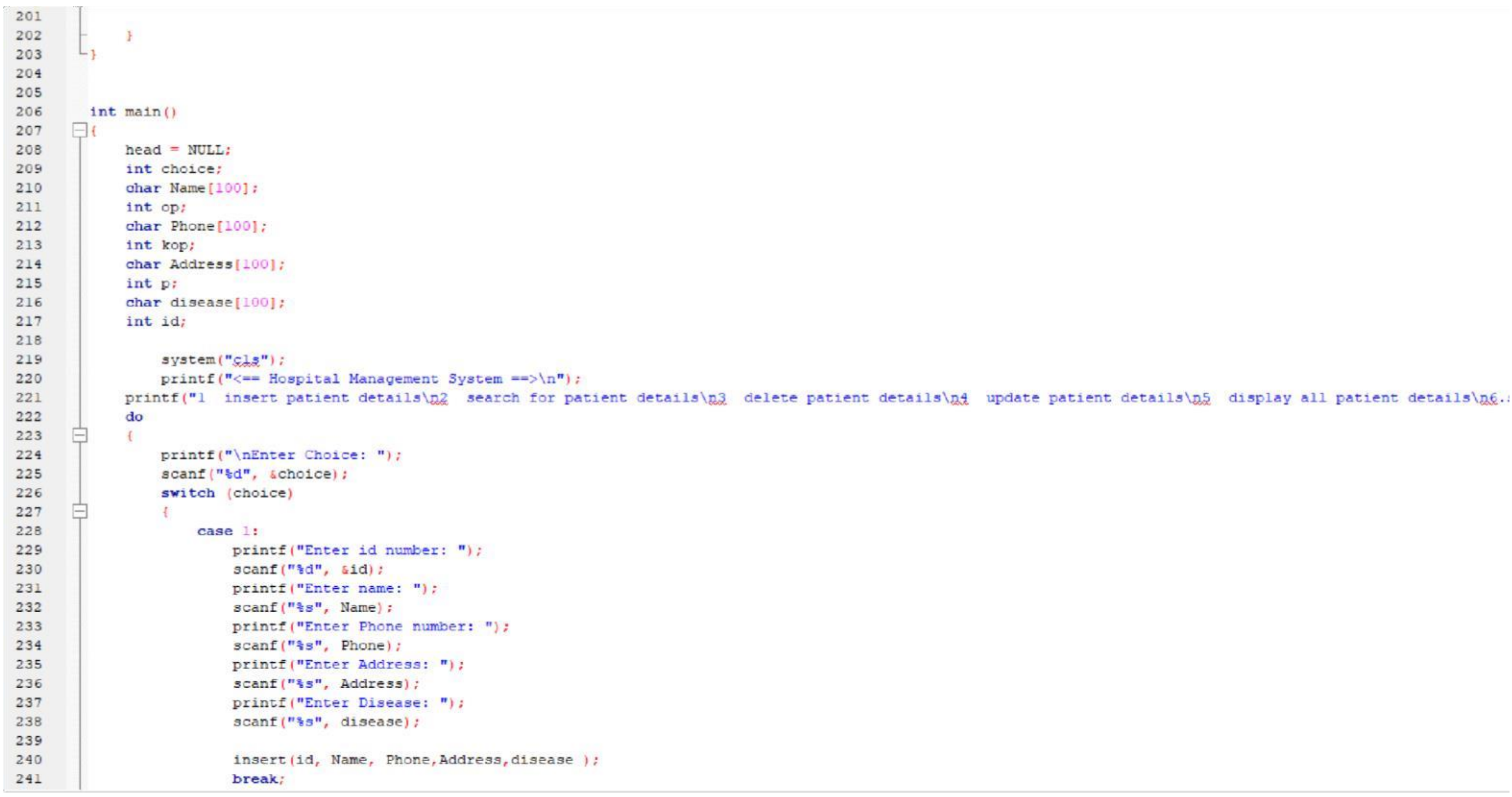
```
void update(int id)
{

    struct patient * temp = head;
    while(temp!=NULL){
if(temp->id==id){
            printf("Record with id %d Found !!!\n", id);
            printf("Enter new name: ");
            scanf("%s", temp->Name);
            printf("Enter new phone number: ");
            scanf("%s", temp->Phone);
            printf("Enter new address: ");
            scanf("%s",&temp->Address);
            printf("Enter new disease: ");
            scanf("%s",&temp->disease);
            printf("Updation Successful!!!\n");
            return;
        }
        temp = temp->next;

    }
    printf("patient with id number %d is not found !!!\n", id);
}
void Delete(int id)
{
    struct patient * temp1 = head;
    struct patient * temp2 = head;
while(temp1!=NULL){

        if(temp1->id==id){

            printf("Record with id number %d Found !!!\n", id);

            if(temp1==temp2){

                head = head->next;
                free(temp1);
            }
            else{

                temp2->next = temp1->next;
```



