U19CS076 DBMS ASSIGNMENT 2

CODE:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//doctor structure

struct doctor{

            int id;

            char name[20];

            int age;

            char gender;

            char area[20];

};

struct date{

 int day;

 int month;

 int year;

};

struct patient{

//to analyse patient doctor relationship

            int p_id;

            int d_id;

            char p_name[20];
            struct date adm_dt;

            struct date dis_dt;

            char p_area[20];

};

//global variable

            int p_count[30];//max number of doctors is 30

void insertdata()

{

            printf("\n1.Add record in doctor file\n2.Add patient-doctor relation data in file\n3.Add patient record data \nChoose a menu option : ");

            int k;scanf("%d",&k);

            if(k<1 ||k>2)

            printf("Wrong/invalid input");

 FILE *fp;

 if(k==1)

 {

            struct doctor d;

   fp = fopen("doctordata.txt","a+");

   if(fp==NULL)

   {

            printf("Failed to load! ");

           return;

   }

             printf("\nEnter Doctor ID: ");
```

```
      scanf("%d", &d.id);
  printf("\nEnter name name: ");
  fflush(stdin);
      scanf("%s", &d.name);
  printf("\n Enter gender (M/F): ");
  fflush(stdin);
  scanf("%c", &d.gender);
   printf("\n Enter age :");
  fflush(stdin);
  scanf("%d", &d.age);
  printf("\n Enter area of residency :");
  fflush(stdin);
  scanf("%s", &d.area);
  fwrite(&d,sizeof(struct doctor),1,fp);
  p_count[d.id]=0;
  printf("\nSUCCESSFULY PRINTED DATA IN FILE doctordatadata.txt\n");
  fclose(fp);

}
else if(k==2)
{
        struct patient p;int k;

 fp = fopen("patientdata.txt","a+");
```

```
if(fp==NULL)
{
        printf("Failed to load! ");
       return;
}
         printf("\nEnter Patient ID: ");
     scanf("%d", &p.p_id);
printf("\nEnter Doctor ID: ");
int t,c=0;
scanf("%d",&t);
struct doctor d;
FILE *fp2=fopen("doctordata.txt","r+");
while(fread(&d,sizeof(struct doctor),1,fp2))
{

        if(t==d.id)
                   {
                   p.d_id=t;

                    c=1;
                   break;
                              }
        }
```

```
            if(c==0)
                    printf("Doctor ID doesnt exist");
            else
            {
            printf("Enter patient name :");
            fflush(stdin);
          scanf("%s", &p.p_name);
            printf("Enter patient area :");
            fflush(stdin);
          scanf("%s", &p.p_area);
            printf("Enter admission date :");
            fflush(stdin);
       scanf("%d %d %d", &p.adm_dt.day, &p.adm_dt.month, &p.adm_dt.year);
            printf("Enter discharge date :");
            fflush(stdin);
       scanf("%d %d %d", &p.dis_dt.day, &p.dis_dt.month, &p.dis_dt.year);
                     printf("\nSUCCESSFULY PRINTED DATA IN FILE patient_doc record data.txt\n");
            fwrite(&p,sizeof(struct patient),1,fp);


            }
      fclose(fp);
    }
  }
```

```
void deletedata()
{       printf("1>Delete from doctor records");
        printf("2>Delete from patient records");
        int k;scanf("%d",&k);
        if(k==1)
        {
                struct doctor d;
        FILE *fp;
        FILE *fp1;
        fp=fopen("doctordata.txt","r+");
        fp1=fopen("copydocdata.txt","a+");
        if(fp==NULL)
  {
        printf("Failed to load/Empty records!");
        return;
  }
        int n,f=0;
        printf("\nEnter the Doctor ID to be deleted : ");
        scanf("%d",&n);
          while(fread(&d,sizeof(struct doctor),1,fp))
          {
                        if(d.id!=n)
                  {
                        fwrite(&d,sizeof(struct doctor),1,fp1);
                        }
                        else
                {
                f=1;
                }
           }
           if(f==1)
   printf("Successfully Deleted record with ID %d",n);
  else
   printf("NO doctor data exists with ID %d",n);
           fclose(fp);
           fclose(fp1);
           remove("doctordata.txt");
  rename("copydocdata.txt","patientdata.txt");
        }
        else if(k==2)
        {
                                 struct patient p;
        FILE *fp;
        FILE *fp1;
        fp=fopen("patientdata.txt","r+");
        fp1=fopen("copypatdata.txt","a+");
        if(fp==NULL)
{
        printf("Failed to load/Empty records!");
        return;
}
        else
                    {
                    int n,f=0;
                    printf("\nEnter the Patient Admission Number to be deleted : ");
                    scanf("%d",&n);
                      while(fread(&p,sizeof(struct patient),1,fp))
                      {
                              if(p.p_id!=n)
                      {
                              fwrite(&p,sizeof(struct patient),1,fp1);
                      }
                      else
                {
                f=1;
                }
                      }
                      if(f==1)
                printf("Successfully Deleted record with ID %d",n);
               else
                printf("NO patient data exists with ID %d",n);
                      fclose(fp);
```

```
                              fclose(fp1);

                              remove("patientdata.txt");

                    rename("copypatdata.txt","patientdata.txt");

                           }

             }

}

void display()

{


    struct doctor d;

    struct patient p;

    FILE *fp1,*fp2;

    fp1=fopen("doctordata.txt","r+");

    fp2=fopen("patientdata.txt","r+");


    if(fp1==NULL ||fp2==NULL)

    {

    printf("Failed to load file / file is empty!!!");

    }

    else

    {

        printf("Printing all doctor records");

    while(fread(&d,sizeof(struct doctor),1,fp1))

    {
```

```
  printf("\nDoctor ID : %d",d.id);
  printf("\n Name : %s",d.name);
  printf("\nGender : %c",d.gender);
  printf("\nAge : %d",d.age);
  printf("\nArea : %s",d.area);
  printf("\n======================================\n");
       }
       printf("\n\t\t*********************************************************\t\t\n");
   printf("Printing all patient records");
  while(fread(&p,sizeof(struct patient),1,fp2))
  {
  printf("\nPatient ID : %d",p.p_id);
  printf("\nDoctor ID : %d",p.d_id);
    printf("\nFirst Name : %s",p.p_name);
  printf("\nArea :%s",p.p_area);
   printf("\nAdmission Date : %d / %d / %d",p.adm_dt.day,p.adm_dt.month,p.adm_dt.year);
  printf("\nDischarge Date : %d / %d / %d",p.dis_dt.day,p.dis_dt.month,p.dis_dt.year);
  printf("\n======================================\n");
  }

  }
  fclose(fp1);
  fclose(fp2);
}
```

```
void count()
{
        int t;
        for(t=0;t<30;t++)//function to refresh count
        {
                if(p_count[t]==0 || p_count[t]==-1)
                continue;
                else
                p_count[t]=0;
        }
                FILE *fp;
        struct patient p;
         fp=fopen("patientdata.txt","r+");
   while(fread(&p,sizeof(struct patient),1,fp))
       {
          p_count[p.d_id]++;
       }
     fclose(fp);
}

void count_patients()
{
     count();
   int i;
    for(i=0;i<30;i++)
   {
     if(p_count[i]!=-1)
          printf("\nNumber of patient for Doctor ID %d : %d",i,p_count[i]);
         }
}

void add_rec()
{
        FILE *fp1=fopen("patientdata.txt","a+");
        FILE *fp2=fopen("doctordata.txt","a+");


        struct patient p;
        struct doctor d;
        printf("Enter patient ID: ");
        fflush(stdin);
        scanf("%d",&p.p_id);
        count();
//all array with no doctor ID is -1 in p_count
        //Entering doctor id with least patients
   fclose(fp1);
        fp1=fopen("patientdata.txt","a+");
        //finding miminum element
```

```
int i,s,t=100;//max number of patients in hospital is 100


char n[20];

strcpy(n,d.name);

for(i=1;i<30;i++)

{

        if(p_count[i]!=-1 && p_count[i]<t)

        {

                        t=p_count[i];

                        s=i;

                        printf("%d %d",t,p_count[i]);

                        fseek(fp2,0,SEEK_SET);       //brings pointer back to start

                        while(fread(&d,sizeof(struct doctor),1,fp2))

{

 if(d.id==i)

 {

                strcpy(n,d.name);

                  }

          }

        }

        else if(p_count[i]==t && p_count[i]!=-1)

        {

                fseek(fp2,0,SEEK_SET);

                 while(fread(&d,sizeof(struct doctor),1,fp2))
```

```
     {
       if(d.id==i)
       {
                if(strcmp(d.name,n)<0)
                {
                          strcpy(n,d.name);
                          t=p_count[i];
                          s=i;
                                    }
                            }
     }
                }
      }
      p.d_id=s;
                printf("Enter patient name :");
      fflush(stdin);
    scanf("%s", &p.p_name);
      printf("Enter patient area :");
      fflush(stdin);
    scanf("%s", &p.p_area);
      printf("Enter admission date :");
      fflush(stdin);
scanf("%d %d %d", &p.adm_dt.day, &p.adm_dt.month, &p.adm_dt.year);
      printf("Enter discharge date :");
```

```
        fflush(stdin);
   scanf("%d %d %d", &p.dis_dt.day, &p.dis_dt.month, &p.dis_dt.year);
        fwrite(&p,sizeof(struct patient),1,fp1);
        printf("**Record added with %d patient Id and %d Doctor id**",p.p_id,p.d_id);
        p_count[t]++;
        fclose(fp1);
        fclose(fp2);
}

void summary()
{       printf("\nGenerate various summary reports DOCTOR wise on the field given by the
user");
        printf("\n1> Display count of all patients");
        printf("\n2> Display count of patient based on gender of doctor");
        printf("\n3> Display age wise count of patient based on doctor");
        printf("\n4> Display count of patient areawise");
        printf("\nEnter an option : ");
        int c;
        scanf("%d",&c);
                FILE *fp1=fopen("patientdata.txt","a+");
        FILE *fp2=fopen("doctordata.txt","a+");
        int count=0,male=0,female=0;
        struct patient p;
        struct doctor d;
```

```
 switch(c)
 {
           case 1:
           {
                      while(fread(&p,sizeof(struct patient),1,fp1))
                       {
                        count++;
                       }
                        printf("Total number of patients are %d",count);
           }break;
           case 2:
                     {
                     //        fseek(fp1,0,SEEK_SET);
                               while(fread(&d,sizeof(struct doctor),1,fp2))
            {
           if(d.gender=='M' || d.gender=='m')
             male+=p_count[d.id];
           else if(d.gender=='F' || d.gender=='f')
             female+=p_count[d.id];
 }
printf("\nTotal number of patients treated by male Doctors are %d",male);
printf("\nTotal number of patients treated by female Doctors  are %d",female);
int other=0;
if(count-male-female>0)
```

```
  other=count-male-female;
printf("\nTotal number of  patients treated by Other gender Doctors  are %d\n",other);
                }break;
        case 3:
                {
                        //      fseek(fp1,0,SEEK_SET);
                                fseek(fp2,0,SEEK_SET);
                                int age[100]={0};
                                 while(fread(&d,sizeof(struct doctor),1,fp2))
                                 {
                                                 age[d.age]+=p_count[d.id];
                                 }
                                  int i=1;
for(i;i<100;i++)

{
if(age[i]!=0)
printf("\nNumber of people of age %d : %d",i,age[i]);
}


                }break;
        case 4:
                {
                struct doctor newd;
```

```
           int c,f=0,v=0;

                                                            fseek(fp2,0,SEEK_SET);

char word[20],visited[count][20];      //finding all unique areas

while(fread(&d,sizeof(struct doctor),1,fp2))

{

   c=0;

   f=0;//flag

   strcpy(word,d.area);int i;

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

   {

      if(strcmp(word,visited[i])==0)

      {

         f=1;

      }

   }

   if(f==1)    //word already exists in visited array

   {

      continue;

   }

   else

   {

      strcpy(visited[v],word);//add word is visited array

      v++;

