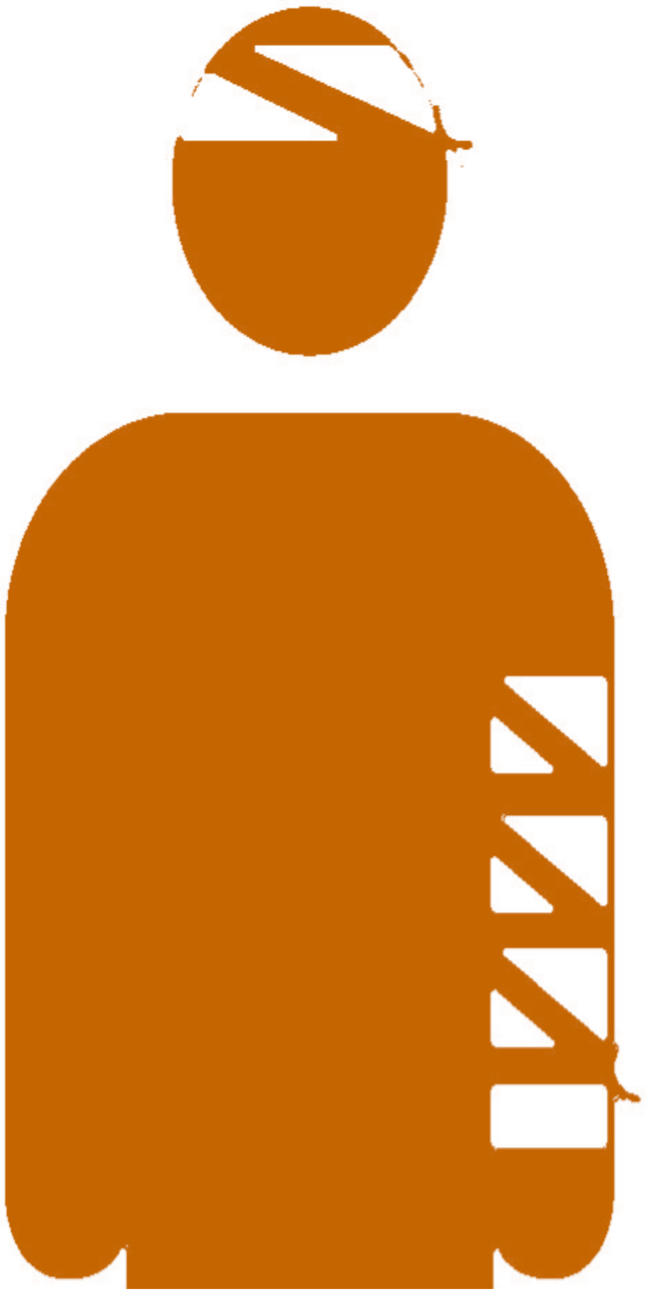
C.SC. PROJECT

Hospital Management System

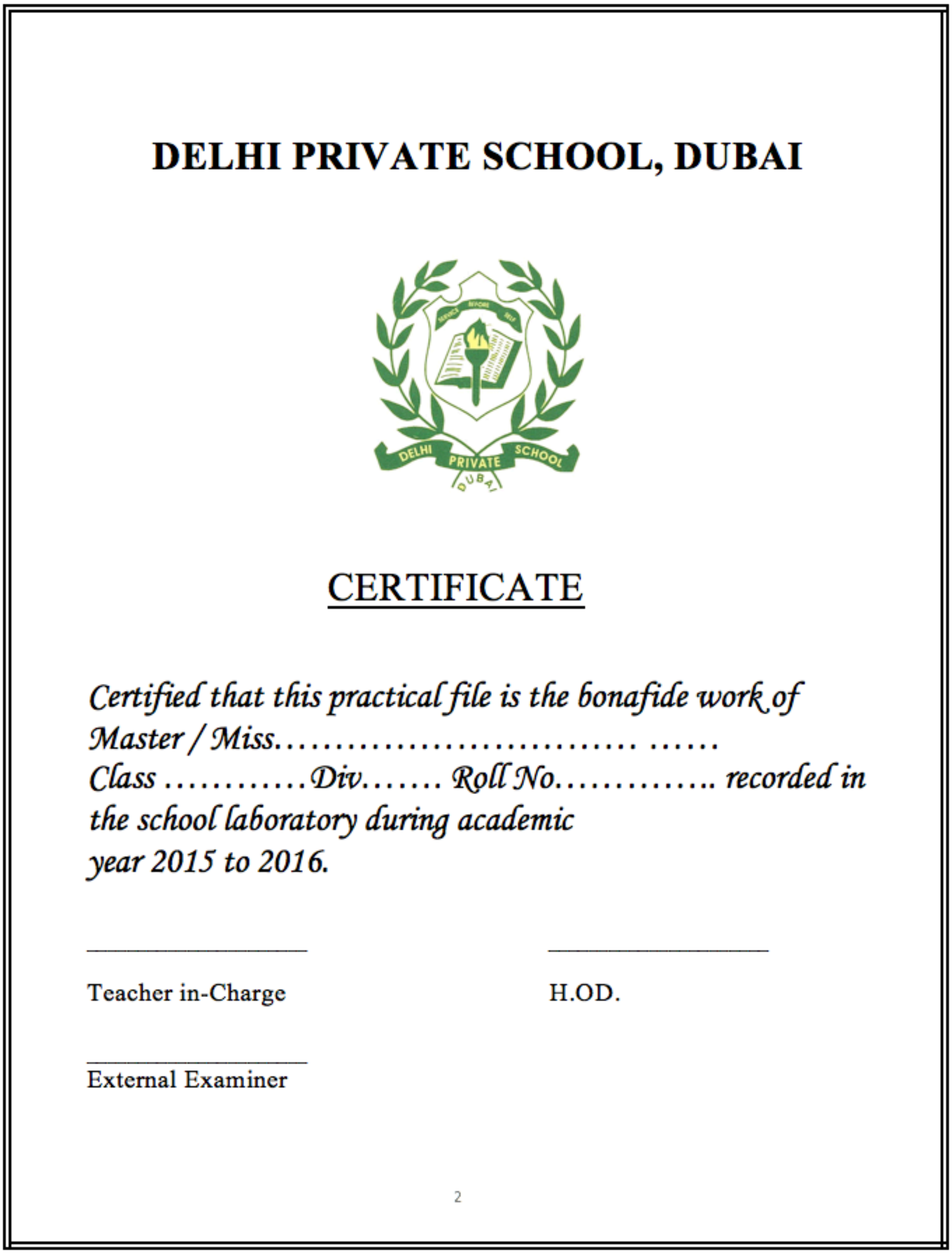


By

Manav Chawla

Grade XII-A

(Academic year 2015-2016)



DELHI PRIVATE SCHOOL, DUBAI

<u>CERTIFICATE</u>

Certified that this practical file is the bonafide work of Master / Miss.............................. ......
Class ....Div....... Roll No.............. recorded in the school laboratory during academic year 2015 to 2016.

______________________ ______________________

Teacher in-Charge H.OD.

____

External Examiner

2

Table of Contents

Acknowledgement

I hereby express gratitude toward my principal Ms. Rashmi Nandkeolyar, my Computer Science teacher Ms. Richi and HOD Ms. Nimmi.

I would like to thank Ms. Richi for her support and guidance in our endeavor to create a smart and powerful program, and for being a wonderful teacher throughout our learning process.