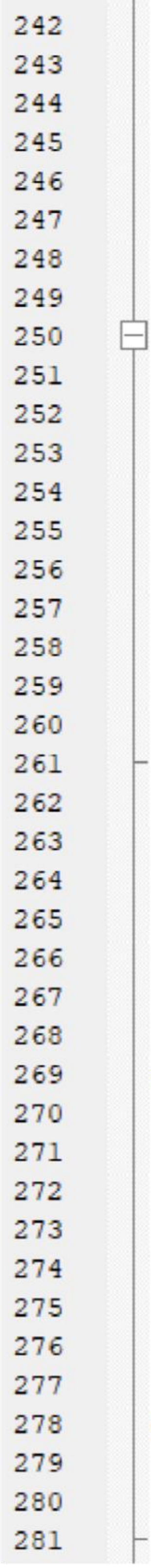
```c
                free(temp1);
}

            printf("Record Successfully Deleted !!!\n");
            return;

 }
        temp2 = temp1;
        temp1 = temp1->next;

    }
    printf("Patient with id number %d is not found !!!\n", id);

}


int  display()
{
    int count=0;
    struct patient * temp = head;
    while(temp!=NULL){

        printf("id: %d\n", temp->id);
        printf("Name: %s\n", temp->Name);
        printf("Phone: %s\n", temp->Phone);
        printf("Address: %s\n", temp->Address);
        printf("disease:%s\n", temp->disease);
        temp = temp->next;
        count++;

    }

return count;

}
void swap(struct patient *a, struct patient *b)
{
    int temp = a->id;
    a->id = b->id;
    b->id = temp;
```



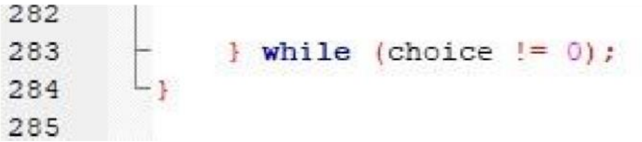
```
}


void sorting(int count)
{
      int swapped, i;
     struct patient  *ptr1;
     struct patient *lptr = NULL;


    if (head == NULL)
        return;

    do
    {
        swapped = 0;
        ptr1 = head;

        while (ptr1->next != lptr)
        {
            if (ptr1->id> ptr1->next->id)
            {
                swap(ptr1, ptr1->next);
                swapped = 1;
            }
            ptr1 = ptr1->next;
        }
        lptr = ptr1;
    }
    while (swapped);


     int op=0;
     printf ("\nDo you want sorted information(0/1): ");
     scanf("%d",&op);
    if(op==1)
    {
    display();

```



```

    }
}


int main()
{
    head = NULL;
    int choice;
    char Name[100];
    int op;
    char Phone[100];
    int kop;
    char Address[100];
    int p;
    char disease[100];
    int id;

        system("cls");
        printf("<== Hospital Management System ==>\n");
    printf("1  insert patient details\n2  search for patient details\n3  delete patient details\n4  update patient details\n5  display all patient details\n6.
    do
    {
        printf("\nEnter Choice: ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                printf("Enter id number: ");
                scanf("%d", &id);
                printf("Enter name: ");
                scanf("%s", Name);
                printf("Enter Phone number: ");
                scanf("%s", Phone);
                printf("Enter Address: ");
                scanf("%s", Address);
                printf("Enter Disease: ");
                scanf("%s", disease);

                insert(id, Name, Phone,Address,disease );
                break;
```