   }
```

```
          FILE *fpnew;

          fpnew=fopen("doctordata.txt","r+");

          while(fread(&newd,sizeof(struct doctor),1,fpnew))

          {

          if(strcmp(word,newd.area)==0)

          c+=p_count[newd.id];

          }

          fclose(fpnew);


          printf("\nNumber of people in area %s : %d",word,c);

                        }

                        }break;

                default:printf("Invalid Input\n");

        }

        fclose(fp1);

        fclose(fp2);

}


void list()

{

        printf("\nList all the patient records of the file in ascending order");

        printf("\n1> Admitted on same date");

        printf("\n2> Discharged on same date");

        printf("\n3> Patients treated under the doctor");
```

```
    printf("\n4> Patients treated under the male doctor");

    printf("\n5> Patients treated under the female doctor area wise");

    printf("\nEnter an option : ");

    int c;

    scanf("%d",&c);

    switch(c)

    {

        case 1:

        {

            struct date d;

            struct patient p,pat[20],temp;

            int cnt=0,i,j,flag;

            FILE *fp;

            printf("\nEnter the admission date:");

            scanf("%d %d %d",&d.day,&d.month,&d.year);

            fp=fopen("patientdata.txt", "r+");

            if(fp==NULL)

                printf("Failed to open file");

            else

            {

                while(fread(&p,sizeof(struct patient),1,fp))

                {

                    if(p.adm_dt.day==d.day && p.adm_dt.month==d.month
&&
        p.adm_dt.year==d.year)
        {
        pat[cnt]=p;
        cnt++;
        }
    }
}
fclose(fp);
if(cnt==0)
{
    printf("\nNo such record found!!");
}
else
{
    printf("\nASCENDING ORDER ACCORDING TO PATIENT NAME");
    for(i=1;i<cnt;i++) //Arranging in ascending order
    {
        flag=0;
        for(j=0;j<cnt-i;j++)
        {
            if(strcmp(pat[j+1].p_name,pat[j].p_name)<0)
            {
                temp=pat[j];
                pat[j]=pat[j+1];
```

```
                        pat[j+1]=temp;
                        flag=1;
                    }
                }
                if(flag==0)
                break;
            }
            for(i=0;i<cnt;i++)
            {
                printf("\n\nPatient ID:%d",pat[i].p_id);
                printf("\nDoctor ID:%d",pat[i].d_id);
                printf("\nName:%s",pat[i].p_name);
                   printf("\nArea :%s",p.p_area);
                printf("\nAdmission Date:");

printf("%d/%d/%d",pat[i].adm_dt.day,pat[i].adm_dt.month,pat[i].adm_dt.year);
                printf("\nDischarge Date:");

printf("%d/%d/%d",pat[i].dis_dt.day,pat[i].dis_dt.month,pat[i].dis_dt.year);
            }
        }
    }break;
    case 2:
        {
                struct date d;
struct patient p,pat[20],temp;

int cnt=0,i,j,flag;

FILE *fp;

printf("\nEnter the discharge date:");

scanf("%d %d %d",&d.day,&d.month,&d.year);

fp=fopen("patientdata.txt", "r+");

if(fp==NULL)

        printf("Failed to open file");

else

{

        while(fread(&p,sizeof(struct patient),1,fp))

        {

                if(p.dis_dt.day==d.day && p.dis_dt.month==d.month && p.dis_dt.year==d.year)

                {

                pat[cnt]=p;

                cnt++;

                }

        }

}

fclose(fp);

if(cnt==0)

{

        printf("\nNo such record found!!");
```

```
}
else
{
    printf("\nASCENDING ORDER ACCORDING TO PATIENT NAME");
    for(i=1;i<cnt;i++) //Arranging in ascending order
    {
        flag=0;
        for(j=0;j<cnt-i;j++)
        {
            if(strcmp(pat[j+1].p_name,pat[j].p_name)<0)
            {
                temp=pat[j];
                pat[j]=pat[j+1];
                pat[j+1]=temp;
                flag=1;
            }
        }
        if(flag==0)
        break;
    }
    for(i=0;i<cnt;i++)
    {
        printf("\n\nPatient ID:%d",pat[i].p_id);
        printf("\nDoctor ID:%d",pat[i].d_id);
```

```
                printf("\nName:%s",pat[i].p_name);
                   printf("\nArea :%s",p.p_area);
                printf("\nAdmission Date:");

printf("%d/%d/%d",pat[i].adm_dt.day,pat[i].adm_dt.month,pat[i].adm_dt.year);
                printf("\nDischarge Date:");

printf("%d/%d/%d",pat[i].dis_dt.day,pat[i].dis_dt.month,pat[i].dis_dt.year);
            }
        }
        }break;
    case 3:
        {
            struct patient pat[10],p,pd,temp;
            struct doctor d;
        FILE *fp;
        fp=fopen("doctordata.txt","r+");
        if(fp==NULL)
        {
                    printf("Failed to open file");
                    break;
        }

            while(fread(&d,sizeof(struct doctor),1,fp))
            {
```

```
            printf("\n***************************DOCTOR ID:%d",d.id);
            printf("\n***************************DOCTOR
NAME:%s",d.name);
            FILE *fp2;
            fp2=fopen("patientdata.txt","r+");
            int P[10],cnt=0,i,j=0,flag;
            while(fread(&p,sizeof(struct patient),1,fp2))
            {
                if(d.id==p.d_id)
                {
                    P[cnt]=p.p_id;
                    cnt++;
                }
            }
            fseek(fp2,0,SEEK_SET);
            while(fread(&p,sizeof(struct patient),1,fp2))
            {
                for(i=0;i<cnt;i++)
                {
                    if(P[i]==p.p_id)
                    {
                        pat[j]=p;
                        j++;
                    }
```

```
    }
}
fclose(fp2);
printf("\nPATIENTS TREATED UNDER DOCTOR:");
if(cnt==0)
    printf("None\n");
for(i=1;i<cnt;i++)
//Arranging in ascending order according to name
{
    flag=0;
    for(j=0;j<cnt-i;j++)
    {
        if(strcmp(pat[j+1].p_name,pat[j].p_name)<0)
        {
            temp=pat[j];
            pat[j]=pat[j+1];
            pat[j+1]=temp;
            flag=1;
        }
    }
if(flag==0)
break;
}
    for(i=0;i<cnt;i++)
```

```
                {
                    printf("\n\nPatient ID:%d",pat[i].p_id);
                    printf("\n\nDoctor ID:%d",pat[i].d_id);
                printf("\nName:%s",pat[i].p_name);
                  printf("\nArea :%s",p.p_area);
                    printf("\nAdmission Date:");

printf("%d/%d/%d",pat[i].adm_dt.day,pat[i].adm_dt.month,pat[i].adm_dt.year);
                    printf("\nDischarge Date:");

printf("%d/%d/%d",pat[i].dis_dt.day,pat[i].dis_dt.month,pat[i].dis_dt.year);
                }
            }
        }break;
    case 4:
        {
                struct patient pat[10],p,temp;
                struct doctor d;
                FILE *fp;
                fp=fopen("doctordata.txt","r+");
                if(fp==NULL)
                {
                printf("Failed to open file");break;
                }
                while(fread(&d,sizeof(struct doctor),1,fp))
```

```
                    {
                    if(d.gender=='M' || d.gender=='m')
                    {
                        printf("\n***************************DOCTOR
ID:%d",d.id);
                        printf("\n***************************DOCTOR
NAME: %s",d.name);
                        FILE *fp2;
                        fp2=fopen("patientdata.txt","r+");
                        int P[10],cnt=0,i,j=0,flag;
                        while(fread(&p,sizeof(struct patient),1,fp2))
                        {
                            if(d.id==p.d_id)
                            {
                                P[cnt]=p.p_id;
                                cnt++;
                            }
                        }
                        fseek(fp2,0,SEEK_SET);
                        while(fread(&p,sizeof(struct patient),1,fp2))
                        {
                            for(i=0;i<cnt;i++)
                            {
                                if(P[i]==p.p_id)
                                {
                                    pat[j]=p;
                                    j++;
                                }
                            }
                        }
                        fclose(fp2);
                        printf("\nPATIENTS TREATED UNDER DOCTOR:");
                        if(cnt==0)
                        printf("No records\n");
                        for(i=1;i<cnt;i++) //Arranging in ascending order
according to name
                        {
                            flag=0;
                            for(j=0;j<cnt-i;j++)
                            {

    if(strcmp(pat[j+1].p_name,pat[j].p_name)<0)
                                {
                                    temp=pat[j];
                                    pat[j]=pat[j+1];
                                    pat[j+1]=temp;
                                    flag=1;
                                }
                            }
                            if(flag==0)
                            break;
                        }
                    for(i=0;i<cnt;i++)
                    {
                        printf("\n\nPatient ID:%d",pat[i].p_id);
                        printf("\n\nDoctor ID:%d",pat[i].d_id);
                    printf("\nName:%s",pat[i].p_name);
                       printf("\nArea :%s",p.p_area);
                        printf("\nAdmission Date:");

    printf("%d/%d/%d",pat[i].adm_dt.day,pat[i].adm_dt.month,pat[i].adm_dt.year);
                        printf("\nDischarge Date:");

    printf("%d/%d/%d",pat[i].dis_dt.day,pat[i].dis_dt.month,pat[i].dis_dt.year);
                    }
                }
             }
            }break;
        case 5:
            {
                struct patient pat[10],p,temp;
                struct doctor d;
                FILE *fp;
                fp=fopen("doctordata.txt","r+");
                if(fp==NULL)
```

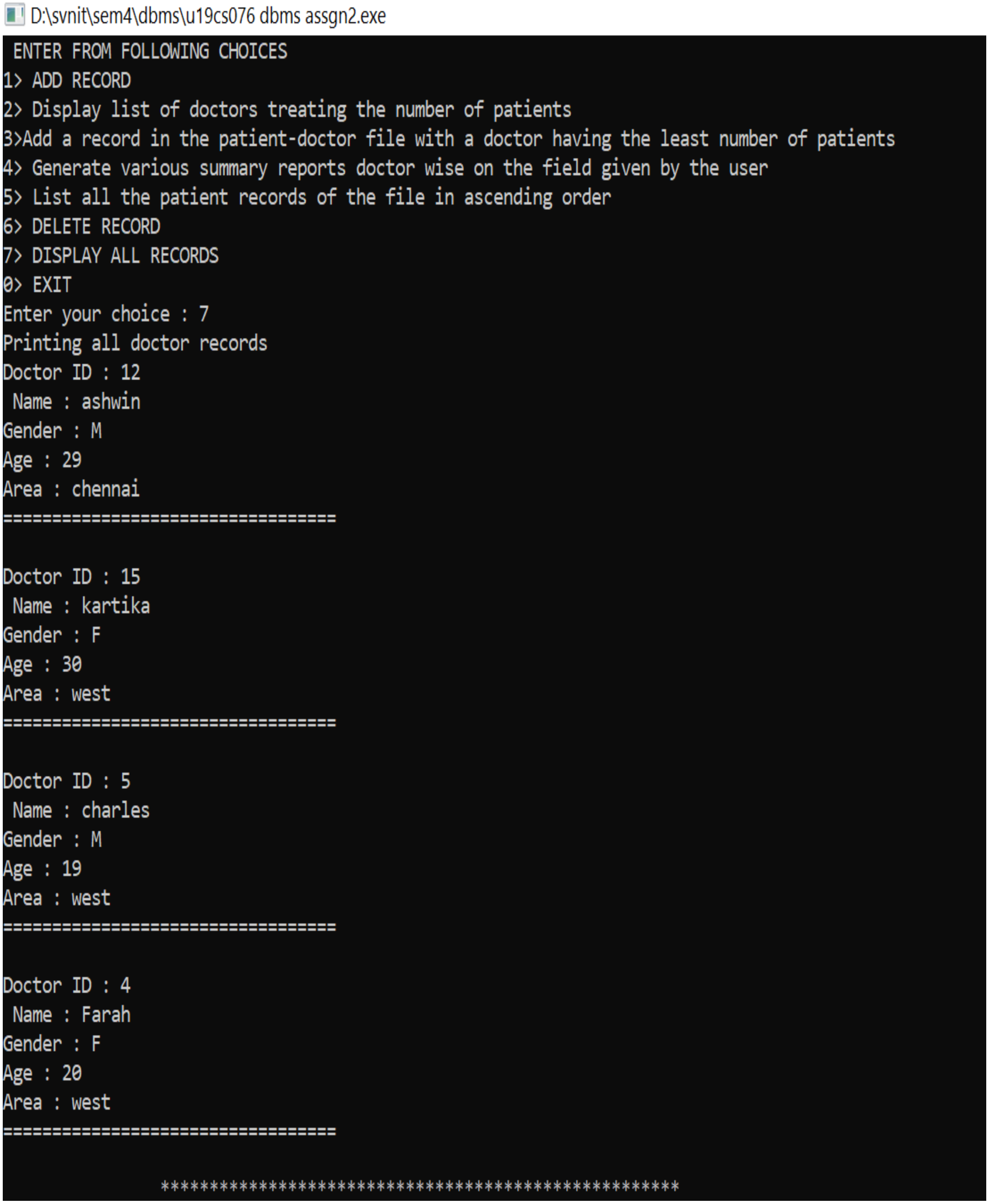
```
{
printf("Failed to open file");break;
}
while(fread(&d,sizeof(struct doctor),1,fp))
{
if(d.gender=='F' || d.gender=='f')
{
        printf("\n***************************DOCTOR
ID:%d",d.id);
        printf("\n***************************DOCTOR NAME:
%s",d.name);
FILE *fp2;
fp2=fopen("patientdata.txt","r+");
int P[10],cnt=0,i,j=0,flag;
while(fread(&p,sizeof(struct patient),1,fp2))
{
        if(d.id==p.d_id)
        {
                P[cnt]=p.p_id;
                cnt++;
        }
}
fseek(fp2,0,SEEK_SET);
while(fread(&p,sizeof(struct patient),1,fp2))
{
    for(i=0;i<cnt;i++)