I would also like to thank Ms Nimmi, HOD of Computer Science for providing me with the facilities and platform required to complete my project successfully.

I would like to thank my parents who supported and also helped me in all my endeavours. And last but not the least I thank the Almighty for whatever I have achieved till now.

Problem Definition

- The creation and manipulation of a hospital management system that holds patient details. The user of the program must be able to –
  - **View patient records**
  - **Add patient records**
  - **Modify patient records**
  - **Delete patient records**
  - **Admit a patient**
  - **Discharge a patient**

- All outputs and inputs must be in a user-friendly and aesthetically pleasing format.

- The program listing should be well-documented and must make use of meaningful variable and function names.

- The program should utilize the concept of data file handling to execute the necessary operations.

Hardware & Software

The following hardware and software will be required to run this program:

Hardware

- Personal Computer with Intel 80386 (32-bit microprocessor) or higher
- Keyboard
- Monitor
- Mouse

Software

The program will run on

- PC-DOS (MS-DOS) 4.01 or later
- Microsoft Windows 3.1 or later

Structure Definition

Structure Name: ‘record’

Sl No.	Member Name	Member Type	Member Size	Constraint	Example
1.	PatientID	int			7981
2.	Name	char	50		Ashmeet Kaur
3.	WardNo	int			101
4.	Status	char		A/D	A

File Contents

Sl No.	PatientID	Name	WardNo	Status
1.	7981	Ashmeet Kaur	101	D
2.	8012	Raniya K.P.	107	A
3.	1123	Rachna Mallara	109	D
4.	3017	Anna Mathew	102	A
5.	7019	Aishwarya Suresh	110	A
6.	5076	Amritha J.K.	103	D
7.	5715	Siona Fernandes	106	A
8.	4996	Arathi Radeesh	114	D
9.	9917	Aksha Sajeev	105	D
10.	1112	Neela Ramesh	104	A

Program Algorithm

Step 1: Start.

Step 2: Declare header files required to use library functions like cout, cin, gets(), etc.

Step 3: Declare the global structure 'record' that will hold various patient details like patient ID, patient name, ward number, and status.

Step 4: Declare six function prototypes, of return type void, for the six functions the program performs (view, add, modify or delete a patient record, or admit/discharge a patient).

Step 5: Define a function design() of type void to be used in formatted display (this is used in several places to avoid repeating code for similar displays).

Step 6: Declare Main function.

- Declare a variable 'opt' as a choice variable.
- Start a do-while loop. Within the loop:
    - Statements that output a menu for the user, detailing all the functions that the program performs. The menu instructs the user on how to input data into the program.
    - Input user's choice (1-7) and store it in local variable 'opt'.
        - If choice is 1, UDF View() is called.
        - If choice is 2, UDF Add() is called.
        - If choice is 3, UDF Modify() is called.
        - If choice is 4, UDF Delete() is called.
        - If choice is 5, UDF Admit() is called.
        - If choice is 6, UDF Discharge() is called.
    - Execute the loop while user-input choice is not 7.

Step 7: Define UDF View().

- Declare structure of type record ‘Patient’, integer PID (Patient ID) and a check variable ‘found’ (initialized as 0).
- Open binary file in input mode.
- Input ID number of patient record to be viewed.
- Use a while loop to read the binary file record by record, until a match is found.

```
If (a match is found)
        -Output all the details of the patient
        -Assign found as 1.
```

- If a match is not found, output “RECORD NOT FOUND”.
- After the loop is closed, the file is closed.

Step 8: Define UDF Add().

- Declare structure of type record called ‘Patient’ and choice variable ‘ch’.
- Open the binary file to which records are to be added in output mode.
- Use a do while loop to:

```
-Input new patient ID, name, ward number and status in the record Patient.
-Now write this new record in the binary file.
-Ask whether the user wants to add more records.
        If(choice is ‘Yes’ represented by ‘y’ OR ‘Y’)
                The loop is repeated
        Else
                The loop is exited
-The file is closed after the loop is exited.
```

Step 9: Define UDF Delete().

- Declare a structure of type record called ‘Patient’, integer PID and check variable ‘found’ is initialized with 0.
- Open the binary files from which the record is to be deleted in input mode and create a new binary file in output mode.
- Ask for the Patient ID of the record to be deleted.

- Once this has been input, a while loop that reads the file opened in input mode record by record will try to find a match.

```
If (Match is not found)
        Write the record onto the new file created
If (Match found)
    o   Output details of the record and assign found
        as 1
```

- The loop has finished reading the entire file(loop exited)
- Close the files.
- Now check whether found is 0 or 1.

```
If(found is 0)
        Output "RECORD NOT FOUND"
Else if(found is 1)
        Output "RECORD DELETED"
```

- Remove the original file.
- Rename the new file as the original file.

Step 10: Define UDF Modify().

- Declare a structure of type record called 'Patient', integer PID, a check variable 'found' is initialized with 0 and choice variable YN.
- Open the binary files from which the record is to be modified in input mode and create a new binary file in output mode.
- Ask for the Patient ID of the record to be modified.
- Once this has been input, a while loop that reads the file opened in input mode record by record will try to find a match.

```
If (Match is not found)
        Write the record onto the new file created
If (Match found)
    o   Ask detail by detail whether it is to be
        modified or not and accept the new entries
        based on whether choice is Yes or No.
    o   Now write the modified record onto the new
        file. Assign found as 1.
```

```
  o The loop has finished reading the entire file(loop
    exited).
  o Close the files.
  o Now check whether found is 0 or 1.
        If(found is 0)
                Output "RECORD NOT FOUND"
  o Remove the original file.
  o Rename the new file as the original file.
```

Step 11: Declare UDF Admit().

```
  o Declare structure of type record 'Patient', integer PID
    (Patient ID), a check variable 'found' (initialized as 0) and
    choice variable YN.
  o Open binary file to be read from in input mode and create
    a new binary file in output mode.
  o Input ID number of patient record to be admitted.
  o Use a while loop to read the binary file record by record,
    until a match is found.
        If (a match not found)
            Write the record as such onto the new file
        If (a match is found)
            -Output all the details of the patient
            -Assign found as 1.
            o Check whether patient status is already 'A'.
            o If yes, give a message citing that the patient
              is already admitted
            o If No, Ask whether the patient is to be
              admitted.
              If(Yes)
                  o Change status as 'A' and give a
                    message citing that the patient has
                    been admitted
              Else
                  o Give a message citing that the patient
                    has not been admitted

            o Now rewrite the record onto the new file.
  The loop will exit after completion of reading.
  o If a match is not found, output "RECORD NOT FOUND".
```

- Close the files.
- Now remove the original file and rename the new file with the original one's name.

Step 12: Declare UDF Discharge().

- Declare structure of type record 'Patient', integer PID (Patient ID), a check variable 'found' (initialized as 0) and choice variable YN.
- Open binary file to be read from in input mode and create a new binary file in output mode.
- Input ID number of patient record to be discharged.
- Use a while loop to read the binary file record by record, until a match is found.

```
If (a match not found)
    Write the record as such onto the new file
If (a match is found)
    -Output all the details of the patient
    -Assign found as 1.
     o Check whether patient status is already 'D'.
     o If yes, give a message citing that the patient
       is already discharged
     o If No, Ask whether the patient is to be
       discharged.
       If(Yes)
           o Change status as 'D' and give a
             message citing that the patient has
             been discharged

       Else
           o Give a message citing that the patient
             has not been discharged

     o Now rewrite the record onto the new file.
```

The loop will exit after completion of reading.