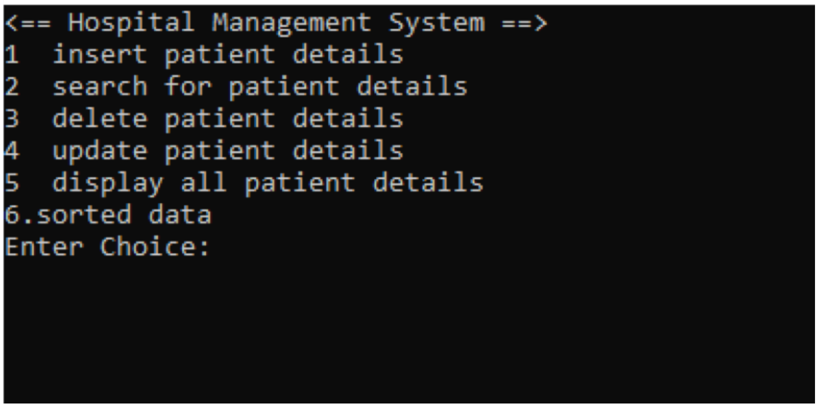


```

            case 2:
                printf("id number to search: ");
            scanf("%d", &id);

            if (search(head,id) == NULL)
        printf("Value not present\n");
    else
    {
        struct patient * lol=search(head,id);

        printf("id Number: %d\n", lol->id);
        printf("Name: %s\n", lol->Name);
        printf("Phone: %s\n", lol->Phone);
        printf("Address: %s\n", lol->Address);
        printf("Disease: %s\n", lol->disease);



    }
            break;
            case 3:
                printf("Enter id number to delete: ");
                scanf("%d", &id);
                Delete(id);
                break;

case 4:
                printf("Enter id number to update: ");
                scanf("%d", &id);
                update(id);
                break;
 case 5:
              kop=display();
                break;

case 6:
                    sorting(kop);
                    break;
        }
```



```

    } while (choice != 0);
}

```

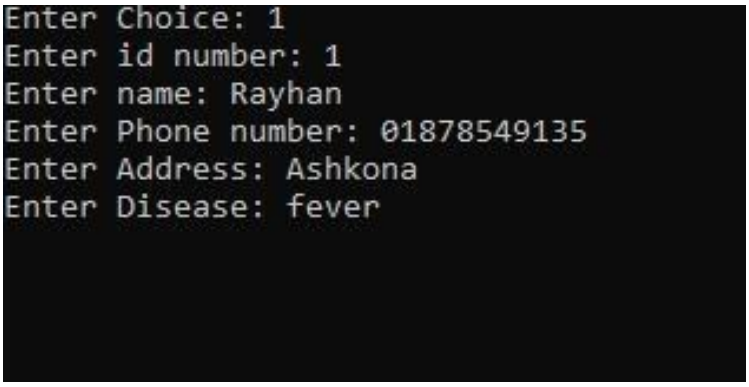
Screenshots



```
<== Hospital Management System ==>
1  insert patient details
2  search for patient details
3  delete patient details
4  update patient details
5  display all patient details
6.sorted data
Enter Choice:
```

Figure 1: main menu function

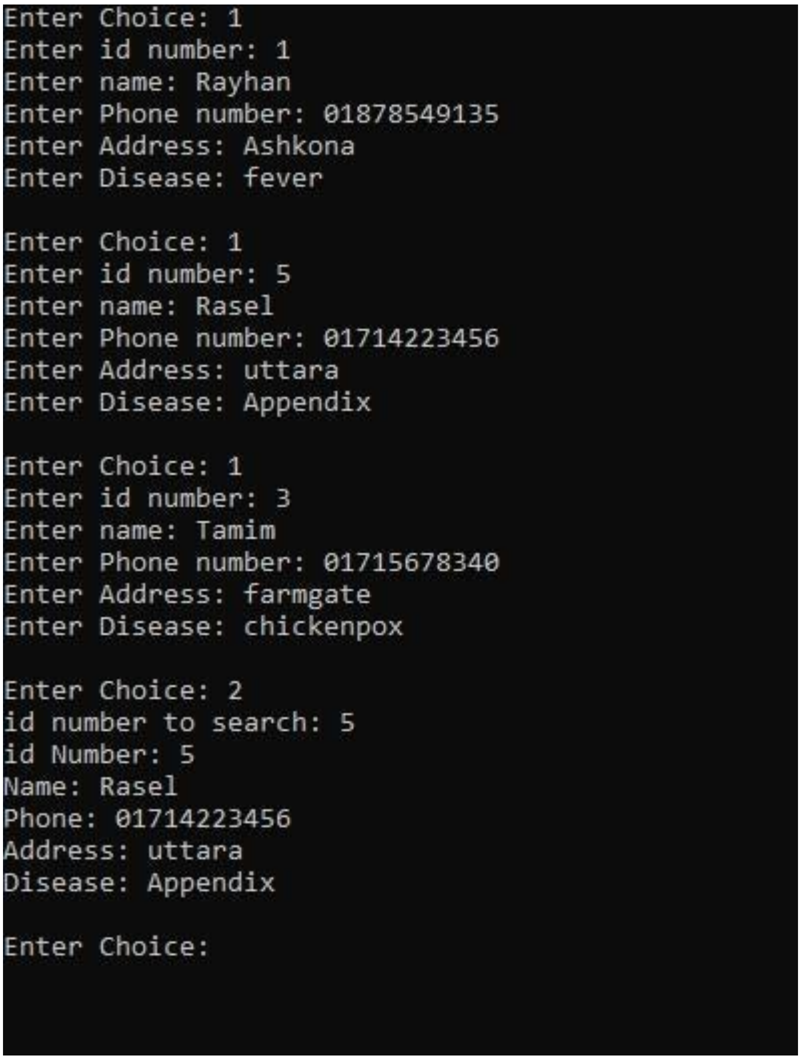
When the program is executed, the user will be directed to the main menu interface. The program is introduced with a few lines of texts. Then six selections are made for the user as the user can choose to insert, search, delete, update, display & sorted data.



```
Enter Choice: 1
Enter id number: 1
Enter name: Rayhan
Enter Phone number: 01878549135
Enter Address: Ashkona
Enter Disease: fever
```