    {

        if(P[i]==p.p_id)

        {

            pat[j]=p;

            j++;

        }

    }

}

fclose(fp2);

printf("\nPATIENTS TREATED UNDER DOCTOR:");

for(i=1;i<cnt;i++) //Arranging in ascending order according to area

{

flag=0;

for(j=0;j<cnt-i;j++)

{

if(strcmp(pat[j+1].p_area,pat[j].p_area)<0)

{

temp=pat[j];

pat[j]=pat[j+1];

pat[j+1]=temp;

flag=1;

}

}
```

```
                if(flag==0)
                break;
                }
                    for(i=0;i<cnt;i++)
                    {
                            printf("\n\nPatient ID:%d",pat[i].p_id);
                            printf("\n\nDoctor ID:%d",pat[i].d_id);
                        printf("\nName:%s",pat[i].p_name);
                           printf("\nArea :%s",p.p_area);
                            printf("\nAdmission Date:");

    printf("%d/%d/%d",pat[i].adm_dt.day,pat[i].adm_dt.month,pat[i].adm_dt.year);
                            printf("\nDischarge Date:");

    printf("%d/%d/%d",pat[i].dis_dt.day,pat[i].dis_dt.month,pat[i].dis_dt.year);
                    }
                }
            }
            }break;
        default:printf("\nInvalid input");
    }
}
//MAIN PROGRAM
int main()
{
 FILE *fp;

 int ch;

 char c;

 int t;

            for(t=0;t<30;t++)//function to refresh count

            {

                        p_count[t]=-1;//initial value of all doctors

            }

            count();

            while(1){

            printf("\n\n ENTER FROM FOLLOWING CHOICES \n");

             printf("1> ADD  RECORD\n");

             printf("2> Display list of doctors treating the number of patients\n");

             printf("3>Add a record in the patient-doctor file with a doctor having the least number
of patients \n");//and on equal number of patient then consider the alphabetic order of
doctor’s name

             printf("4> Generate various summary reports doctor wise on the field given by the
user\n");

             printf("5> List all the patient records of the file in ascending order\n");

             printf("6> DELETE RECORD\n");

             printf("7> DISPLAY ALL RECORDS \n");

             printf("0> EXIT\n");

              printf("Enter your choice : ");

              scanf("%d",&ch);

            int k;
```

```
             switch(ch){
              case 1 :insertdata();break;
              case 2:count_patients();break;
              case 3:add_rec();break;
              case 4:summary();break;
              case 5:list();break;
              case 6:deletedata();break;
           case 7:display();break;
          case 0:exit(0);break;
          default:printf("\nInvalid Input!");
          }
          }
}
```

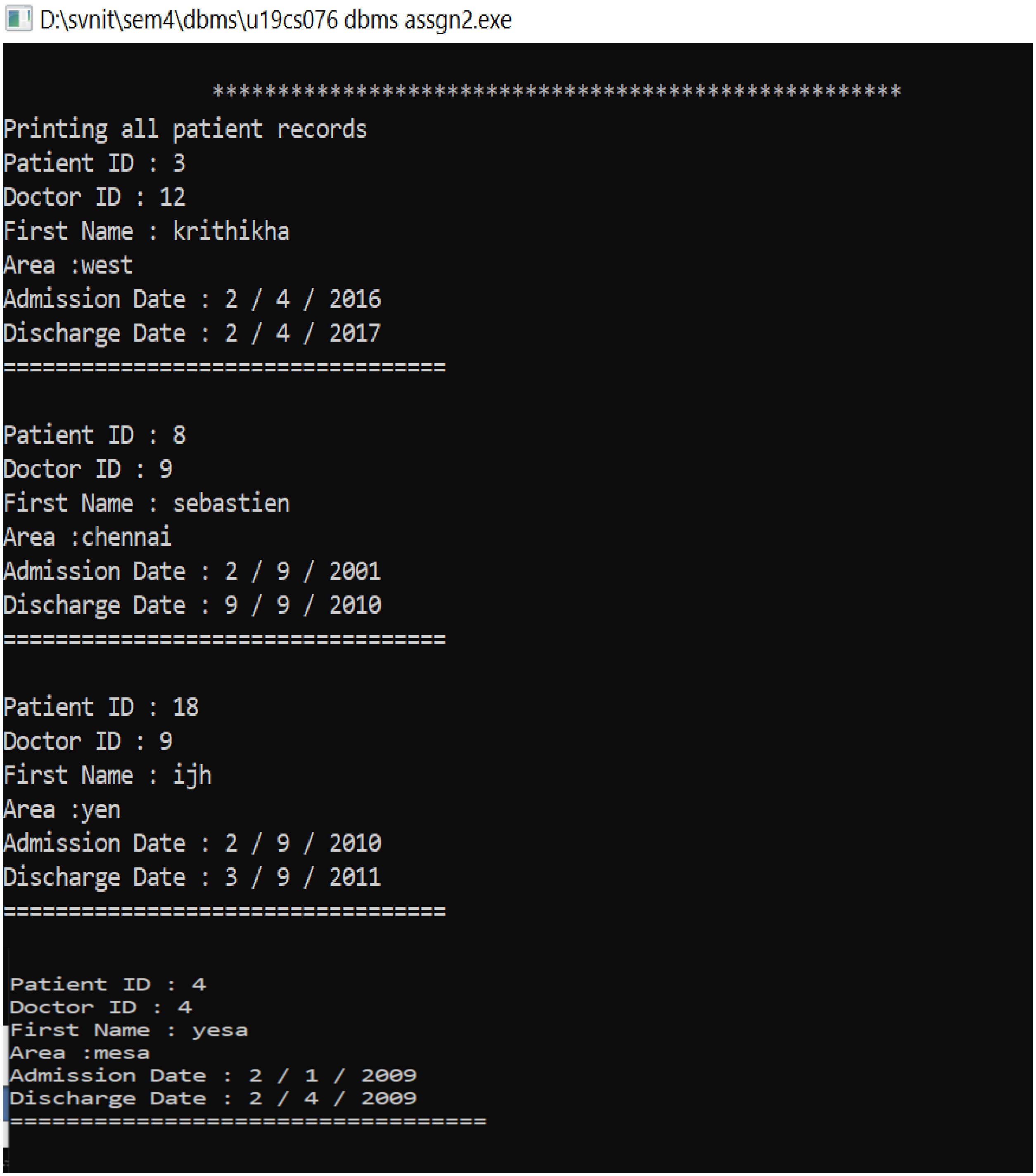
SCREENSHOTS:

2.DISPLAYING ALL RECORDS //INITIALLY



D:\svnit\sem4\dbms\u19cs076 dbms assgn2.exe

```
 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 7
Printing all doctor records
Doctor ID : 12
 Name : ashwin
Gender : M
Age : 29
Area : chennai
==================================

Doctor ID : 15
 Name : kartika
Gender : F
Age : 30
Area : west
==================================

Doctor ID : 5
 Name : charles
Gender : M
Age : 19
Area : west
==================================

Doctor ID : 4
 Name : Farah
Gender : F
Age : 20
Area : west
==================================


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

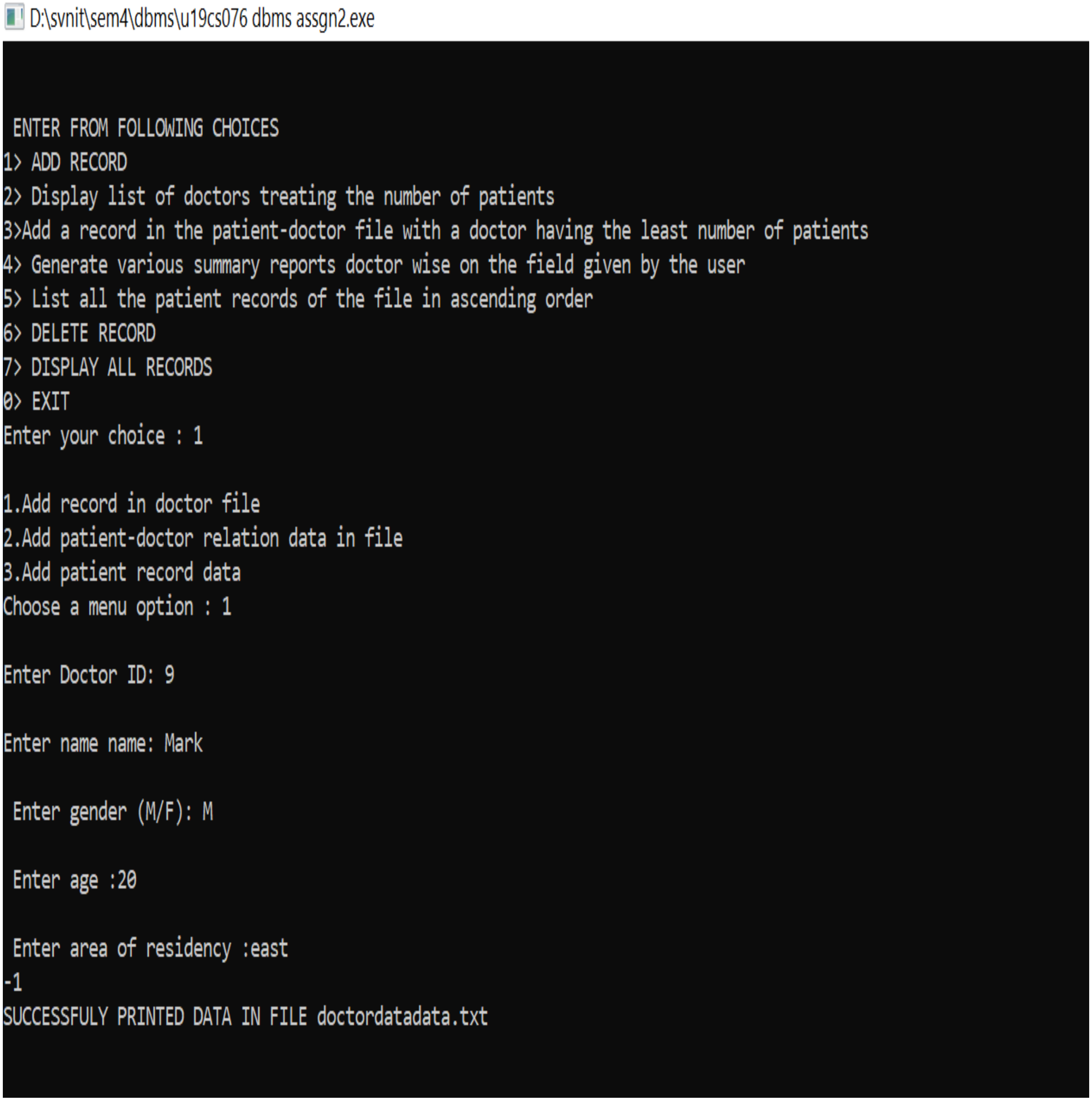


D:\svnit\sem4\dbms\u19cs076 dbms assgn2.exe

```
                ******************************************************
Printing all patient records
Patient ID : 3
Doctor ID : 12
First Name : krithikha
Area :west
Admission Date : 2 / 4 / 2016
Discharge Date : 2 / 4 / 2017
==================================

Patient ID : 8
Doctor ID : 9
First Name : sebastien
Area :chennai
Admission Date : 2 / 9 / 2001
Discharge Date : 9 / 9 / 2010
==================================

Patient ID : 18
Doctor ID : 9
First Name : ijh
Area :yen
Admission Date : 2 / 9 / 2010
Discharge Date : 3 / 9 / 2011
==================================

Patient ID : 4
Doctor ID : 4
First Name : yesa
Area :mesa
Admission Date : 2 / 1 / 2009
Discharge Date : 2 / 4 / 2009
==================================
```