- If a match is not found, output "RECORD NOT FOUND".
- Close the files.
- Now remove the original file and rename the new file with the original one's name.

Step 12: Stop.

Program Code

```
//Header Files
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include <iomanip.h>
#include <fstream.h>

//This program functions as a hospital management system.
//It can be used to view, add, delete or modify a patient's record, and also to admit or
discharge a patient.

//Global Structure Declaration
struct record              /*record is a structure that will hold the patient details Patient ID,
Name, Ward No. and Status.*/
{
        int PatientID;
        char Name[50];
        int WardNo;
        char Status;   //Status holds either the character 'A' (Admitted) or 'D' (Discharged).
};


//Function prototypes
void View();
void Add();
void Modify();
void Delete();
void Admit();
void Discharge();

//For Formatted display
void design()
{
 for (int i=1; i<=80; i++)
  cout<<'*';
 cout<<endl;
}

//Main function
void main()
{
 clrscr();

 int opt;

 do
 {
        design();
        cout<<setw(40)<<"MENU"<<endl<<endl;
        design();
        cout<<" Maintenance: "<<endl;
        cout<<"\n   1- View record\n";
        cout<<"   2- Add record\n";
        cout<<"   3- Delete record\n";
        cout<<"   4- Modify record\n";
        cout<<"\n Transaction:\n";
        cout<<"\n   5- Admit\n";
        cout<<"   6- Discharge\n";
        cout<<"\n   7- Exit\n";

        design();

        cout<<"\t\t     Manav Chawla\n\n";

        design();

        cout<<"Enter your choice: ";
        cin>>opt;

        clrscr();

        design();

        switch(opt)
        {
                case 1: cout<<"VIEW RECORD\n\n";
                        design();
                            View();
                        break;
                case 2: cout<<"ADD RECORD\n\n";
                        design();
                                Add();
                        break;
                case 3: cout<<"DELETE RECORD\n\n";
                        design();
                                Delete();
                        break;
                case 4: cout<<"MODIFY RECORD\n\n";
                        design();
                                Modify();
                        break;
                case 5: cout<<"ADMIT\n\n";
                        design();
                            Admit();
                        break;
                case 6: cout<<"DISCHARGE\n\n";
                        design();
                            Discharge();
}
```

```
  clrscr();

 } while (opt>=1&&opt<=6);

}
```

//Opt 1: View record

```
void View()
{
 record Patient;

 int PID, found=0;

 ifstream ifile("HOSPITAL.DAT", ios::binary);

 cout<<"Enter Patient ID: ";
 cin>>PID;
 cout<<endl;

 while (ifile.read((char*)&Patient, sizeof(record)))
 {
  if (Patient.PatientID==PID)
  {
   cout<<"RECORD FOUND"<<endl<<endl;
            cout<<"Patient ID      : "<<Patient.PatientID<<endl;
            cout<<"Patient Name    : "<<Patient.Name<<endl;
   cout<<"Patient Ward No.: "<<Patient.WardNo<<endl;
            cout<<"Patient Status  : "<<Patient.Status<<endl;

   found=1;
  }
 }

 if (found==0)
  cout<<"RECORD NOT FOUND"<<endl;

 design();

 ifile.close();

 cout<<"Enter any key: ";
 getch();
}
```

//Opt 2: Add record

```
void Add()
{
 record Patient;
 char ch;
 ofstream ofile("HOSPITAL.DAT", ios::binary|ios::app);

 do{
 cout<<"Enter new patient's Patient ID   : ";
 cin>>Patient.PatientID;
 cout<<"Enter new patient's Name         : ";
 gets(Patient.Name);
 cout<<"Enter new patient's Ward No.     : ";
 cin>>Patient.WardNo;
 cout<<"Enter new patient's Status (A/D) : ";
 cin>>Patient.Status;

 ofile.write((char*)&Patient, sizeof(Patient));

 cout<<"\nEnter one more record? (Y/N): ";
 cin>>ch;
 cout<<endl;

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

 design();

 ofile.close();

 cout<<"Enter any key: ";
 getch();
}
```

//Opt 3: Delete record

```
void Delete()
{
 record Patient;

 int found=0;
 int PID;

 ifstream ifile("HOSPITAL.DAT", ios::binary);
 ofstream ofile("TEMP.DAT", ios::binary);

 cout<<"Enter Patient ID: ";
 cin>>PID;
 cout<<endl;

 while (ifile.read((char*)&Patient, sizeof(Patient)))
 {
  if (PID!=Patient.PatientID)
                    ofile.write((char*)&Patient, sizeof(Patient));
  else
          {
                    cout<<"\nPatient ID       : "<<Patient.PatientID<<endl;
                    cout<<"Patient Name      : "<<Patient.Name<<endl;
```

```
                        cout<<"Patient Ward No.   : "<<Patient.WardNo<<endl;
                        cout<<"Patient Status     : "<<Patient.Status<<endl;

                        found=1;

            }

 }

 ifile.close();
 ofile.close();

 if (found==0)
  cout<<"\nRECORD NOT FOUND"<<endl;
 else
            cout<<"\nRECORD DELETED\n"<<endl;

 remove("HOSPITAL.DAT");
 rename("TEMP.DAT", "HOSPITAL.DAT");

 design();

 cout<<"Enter any key: ";
 getch();

}
```

//Opt 4: Modify record

```
void Modify()
{
 record Patient;

 int PID, found=0;
 char YN;

 cout<<"Enter Patient ID: ";
 cin>>PID;
 cout<<endl;

 ifstream ifile("HOSPITAL.DAT",ios::binary);
 ofstream ofile("TEMP.DAT", ios::binary);

 while(ifile.read((char*)&Patient,sizeof(Patient)))
 {
            if(Patient.PatientID!=PID)
              ofile.write((char*)&Patient, sizeof(Patient));

            else
              {
                        cout<<"Patient ID       : "<<Patient.PatientID<<endl;
                        cout<<"Patient Name     : "<<Patient.Name<<endl;
```

```
                    cout<<"Patient Ward No. : "<<Patient.WardNo<<endl;
                    cout<<"Patient Status   : "<<Patient.Status<<endl;

                    cout<<"\nModify patient name? (Y/N): ";
                    cin>>YN;

                    if (YN=='Y'||YN=='y')
                    {
                     cout<<"Enter modified Patient Name: ";
                     gets(Patient.Name);
                    }

            cout<<"\nModify patient Ward No.? (Y/N): ";
            cin>>YN;

            if (YN=='Y'||YN=='y')
            {
                    cout<<"Enter modified Ward No.: ";
                    cin>>Patient.WardNo;
            }


            ofile.write((char*)&Patient, sizeof(Patient));
            found=1;
           }
 }

 if (found==0)
  cout<<"RECORD NOT FOUND"<<endl;

 ifile.close();
 ofile.close();

 remove("HOSPITAL.DAT");
 rename("TEMP.DAT", "HOSPITAL.DAT");

 design();
 cout<<"Enter any key: ";
 getch();

}
```