Figure 2: Insert function

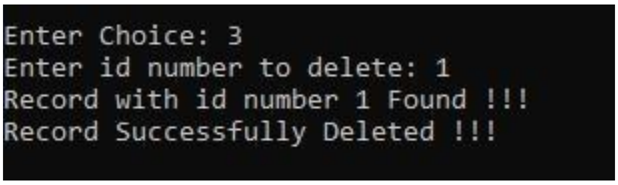
As shown in the interface the patient details has been admitted successfully, after the user has entered the details.



```
Enter Choice: 1
Enter id number: 1
Enter name: Rayhan
Enter Phone number: 01878549135
Enter Address: Ashkona
Enter Disease: fever

Enter Choice: 1
Enter id number: 5
Enter name: Rasel
Enter Phone number: 01714223456
Enter Address: uttara
Enter Disease: Appendix

Enter Choice: 1
Enter id number: 3
Enter name: Tamim
Enter Phone number: 01715678340
Enter Address: farmgate
Enter Disease: chickenpox

Enter Choice: 2
id number to search: 5
id Number: 5
Name: Rasel
Phone: 01714223456
Address: uttara
Disease: Appendix

Enter Choice:
```

Figure 3: Search function

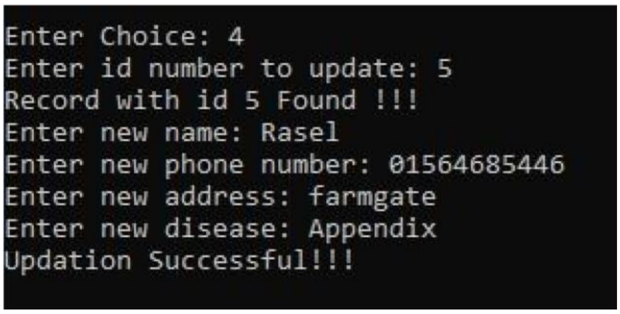
For searching I have use Binary search. After inserting patient details I can search specific patient by using their Id.



```
Enter Choice: 3
Enter id number to delete: 1
Record with id number 1 Found !!!
Record Successfully Deleted !!!
```

Figure 4: Delete function

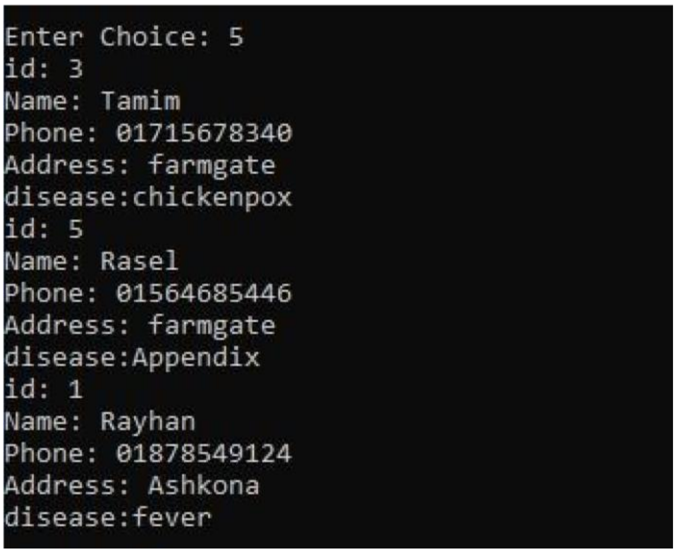
After discharging patient we can easily delete patient detail using delete function.



```
Enter Choice: 4
Enter id number to update: 5
Record with id 5 Found !!!
Enter new name: Rasel
Enter new phone number: 01564685446
Enter new address: farmgate
Enter new disease: Appendix
Updation Successful!!!
```