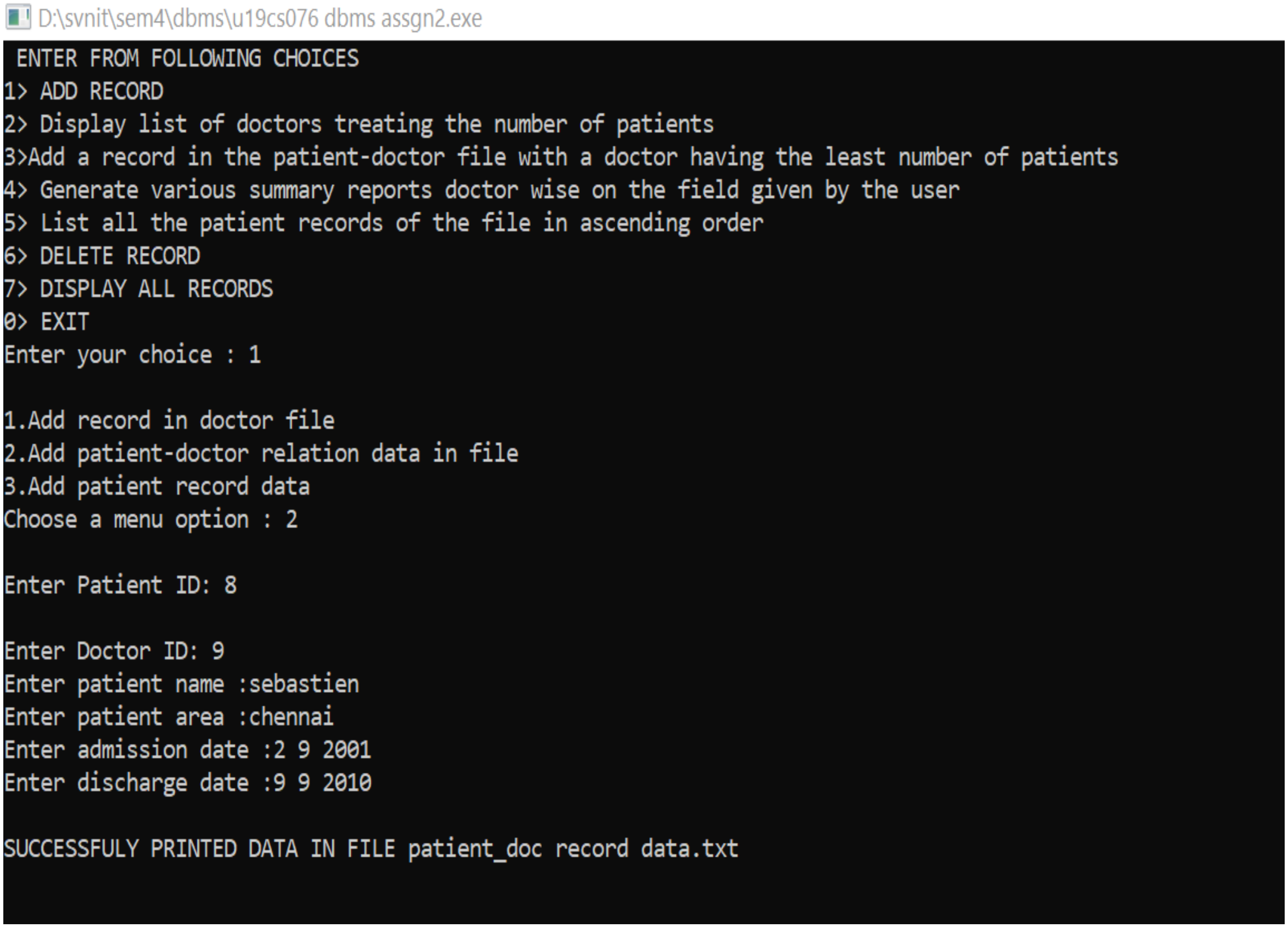
1.ADDING RECORDS

IN DOCTOR FILE



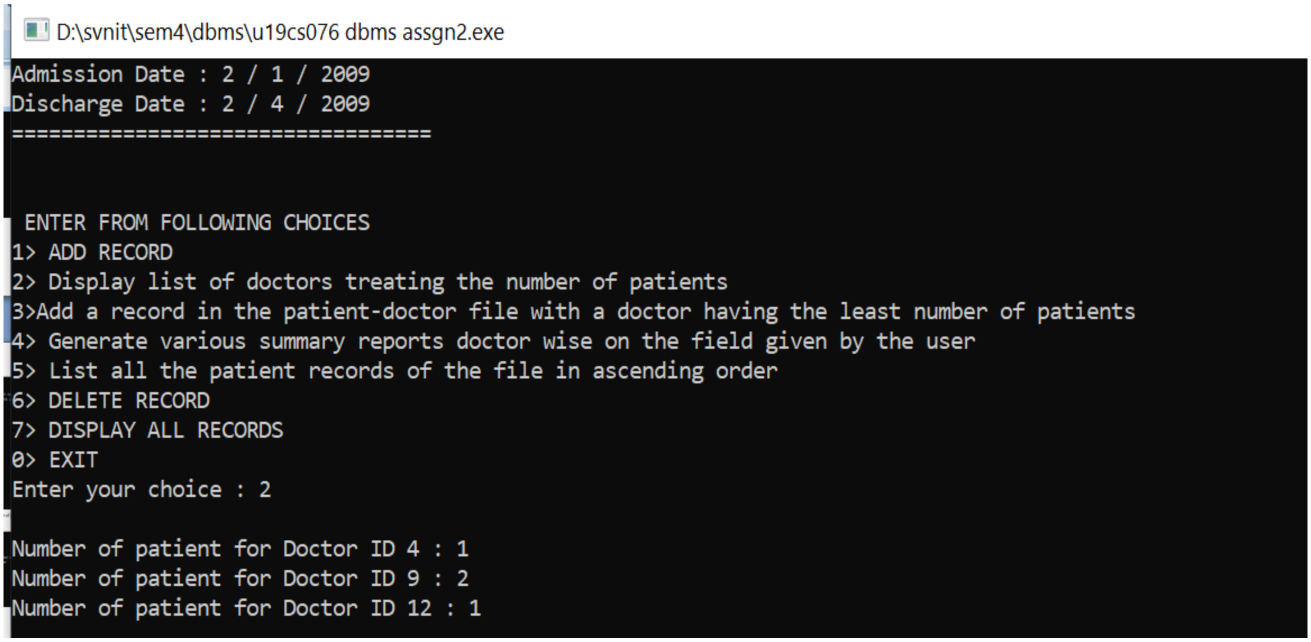
```
 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 1

1.Add record in doctor file
2.Add patient-doctor relation data in file
3.Add patient record data
Choose a menu option : 1

Enter Doctor ID: 9

Enter name name: Mark

 Enter gender (M/F): M

 Enter age :20

 Enter area of residency :east
-1
SUCCESSFULY PRINTED DATA IN FILE doctordatadata.txt
```

In Patient File



```
 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 1

1.Add record in doctor file
2.Add patient-doctor relation data in file
3.Add patient record data
Choose a menu option : 2

Enter Patient ID: 8

Enter Doctor ID: 9
Enter patient name :sebastien
Enter patient area :chennai
Enter admission date :2 9 2001
Enter discharge date :9 9 2010

SUCCESSFULY PRINTED DATA IN FILE patient_doc record data.txt
```

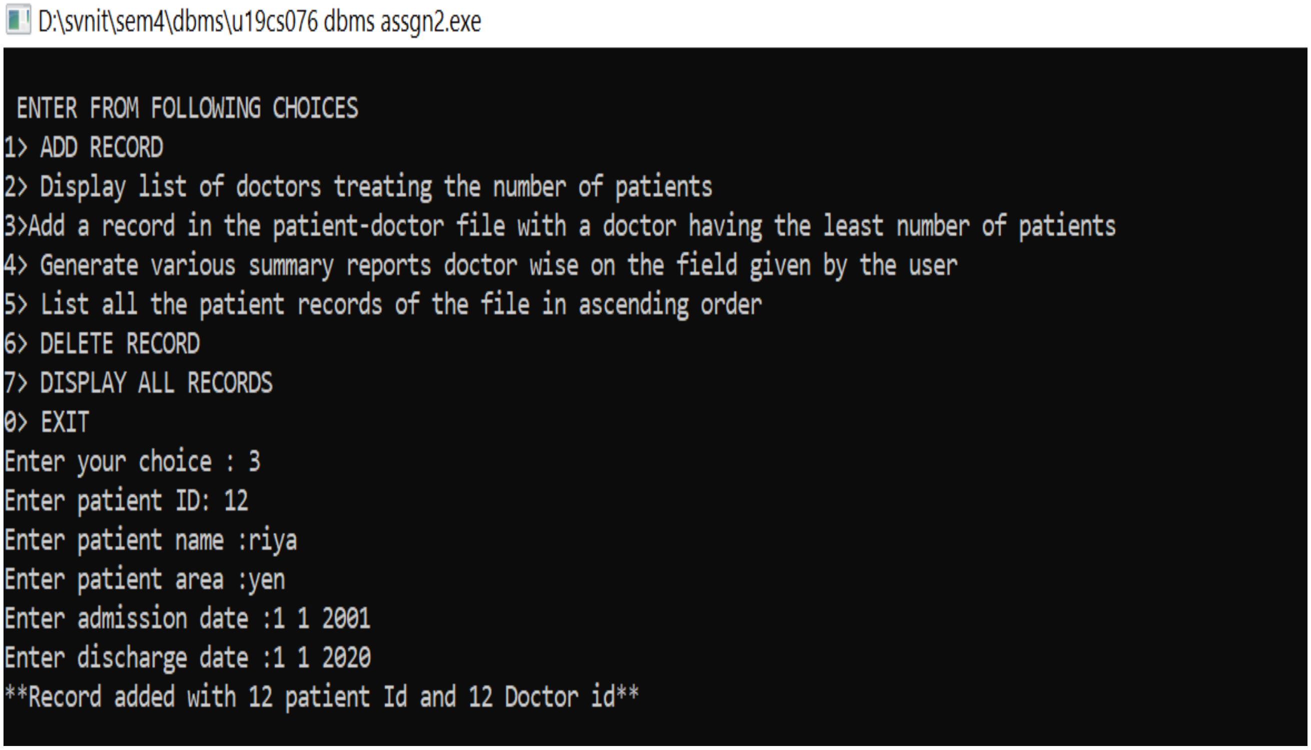
2. Display list of doctors treating the number of patients (Initial number of patient is 0)



```
Admission Date : 2 / 1 / 2009
Discharge Date : 2 / 4 / 2009
================================


 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 2

Number of patient for Doctor ID 4 : 1
Number of patient for Doctor ID 9 : 2
Number of patient for Doctor ID 12 : 1
```