//Opt 5: Admit

```
void Admit()
{
 record Patient;

 int PID, found=0;
 char YN;

 ifstream ifile("HOSPITAL.DAT", ios::binary);
```

```
ofstream ofile("TEMP.DAT", ios::binary);

cout<<"Enter Patient ID: ";
cin>>PID;
cout<<endl;

while(ifile.read((char*)&Patient, sizeof(Patient)))
{
 if (Patient.PatientID!=PID)
        ofile.write((char*)&Patient, sizeof(Patient));

 else
 {
        cout<<"Patient ID       : "<<Patient.PatientID<<endl;
        cout<<"Patient Name     : "<<Patient.Name<<endl;
        cout<<"Patient Ward No. : "<<Patient.WardNo<<endl;
        cout<<"Patient Status   : "<<Patient.Status<<endl<<endl;

        if (Patient.Status=='A')
         cout<<"PATIENT ALREADY ADMITTED!"<<endl;
        else
        {
         cout<<"Admit patient? (Y/N) : ";
         cin>>YN;

         if (YN=='y'||YN=='Y')
         {
                Patient.Status='A';
                cout<<"\nPATIENT ADMITTED\n"<<endl;
         }
         else
          cout<<"\nPATIENT NOT ADMITTED\n"<<endl;



        }
        found=1;
        ofile.write((char*)&Patient, sizeof(Patient));
 }
}

if (found==0)
 cout<<"RECORD NOT FOUND"<<endl;

ifile.close();
ofile.close();

remove("HOSPITAL.DAT");
rename("TEMP.DAT", "HOSPITAL.DAT");

design();
```

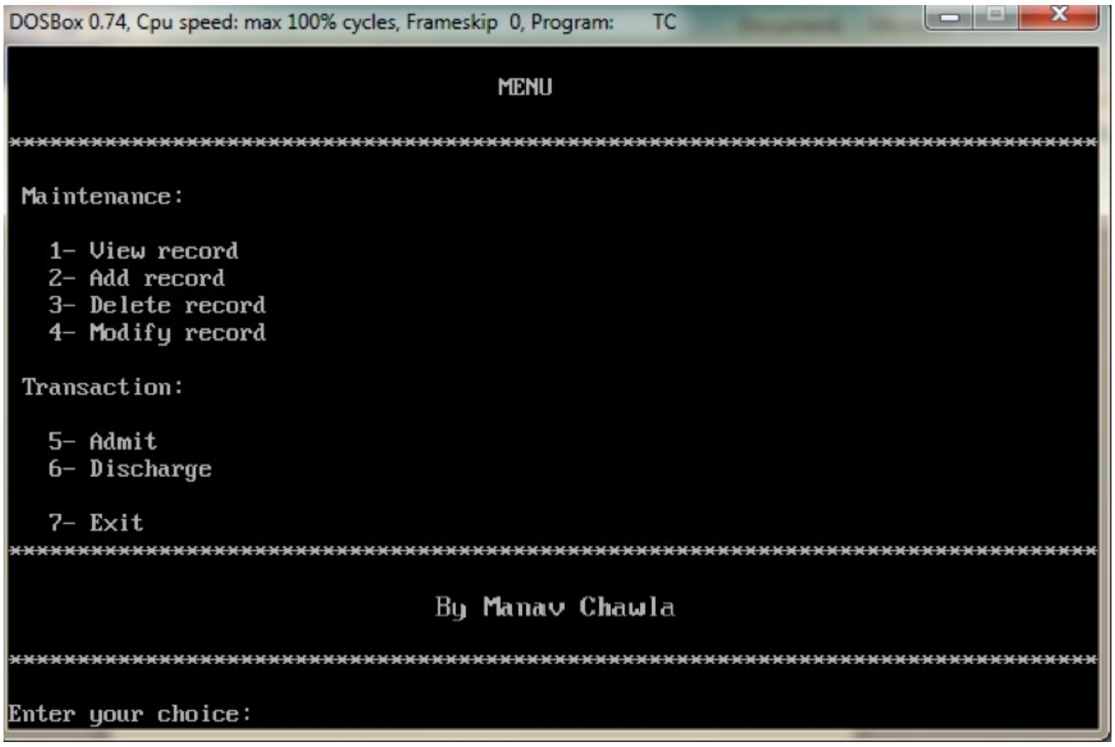
```
 cout<<"Enter any key: ";
 getch();

}
```

//Opt 6: Discharge

```
void Discharge()
{
 record Patient;

 int PID, found=0;
 char YN;

 ifstream ifile("HOSPITAL.DAT", ios::binary);
 ofstream ofile("TEMP.DAT", ios::binary);

 cout<<"Enter Patient ID: ";
 cin>>PID;
 cout<<endl;

 while(ifile.read((char*)&Patient, sizeof(Patient)))
 {
  if (Patient.PatientID!=PID)
        ofile.write((char*)&Patient, sizeof(Patient));

  else
  {
        cout<<"Patient ID       : "<<Patient.PatientID<<endl;
        cout<<"Patient Name     : "<<Patient.Name<<endl;
        cout<<"Patient Ward No. : "<<Patient.WardNo<<endl;
        cout<<"Patient Status   : "<<Patient.Status<<endl<<endl;

        if (Patient.Status=='D')
         cout<<"PATIENT ALREADY DISCHARGED!"<<endl;
        else
        {
         cout<<"Discharge patient? (Y/N) : ";
         cin>>YN;

         if (YN=='y'||YN=='Y')
         {
                Patient.Status='D';
                cout<<"PATIENT DISCHARGED\n"<<endl;
         }
         else
          cout<<"PATIENT NOT DISCHARGED\n"<<endl;



        }
```

```
            found=1;
            ofile.write((char*)&Patient, sizeof(Patient));
  }
 }

 if (found==0)
  cout<<"RECORD NOT FOUND"<<endl;

 ifile.close();
 ofile.close();

 remove("HOSPITAL.DAT");
 rename("TEMP.DAT", "HOSPITAL.DAT");

 design();

 cout<<"\nEnter any key: ";
 getch();

}
```