Figure 5: Update function

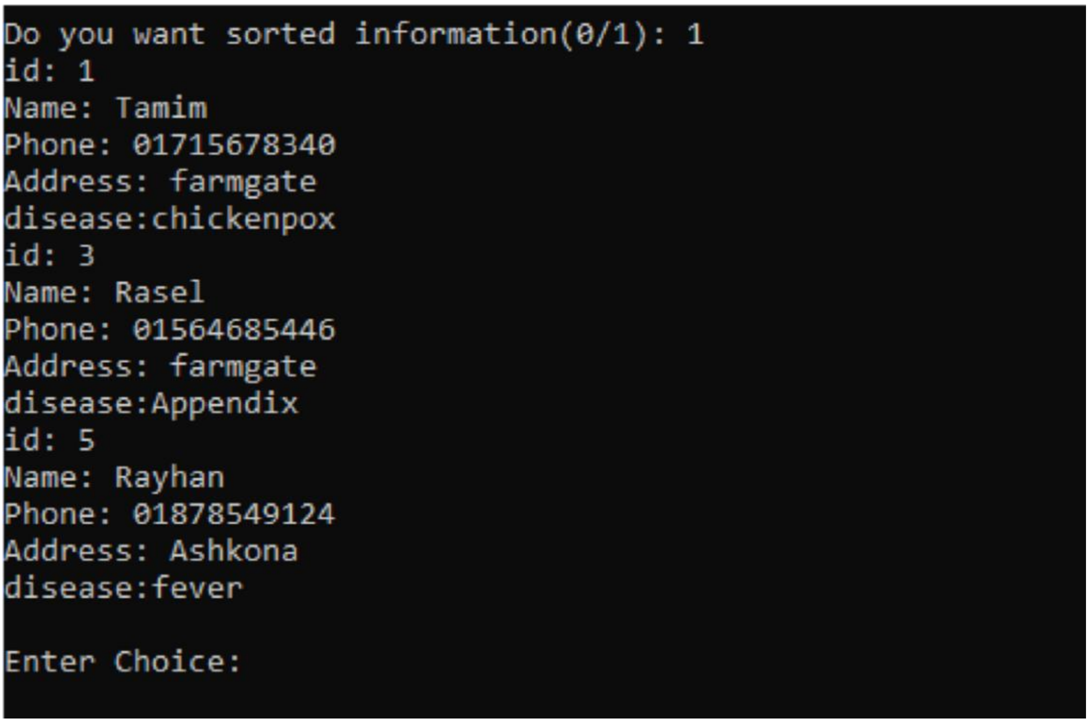
If there is any mistake in patient detail than we can easily update detail using update function. Here I have update phone no, address.



```
Enter Choice: 5
id: 3
Name: Tamim
Phone: 01715678340
Address: farmgate
disease:chickenpox
id: 5
Name: Rasel
Phone: 01564685446
Address: farmgate
disease:Appendix
id: 1
Name: Rayhan
Phone: 01878549124
Address: Ashkona
disease:fever
```

Figure 6: Display function

For any need we can enter choice 5 for display all patient detail.



```
Do you want sorted information(0/1): 1
id: 1
Name: Tamim
Phone: 01715678340
Address: farmgate
disease:chickenpox
id: 3
Name: Rasel
Phone: 01564685446
Address: farmgate
disease:Appendix
id: 5
Name: Rayhan
Phone: 01878549124
Address: Ashkona
disease:fever

Enter Choice:
```

Figure 6: sort function

For serially display patient detail we can use sort function. I have use Bubble sort for this system.

Chapter 3

Conclusion

Learning Outcome

Hospital management system is all about the modernizing a hospital through use of technology. Computers helps in it and take over the manual system for quick and easy functioning. This hospital management system is a quite the reliable and is proven on many stages. All the basic requirements of the hospital are provided in the hospital in order to manage it perfectly and large amount of data can also be stored. It gives many facilities like searching for the detail of patient

Future Scope

- We can use Doctor detail and their specialist option.
- We can add Farmacy system, Emergency sysrem, for making better hospital .