3. Add a record in the patient-doctor file with a doctor having the least number of patients

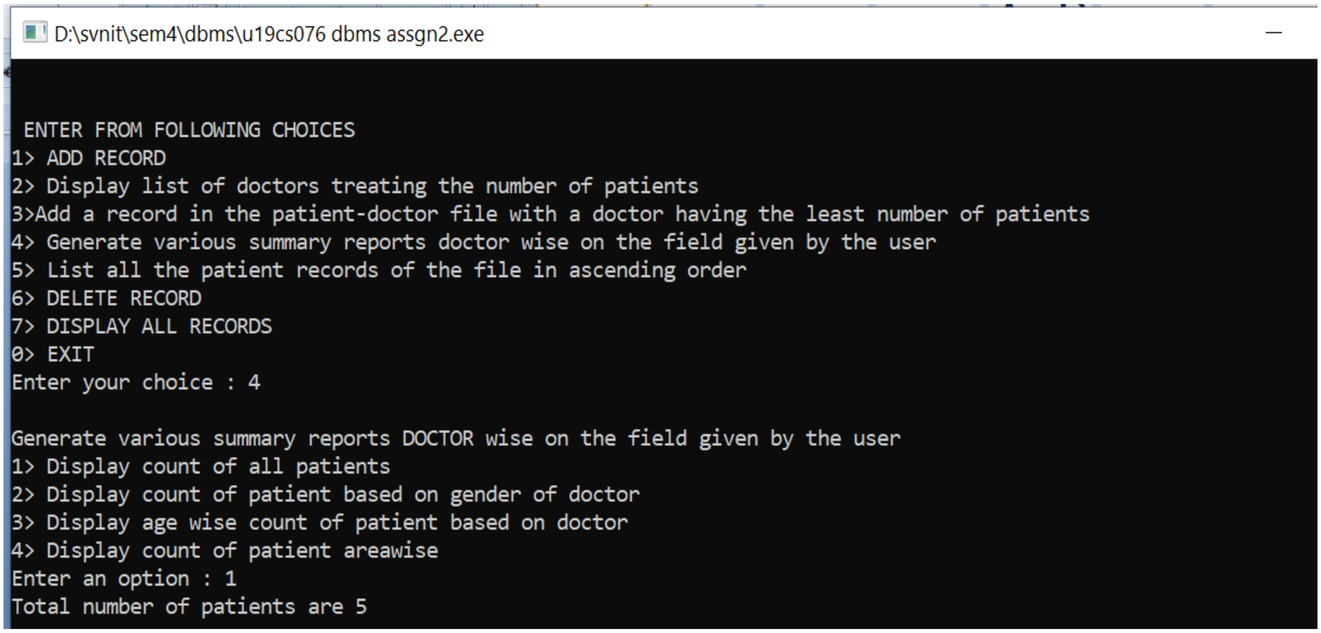
and on equal number of patient then consider the alphabetic order of doctor’s name



```
 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 3
Enter patient ID: 12
Enter patient name :riya
Enter patient area :yen
Enter admission date :1 1 2001
Enter discharge date :1 1 2020
**Record added with 12 patient Id and 12 Doctor id**
```

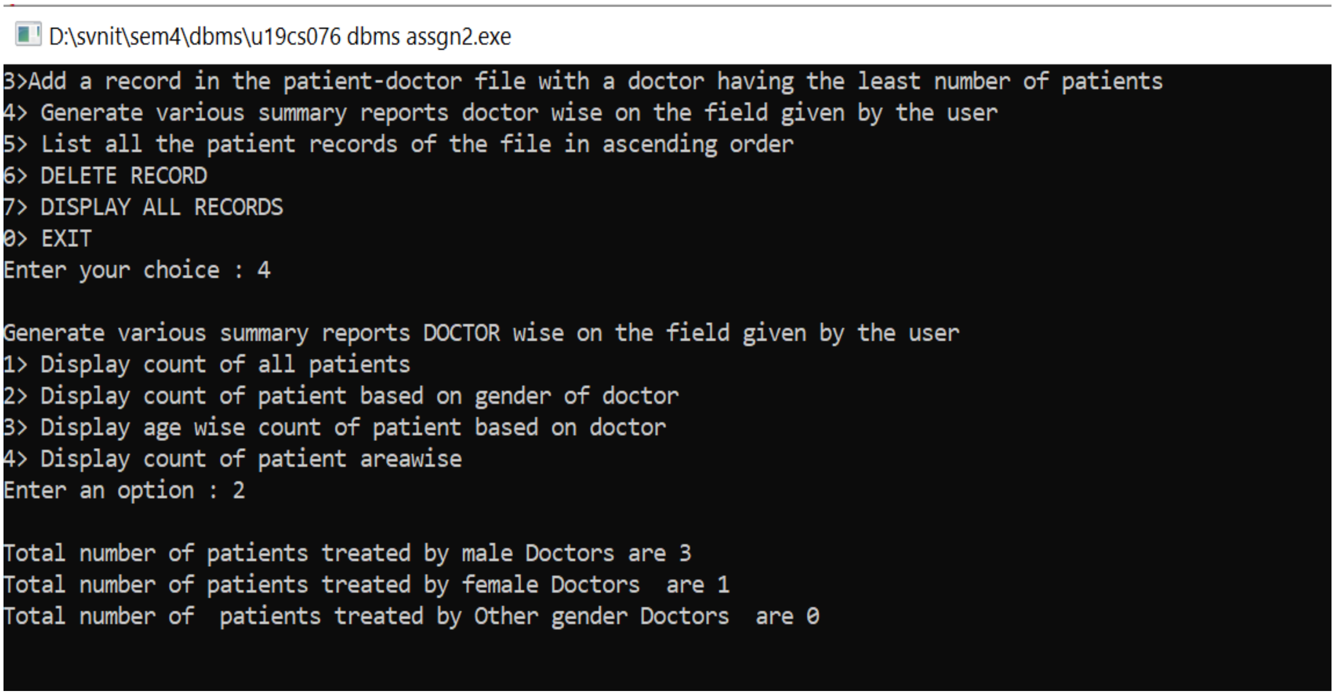
4. Generate various summary reports doctor wise on the field given by the user

i. Display total count of patient



```
 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 4

Generate various summary reports DOCTOR wise on the field given by the user
1> Display count of all patients
2> Display count of patient based on gender of doctor
3> Display age wise count of patient based on doctor
4> Display count of patient areawise
Enter an option : 1
Total number of patients are 5
```

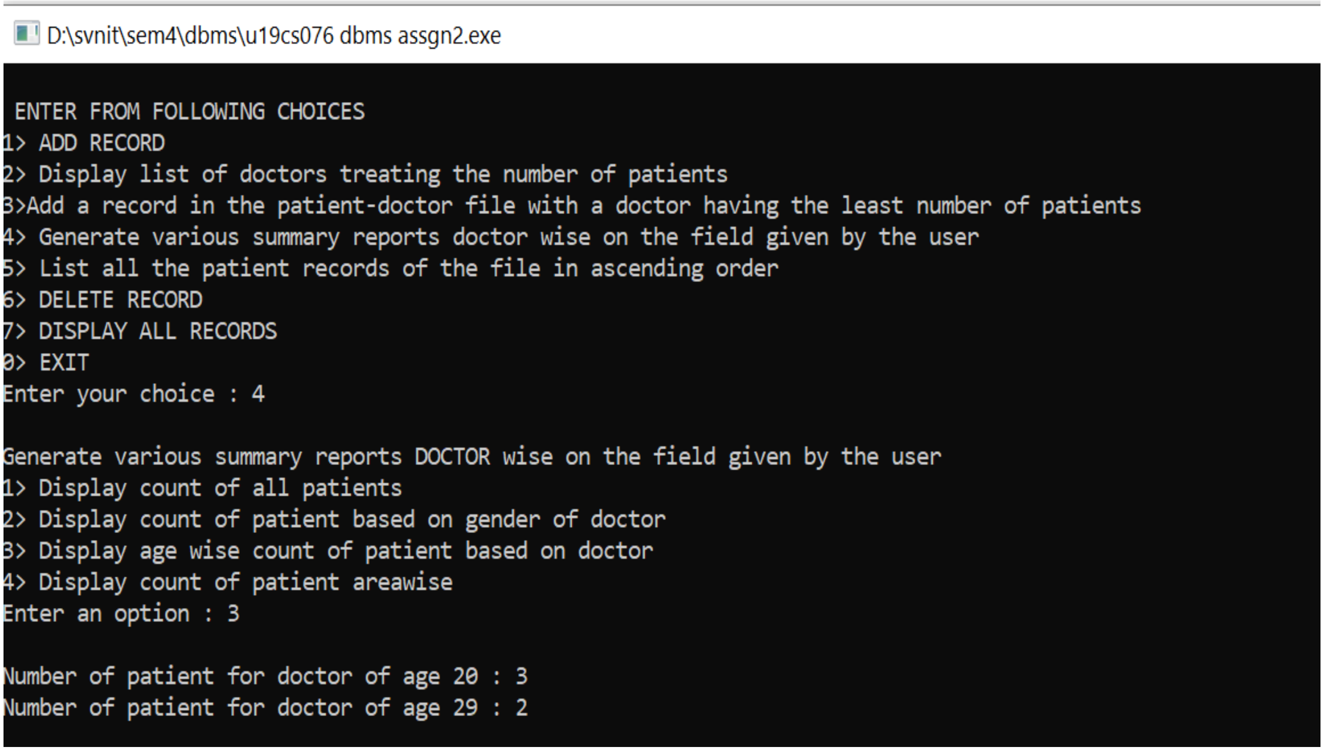
ii. Display the count of patient based on the gender(of doctor)



D:\svnit\sem4\dbms\u19cs076 dbms assgn2.exe

```
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 4

Generate various summary reports DOCTOR wise on the field given by the user
1> Display count of all patients
2> Display count of patient based on gender of doctor
3> Display age wise count of patient based on doctor
4> Display count of patient areawise
Enter an option : 2

Total number of patients treated by male Doctors are 3
Total number of patients treated by female Doctors  are 1
Total number of  patients treated by Other gender Doctors  are 0
```

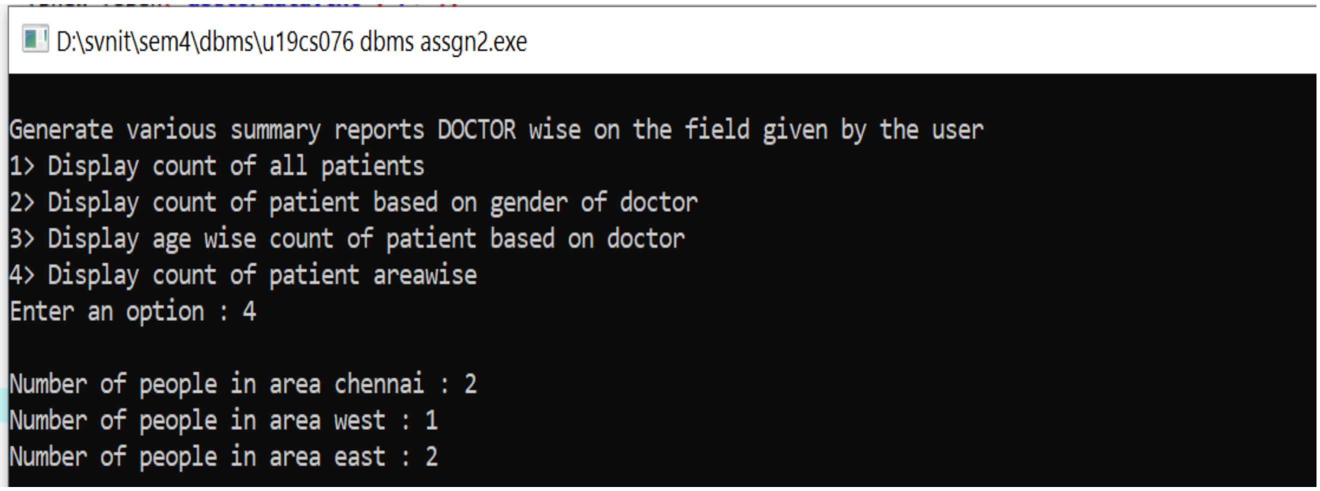
iii. Display the age wise count of patient