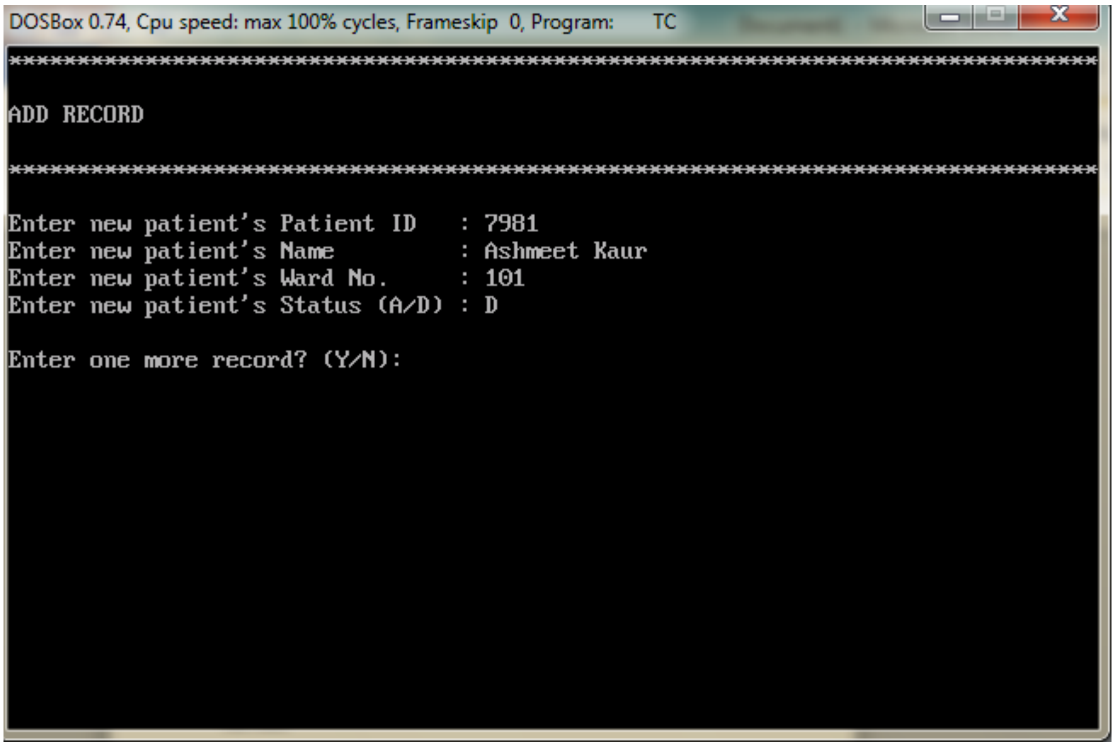
I/O Samples



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
                                        MENU

********************************************************************************

 Maintenance:

   1- View record
   2- Add record
   3- Delete record
   4- Modify record

 Transaction:

   5- Admit
   6- Discharge

   7- Exit
********************************************************************************

                               By Manav Chawla

********************************************************************************

Enter your choice:
```

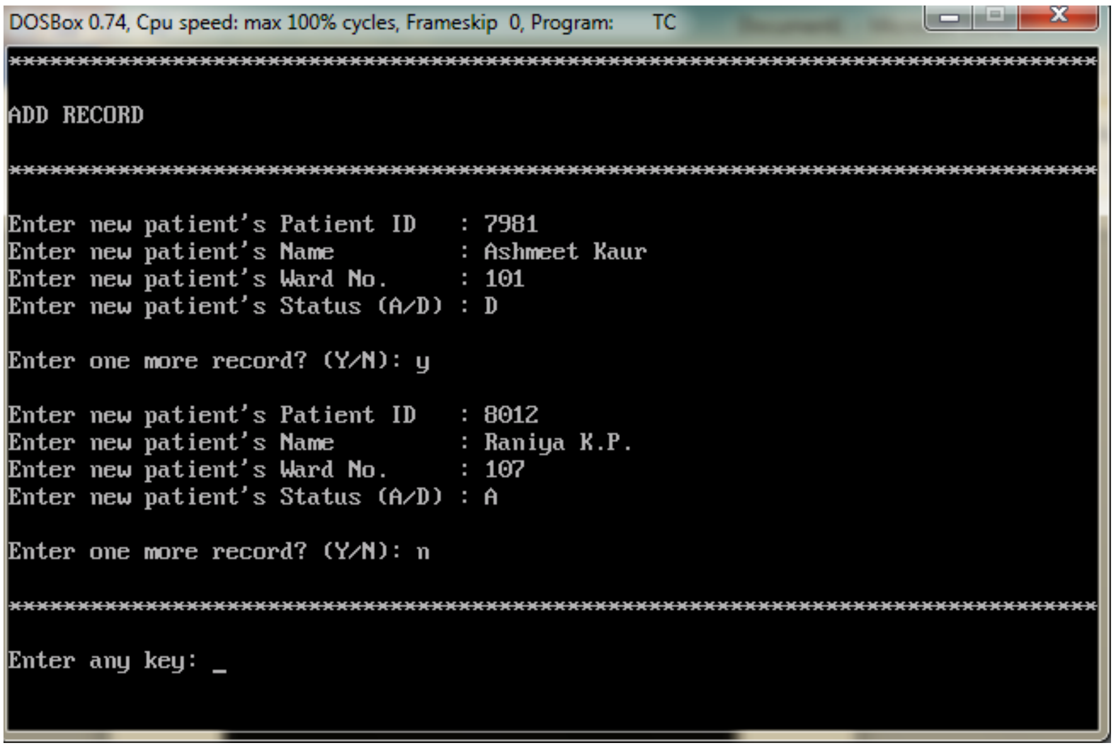
Main Menu



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

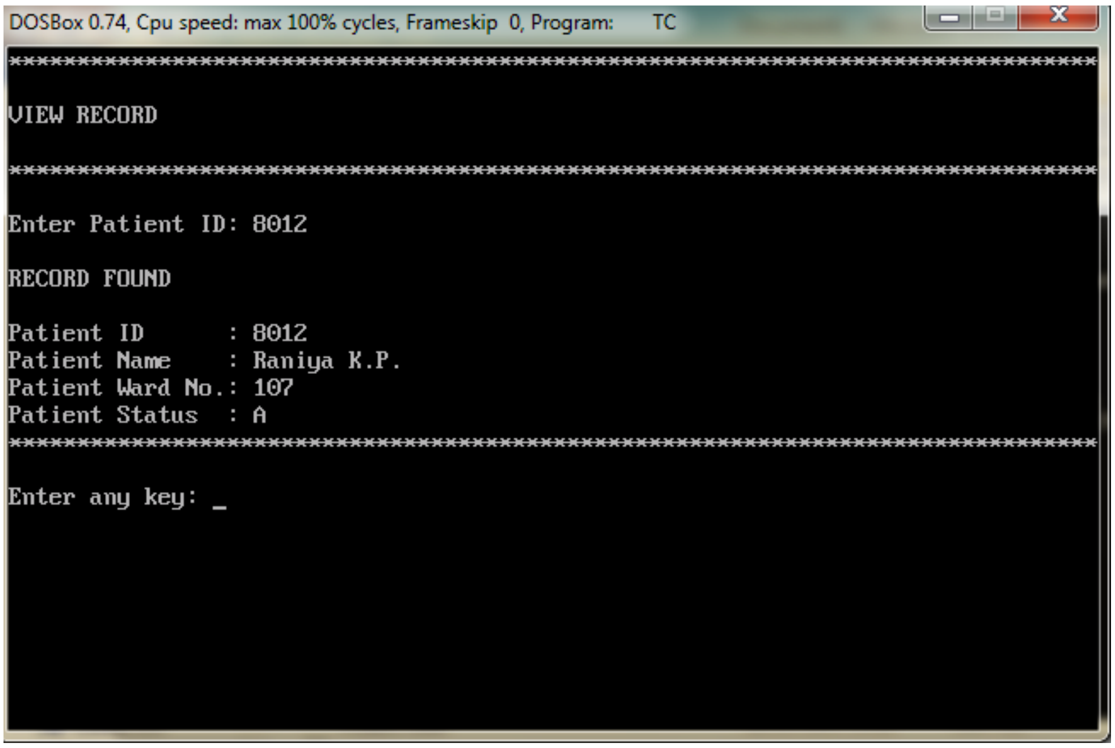
```
********************************************************************************

ADD RECORD

********************************************************************************

Enter new patient's Patient ID   : 7981
Enter new patient's Name         : Ashmeet Kaur
Enter new patient's Ward No.     : 101
Enter new patient's Status (A/D) : D

Enter one more record? (Y/N):
```

Working of the Add Record option (1)


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

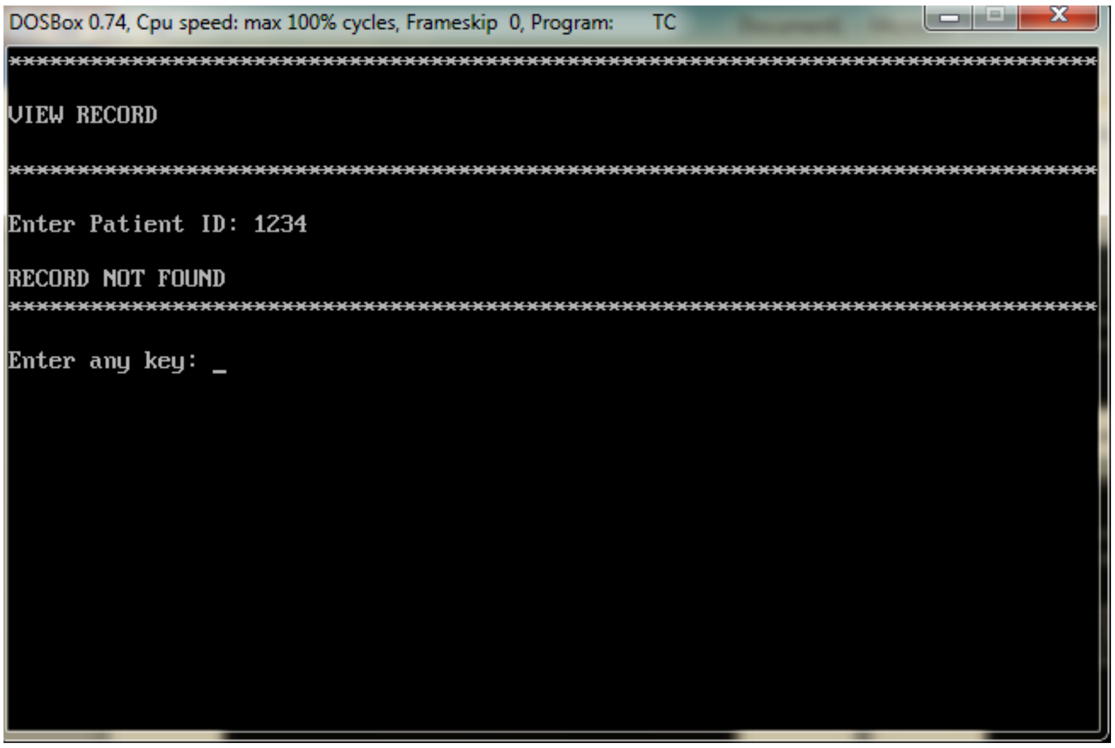
```
********************************************************************************

ADD RECORD

********************************************************************************

Enter new patient's Patient ID  : 7981
Enter new patient's Name        : Ashmeet Kaur
Enter new patient's Ward No.    : 101
Enter new patient's Status (A/D) : D

Enter one more record? (Y/N): y

Enter new patient's Patient ID  : 8012
Enter new patient's Name        : Raniya K.P.
Enter new patient's Ward No.    : 107
Enter new patient's Status (A/D) : A

Enter one more record? (Y/N): n

********************************************************************************

Enter any key: _
```

Working of the Add Record option (2)


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

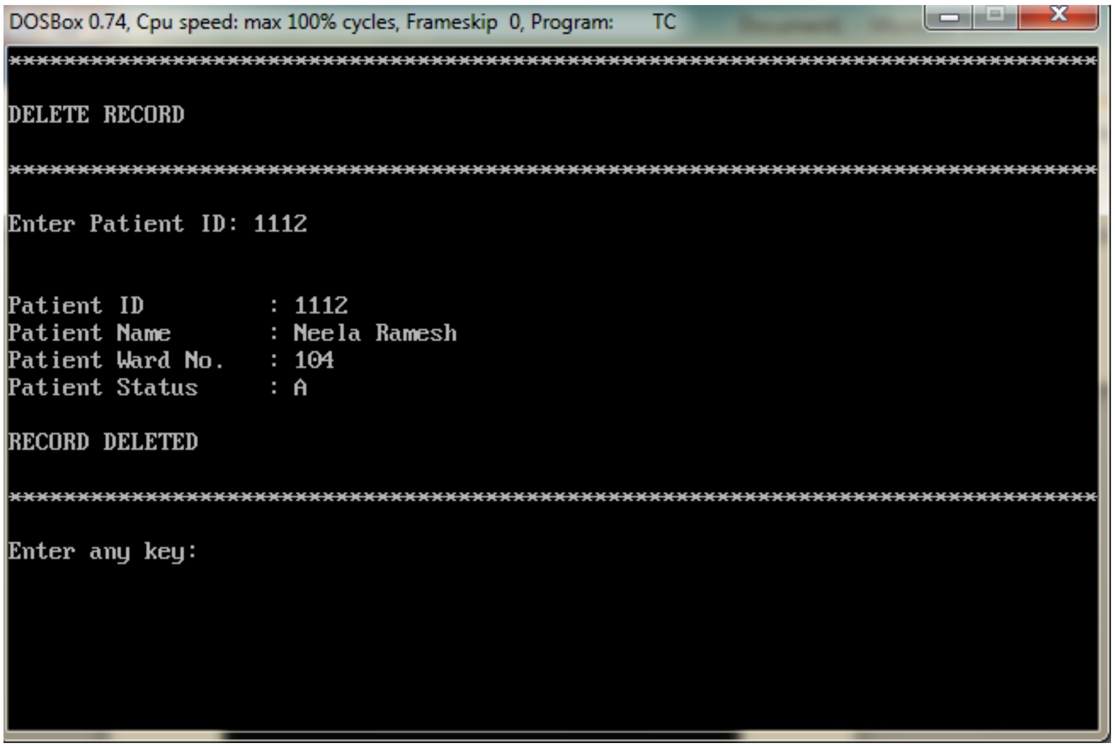
```
********************************************************************************

VIEW RECORD

********************************************************************************

Enter Patient ID: 8012

RECORD FOUND

Patient ID       : 8012
Patient Name     : Raniya K.P.
Patient Ward No.: 107
Patient Status   : A
********************************************************************************

Enter any key: _
```

Working of the View Record option (when record exists in file)


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

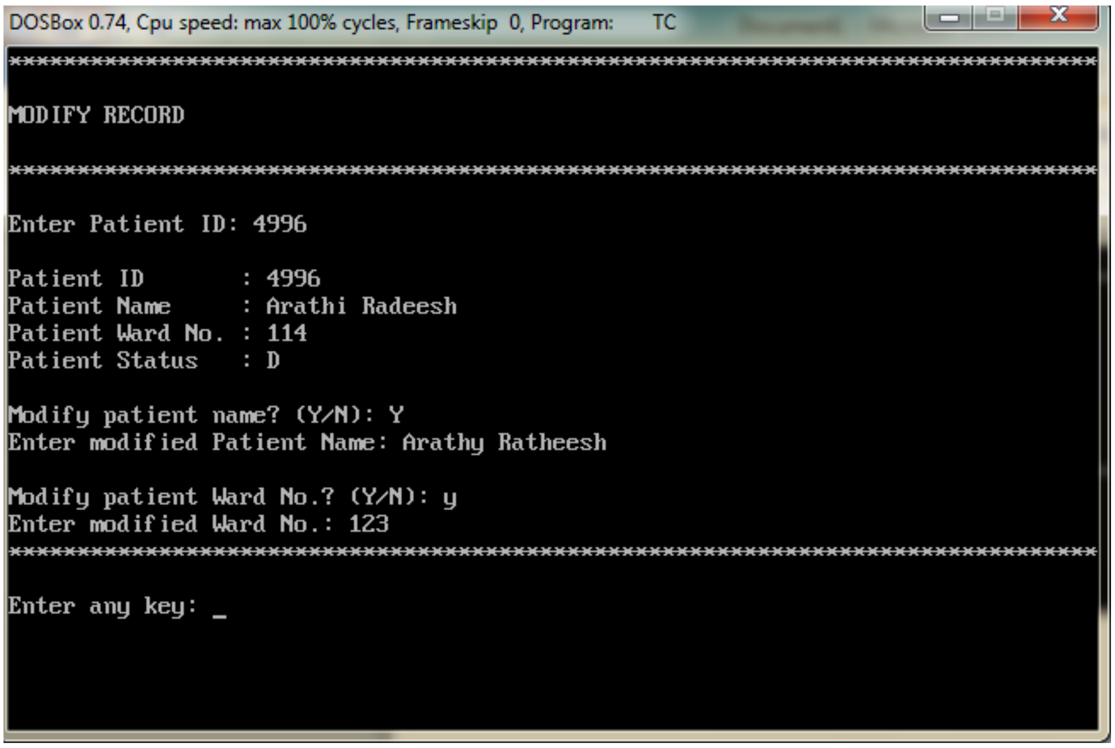
```
********************************************************************************