D:\svnit\sem4\dbms\u19cs076 dbms assgn2.exe

```
 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 4

Generate various summary reports DOCTOR wise on the field given by the user
1> Display count of all patients
2> Display count of patient based on gender of doctor
3> Display age wise count of patient based on doctor
4> Display count of patient areawise
Enter an option : 3

Number of patient for doctor of age 20 : 3
Number of patient for doctor of age 29 : 2
```

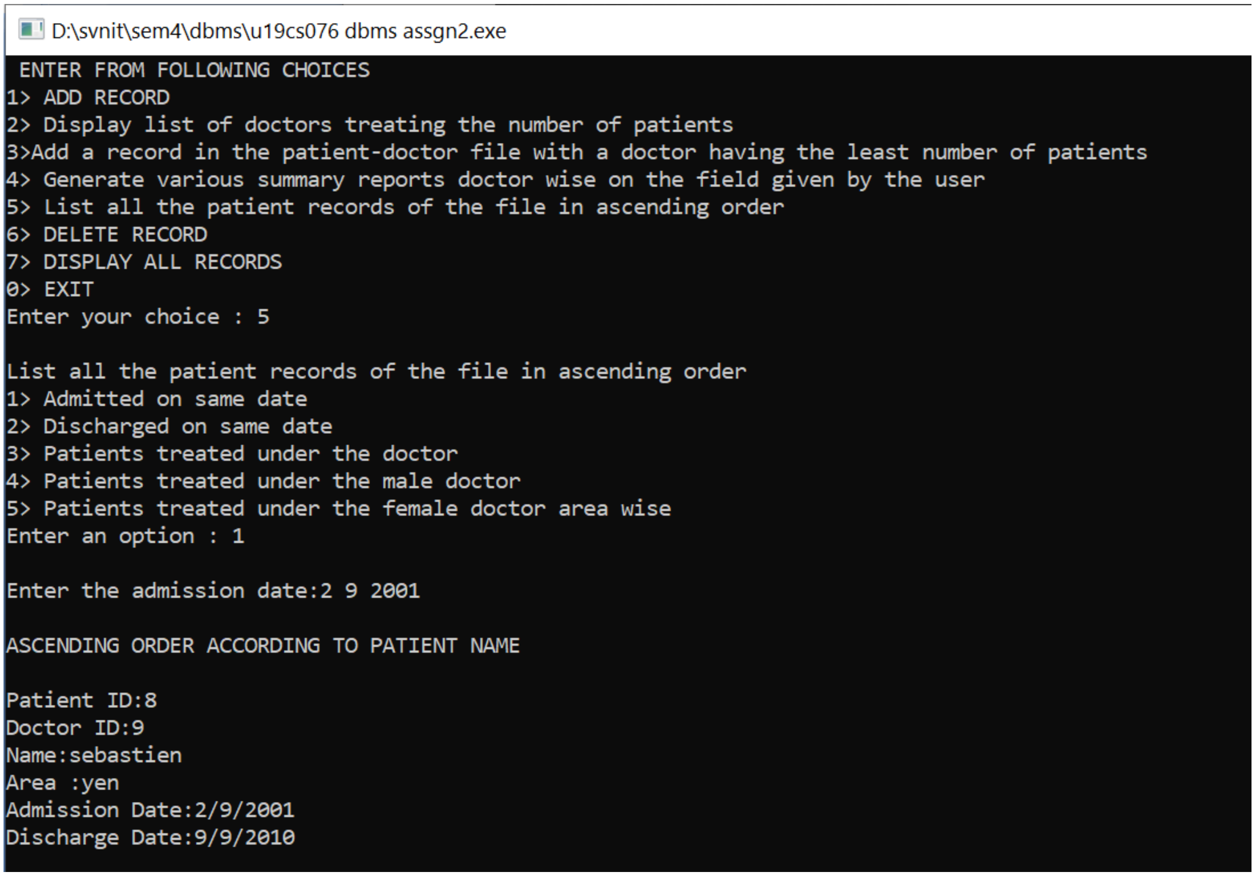
iv. Display the count of patient areawise



```
Generate various summary reports DOCTOR wise on the field given by the user
1> Display count of all patients
2> Display count of patient based on gender of doctor
3> Display age wise count of patient based on doctor
4> Display count of patient areawise
Enter an option : 4

Number of people in area chennai : 2
Number of people in area west : 1
Number of people in area east : 2
```

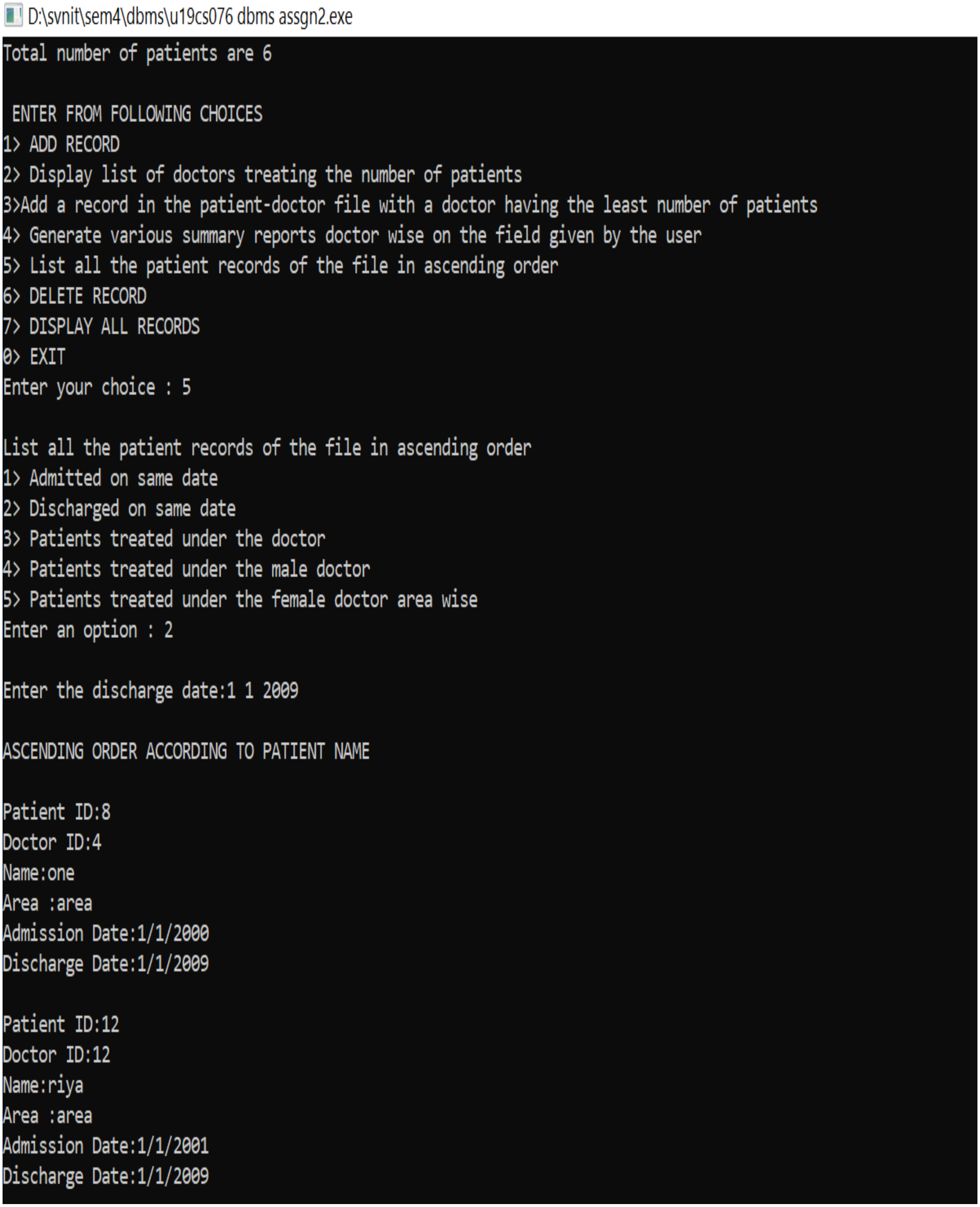
5. List all the patient records of the file in ascending order

i. Admitted on same date



```
 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 5

List all the patient records of the file in ascending order
1> Admitted on same date
2> Discharged on same date
3> Patients treated under the doctor
4> Patients treated under the male doctor
5> Patients treated under the female doctor area wise
Enter an option : 1

Enter the admission date:2 9 2001

ASCENDING ORDER ACCORDING TO PATIENT NAME

Patient ID:8
Doctor ID:9
Name:sebastien
Area :yen
Admission Date:2/9/2001
Discharge Date:9/9/2010
```