VIEW RECORD

********************************************************************************

Enter Patient ID: 1234

RECORD NOT FOUND
********************************************************************************

Enter any key: _
```

Working of the View Record option (when record does not exist in file)


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

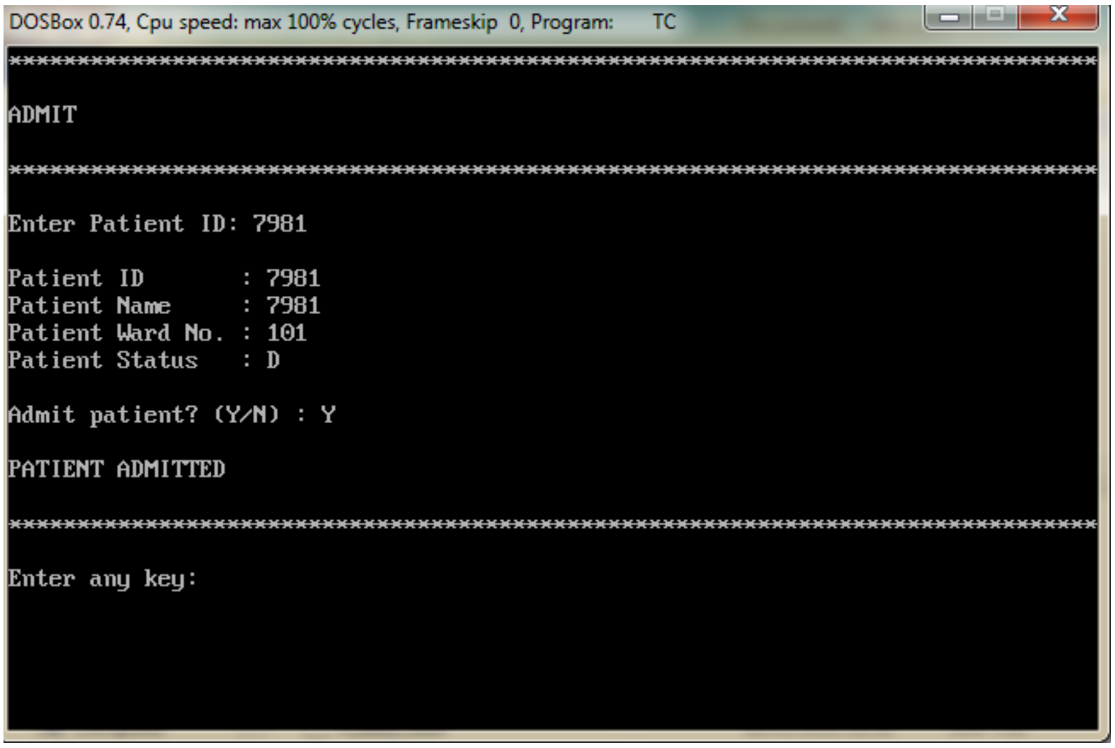
```
********************************************************************************

DELETE RECORD

********************************************************************************

Enter Patient ID: 1112


Patient ID          : 1112
Patient Name        : Neela Ramesh
Patient Ward No.    : 104
Patient Status      : A

RECORD DELETED

********************************************************************************

Enter any key:
```

Working of the Delete Record option


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

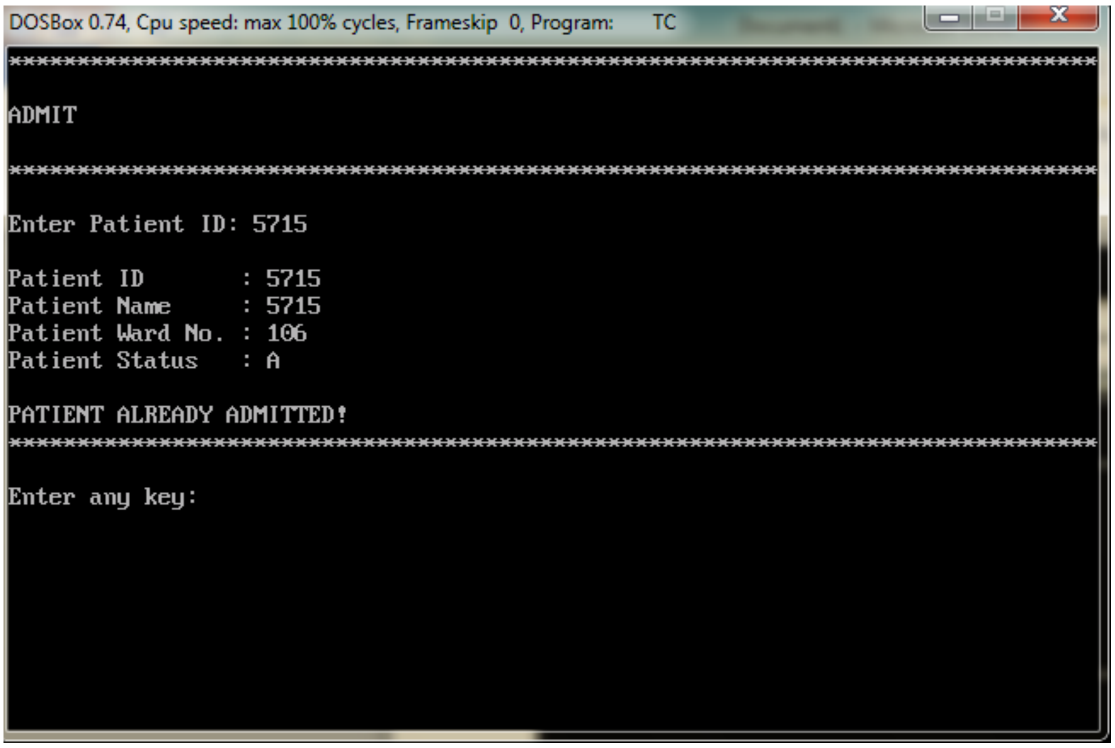
```
*******************************************************************************

MODIFY RECORD

*******************************************************************************

Enter Patient ID: 4996

Patient ID       : 4996
Patient Name     : Arathi Radeesh
Patient Ward No. : 114
Patient Status   : D

Modify patient name? (Y/N): Y
Enter modified Patient Name: Arathy Ratheesh

Modify patient Ward No.? (Y/N): y
Enter modified Ward No.: 123
*******************************************************************************

Enter any key: _
```

Working of the Modify Record option (when both patient name and ward no. are modified)


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

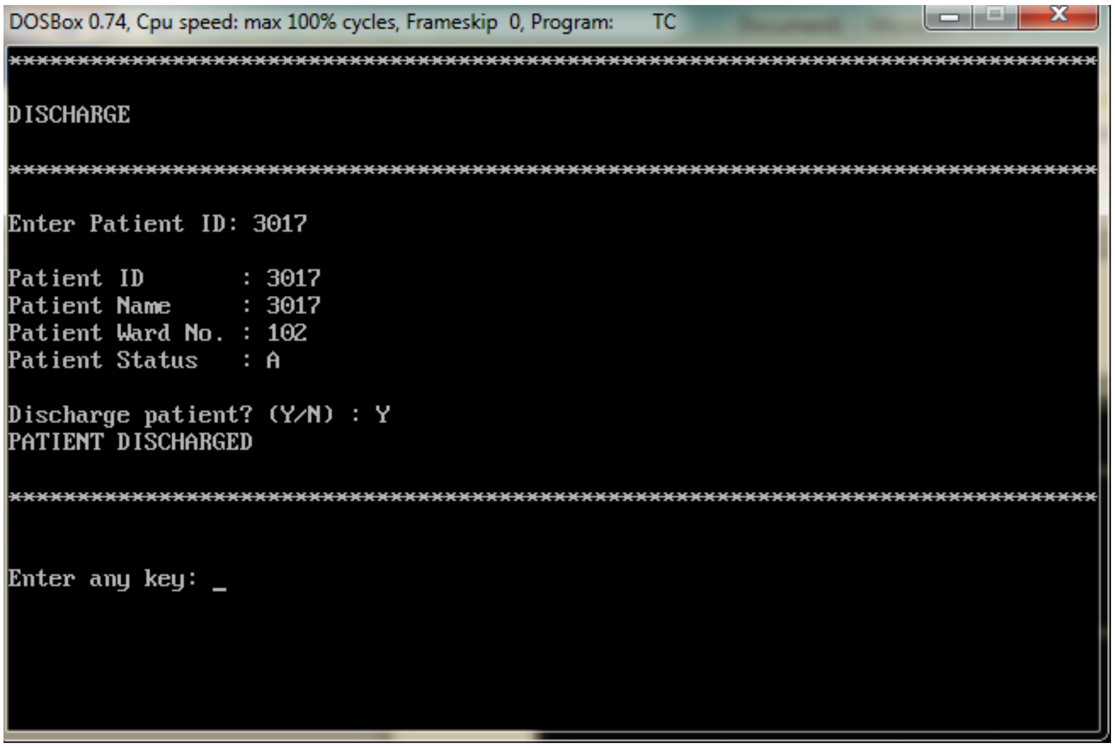
```
*******************************************************************************

ADMIT

*******************************************************************************

Enter Patient ID: 7981

Patient ID       : 7981
Patient Name     : 7981
Patient Ward No. : 101
Patient Status   : D

Admit patient? (Y/N) : Y

PATIENT ADMITTED

*******************************************************************************

Enter any key:
```

Working of the Admit option (when patient Status is ‘D’)


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

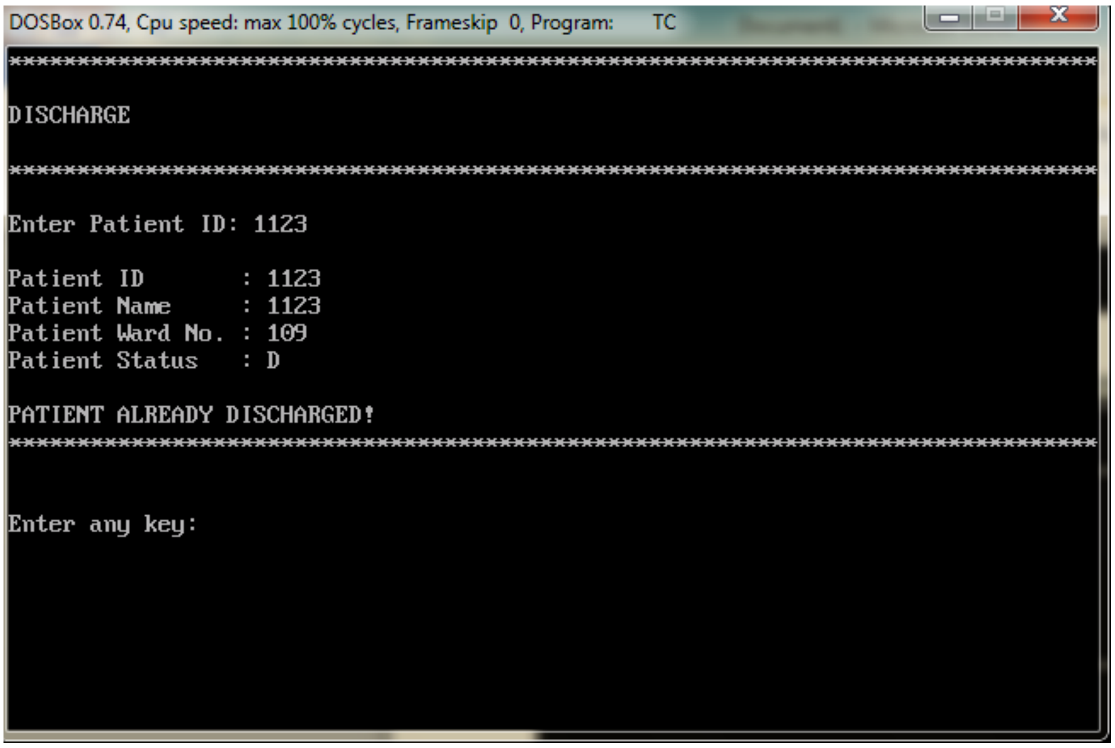
```
********************************************************************************

ADMIT

********************************************************************************

Enter Patient ID: 5715

Patient ID       : 5715
Patient Name     : 5715
Patient Ward No. : 106
Patient Status   : A

PATIENT ALREADY ADMITTED!
********************************************************************************

Enter any key:
```

Working of Admit option (when patient status is ‘A’)


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
********************************************************************************

DISCHARGE

********************************************************************************

Enter Patient ID: 3017

Patient ID       : 3017
Patient Name     : 3017
Patient Ward No. : 102
Patient Status   : A

Discharge patient? (Y/N) : Y
PATIENT DISCHARGED

********************************************************************************


Enter any key: _
```

Working of Discharge option (when patient status is ‘A’)



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
********************************************************************************

DISCHARGE

********************************************************************************

Enter Patient ID: 1123

Patient ID       : 1123
Patient Name     : 1123
Patient Ward No. : 109
Patient Status   : D

PATIENT ALREADY DISCHARGED!
********************************************************************************


Enter any key:
```

Working of the Discharge option (when patient status is ‘D’)

Bibliography

- Theory file notes
- *Computer Science with C++* by Sumita Arora

“IT IS BETTER TO HAVE CODED AND DEBUGGED, THAN TO HAVE NEVER CODED AT ALL.”