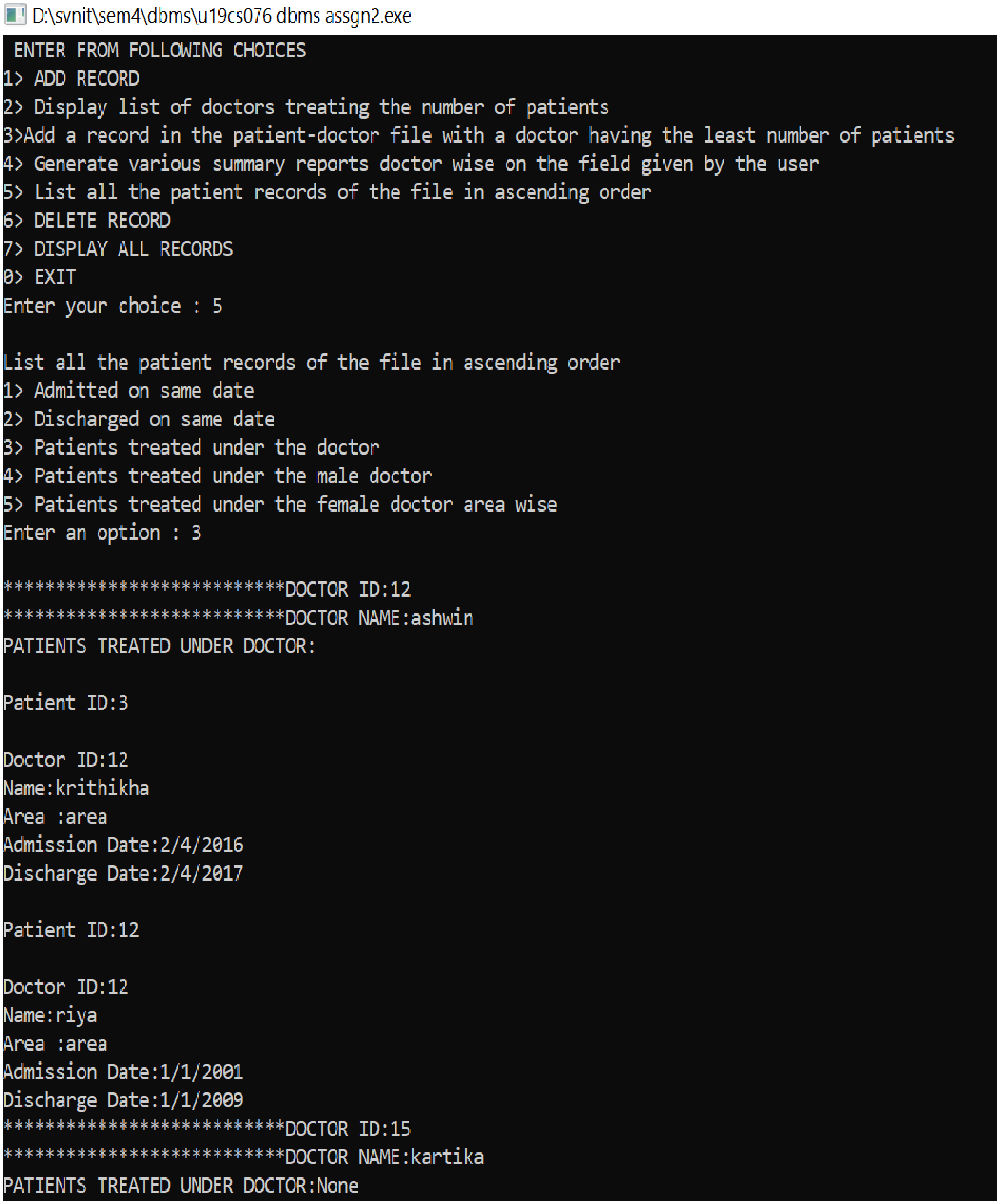
ii. Discharged on same date



D:\svnit\sem4\dbms\u19cs076 dbms assgn2.exe

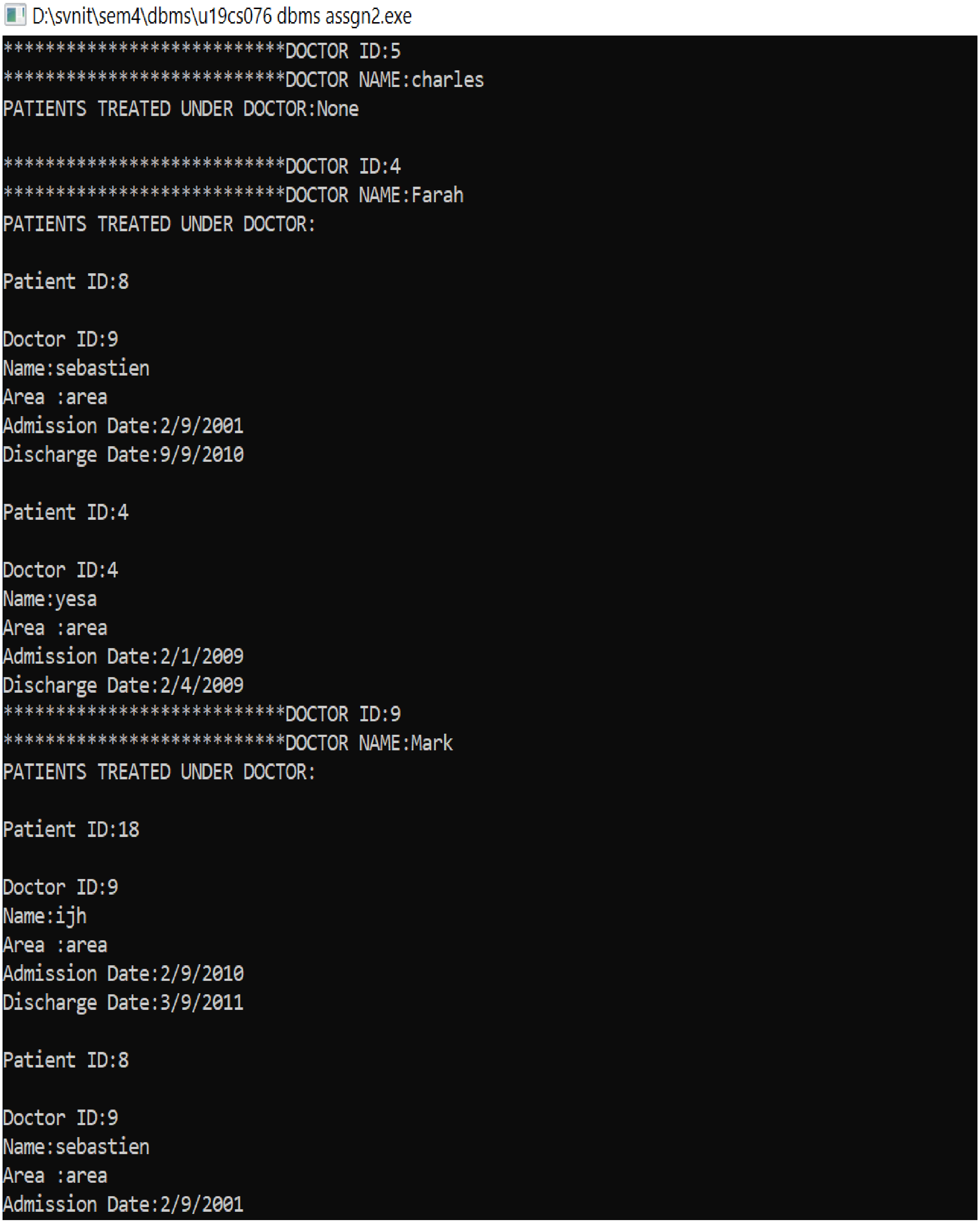
```
Total number of patients are 6

 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 5

List all the patient records of the file in ascending order
1> Admitted on same date
2> Discharged on same date
3> Patients treated under the doctor
4> Patients treated under the male doctor
5> Patients treated under the female doctor area wise
Enter an option : 2

Enter the discharge date:1 1 2009

ASCENDING ORDER ACCORDING TO PATIENT NAME

Patient ID:8
Doctor ID:4
Name:one
Area :area
Admission Date:1/1/2000
Discharge Date:1/1/2009

Patient ID:12
Doctor ID:12
Name:riya
Area :area
Admission Date:1/1/2001
Discharge Date:1/1/2009
```

iii. Patients treated under the doctor



D:\svnit\sem4\dbms\u19cs076 dbms assgn2.exe

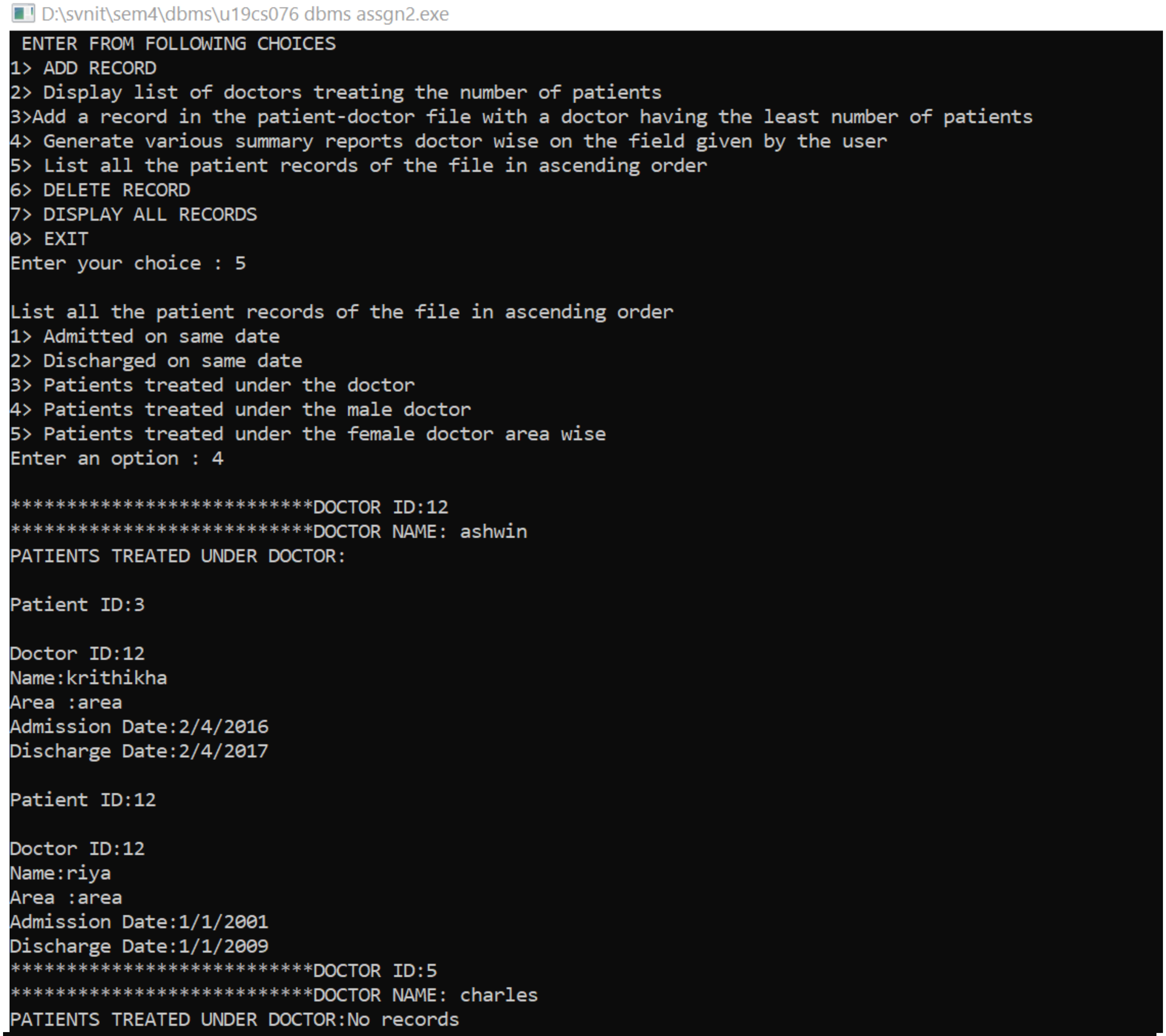
```
 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 5

List all the patient records of the file in ascending order
1> Admitted on same date
2> Discharged on same date
3> Patients treated under the doctor
4> Patients treated under the male doctor
5> Patients treated under the female doctor area wise
Enter an option : 3

***************************DOCTOR ID:12
***************************DOCTOR NAME:ashwin
PATIENTS TREATED UNDER DOCTOR:

Patient ID:3

Doctor ID:12
Name:krithikha
Area :area
Admission Date:2/4/2016
Discharge Date:2/4/2017

Patient ID:12

Doctor ID:12
Name:riya
Area :area
Admission Date:1/1/2001
Discharge Date:1/1/2009
***************************DOCTOR ID:15
***************************DOCTOR NAME:kartika
PATIENTS TREATED UNDER DOCTOR:None
```



D:\svnit\sem4\dbms\u19cs076 dbms assgn2.exe

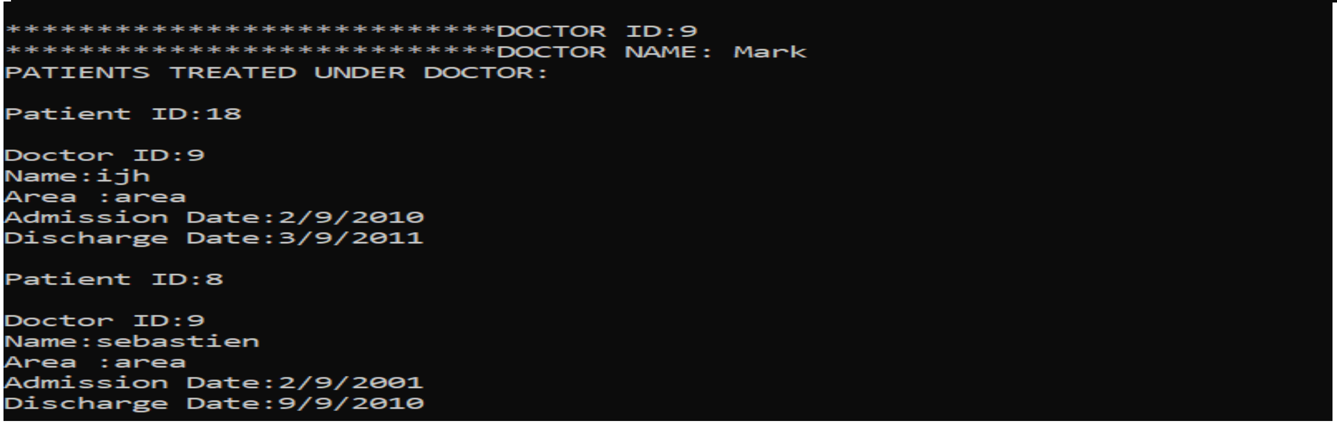
```
***************************DOCTOR ID:5
***************************DOCTOR NAME:charles
PATIENTS TREATED UNDER DOCTOR:None

***************************DOCTOR ID:4
***************************DOCTOR NAME:Farah
PATIENTS TREATED UNDER DOCTOR:

Patient ID:8

Doctor ID:9
Name:sebastien
Area :area
Admission Date:2/9/2001
Discharge Date:9/9/2010

Patient ID:4

Doctor ID:4
Name:yesa
Area :area
Admission Date:2/1/2009
Discharge Date:2/4/2009
***************************DOCTOR ID:9
***************************DOCTOR NAME:Mark
PATIENTS TREATED UNDER DOCTOR:

Patient ID:18

Doctor ID:9
Name:ijh
Area :area
Admission Date:2/9/2010
Discharge Date:3/9/2011

Patient ID:8

Doctor ID:9
Name:sebastien
Area :area
Admission Date:2/9/2001
```

iv. Patients treated under the male doctor



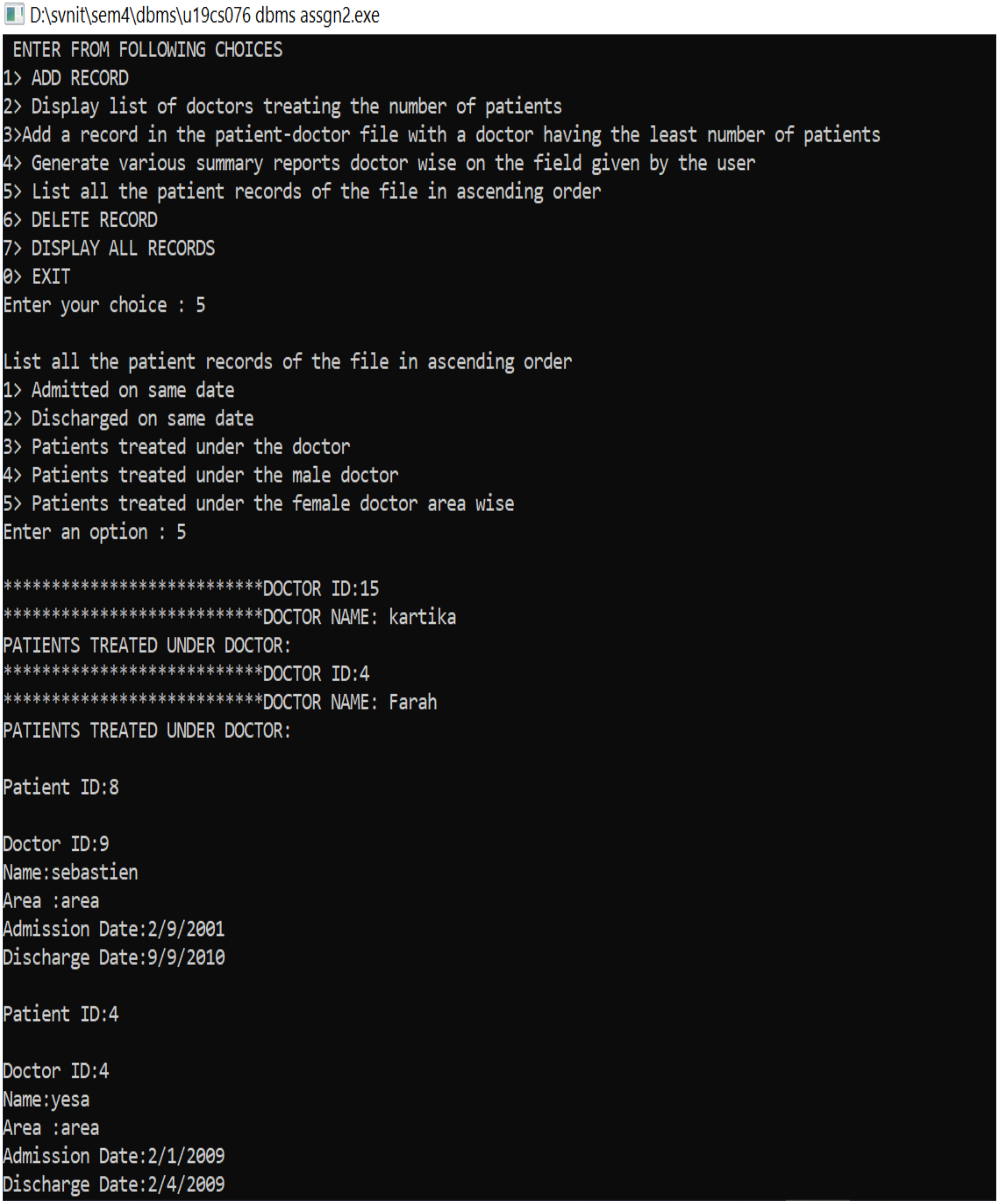
D:\svnit\sem4\dbms\u19cs076 dbms assgn2.exe

```
 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 5

List all the patient records of the file in ascending order
1> Admitted on same date
2> Discharged on same date
3> Patients treated under the doctor
4> Patients treated under the male doctor
5> Patients treated under the female doctor area wise
Enter an option : 4

***************************DOCTOR ID:12
***************************DOCTOR NAME: ashwin
PATIENTS TREATED UNDER DOCTOR:

Patient ID:3

Doctor ID:12
Name:krithikha
Area :area
Admission Date:2/4/2016
Discharge Date:2/4/2017

Patient ID:12

Doctor ID:12
Name:riya
Area :area
Admission Date:1/1/2001
Discharge Date:1/1/2009
***************************DOCTOR ID:5
***************************DOCTOR NAME: charles
PATIENTS TREATED UNDER DOCTOR:No records
```



```
***************************DOCTOR ID:9
***************************DOCTOR NAME: Mark
PATIENTS TREATED UNDER DOCTOR:

Patient ID:18

Doctor ID:9
Name:ijh
Area :area
Admission Date:2/9/2010
Discharge Date:3/9/2011

Patient ID:8

Doctor ID:9
Name:sebastien
Area :area
Admission Date:2/9/2001
Discharge Date:9/9/2010
```

v. Patients treated under the female doctor area wise

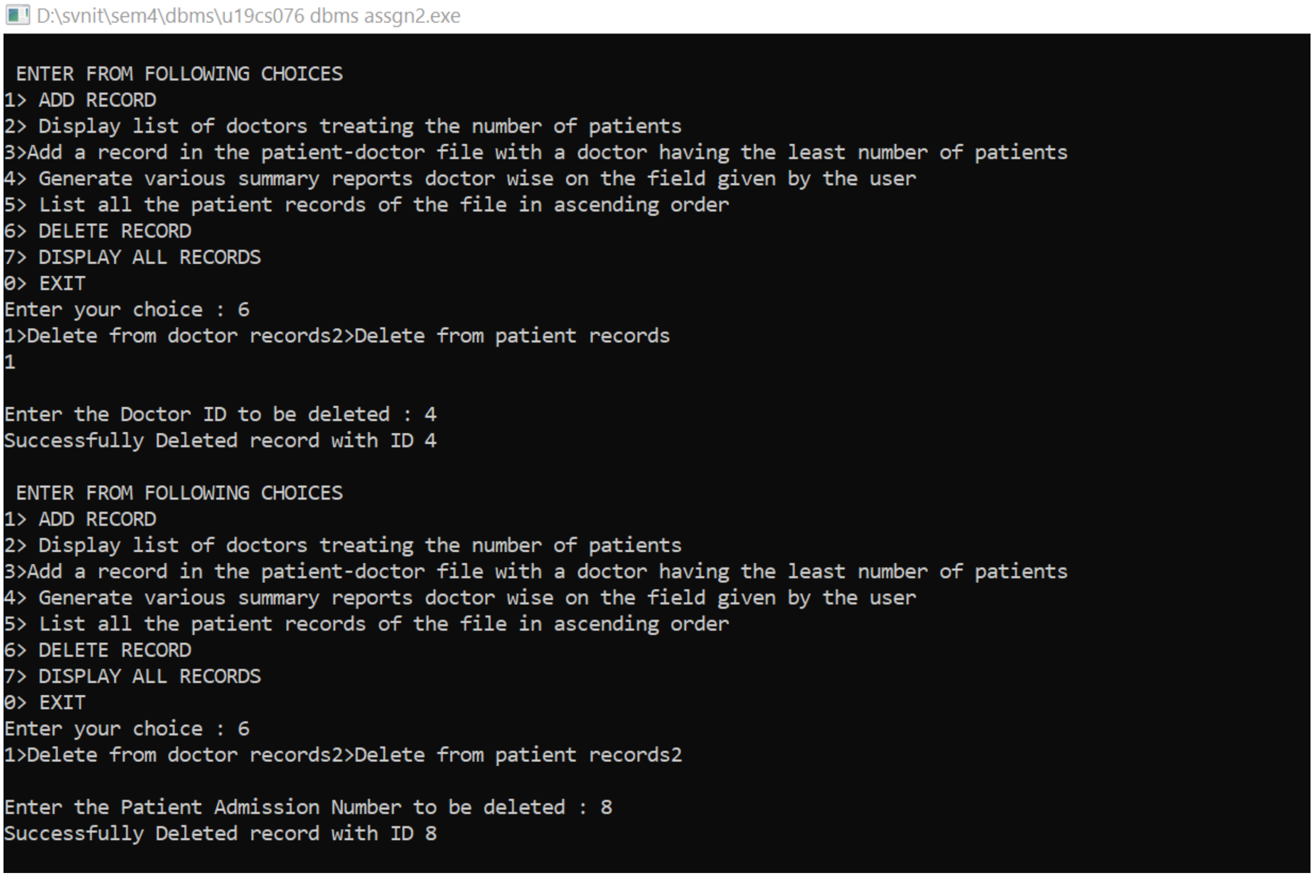


D:\svnit\sem4\dbms\u19cs076 dbms assgn2.exe

```
 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 5

List all the patient records of the file in ascending order
1> Admitted on same date
2> Discharged on same date
3> Patients treated under the doctor
4> Patients treated under the male doctor
5> Patients treated under the female doctor area wise
Enter an option : 5

***************************DOCTOR ID:15
***************************DOCTOR NAME: kartika
PATIENTS TREATED UNDER DOCTOR:
***************************DOCTOR ID:4
***************************DOCTOR NAME: Farah
PATIENTS TREATED UNDER DOCTOR:

Patient ID:8

Doctor ID:9
Name:sebastien
Area :area
Admission Date:2/9/2001
Discharge Date:9/9/2010

Patient ID:4

Doctor ID:4
Name:yesa
Area :area
Admission Date:2/1/2009
Discharge Date:2/4/2009
```

ADDITIONAL FUNCTIONS

DELETE



D:\svnit\sem4\dbms\u19cs076 dbms assgn2.exe

```
 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 6
1>Delete from doctor records2>Delete from patient records
1

Enter the Doctor ID to be deleted : 4
Successfully Deleted record with ID 4

 ENTER FROM FOLLOWING CHOICES
1> ADD RECORD
2> Display list of doctors treating the number of patients
3>Add a record in the patient-doctor file with a doctor having the least number of patients
4> Generate various summary reports doctor wise on the field given by the user
5> List all the patient records of the file in ascending order
6> DELETE RECORD
7> DISPLAY ALL RECORDS
0> EXIT
Enter your choice : 6
1>Delete from doctor records2>Delete from patient records2

Enter the Patient Admission Number to be deleted : 8
Successfully Deleted record with ID 8
```

7.Display all records

As given in page 37

THE END