PROJECT CODE

```
#include<bits/stdc++.h>
#include<iostream>
#include<fstream>
#include<stdio.h>
#include<conio.h>
#include<string>
#include<string.h>
#include<ctype.h>
#include<math.h>
#include<cmath>
#include<stdlib.h>
#include<process.h>
#include<vector>
using namespace std;

void WRITE();
void READ();
void SEARCH_ID();
void COUNT_REC();
void SEARCH();
void DELETION();
void MODIFY_ID();
void MODIFY();
void SORT();
void DISPLAY_FOR_CUSTOMER();
void SEARCH_NAME_FOR_CUSTOMER();
bool LOGIN();
void ADMIN();
void CUSTOMER();
void PURCHASE_FOR_CUSTOMER();

class Medicine
{
   int ID;
   int qty;
   char name[30];
   char cmpname[30];
   char supname[30];
```

```
    float unitCost;
    float saleCost;

public:
    Medicine()
    {
        ID=0;
        qty=0;
        name[0]='\0';
        cmpname[0]='\0';
        supname[0]='\0';
        unitCost=0;
        saleCost=0;
    }

   // Medicine(Medicine &M)
    //{
      //  ID=M.ID;
      //  qty=M.qty;
      //  strcpy(name,M.name);
      //  strcpy(cmpname,M.cmpname);
      //  strcpy(supname,M.supname);
      //  unitCost=M.unitCost;
      //  saleCost=M.saleCost;
    //}

    void indata();
    void outdata();
    void OutDataForCustomer();
    void generate_bill(int quantity);
    bool purchase(int quantity);

    //getter functions
    int retID();
    char*retName();
    char*retCmpname();
    char*retSupname();
};
```

```
void Medicine::indata()
{
    cout<<"\n Enter Medicine ID ";
    cin>>ID;
    cout<<"\n Enter Qty ";
    cin>>qty;
    cin.ignore(numeric_limits<streamsize>::max(),'\n');
    cout<<"\n Enter Medicine name ";
    gets(name);
    cout<<"\n Enter Company name ";
    gets(cmpname);
    cout<<"\n Enter Supplier name ";
    gets(supname);
    //cin.ignore(numeric_limits<streamsize>::max(),'\n');
    //cin.ignore(numeric_limits<streamsize>::max(),'\n');
    cout<<"\n Enter Unit Cost ";
    cin>>unitCost;
    cout<<"\n Enter Sale Cost ";
    cin>>saleCost;
}

void Medicine::outdata()
{
      cout<<endl<<"Medicine ID: "<<ID;
      cout<<endl<<"Name: "<<name;
      cout<<endl<<"Company name: "<<cmpname;
      cout<<endl<<"Supplier name: "<<supname;
      cout<<endl<<"Unit cost: "<<unitCost;
      cout<<endl<<"Sale Cost: "<<saleCost;
      cout<<endl<<"Quantity: "<<qty;
      cout<<endl;
}

void Medicine::OutDataForCustomer()
{
      cout<<"Name: "<<name;
      cout<<endl<<"  Company name: "<<cmpname;
      cout<<endl<<"  Sale Cost: "<<saleCost;
      cout<<endl;
}
```

```
int Medicine::retID()
{
   return(ID);
}

void Medicine::generate_bill(int quantity)
{
   cout<<"\n Your Bill ";
   cout<<"\n Name of the medicine: "<<name;
   cout<<"\n Quantity: "<<quantity;

   float cost;
   cost=quantity*saleCost;
   cout<<"\n Cost: "<<cost;
}

bool Medicine::purchase(int quantity)
{
   if(quantity>qty)
   {
      return false;
   }

   qty=qty-quantity;
   return true;
}

char*Medicine::retCmpname()
{
   return(cmpname);
}

char*Medicine::retName()
{
   return(name);
}

char*Medicine::retSupname()
{
   return(supname);
}

struct CompareId1 {
        bool operator()(Medicine p1, Medicine p2)
        {
                return p1.retID() > p2.retID();
        }
};

struct CompareId {
        bool operator()(Medicine p1, Medicine p2)
        {
                return p1.retID() < p2.retID();
        }
};

struct CompareName {
        bool operator()(Medicine p1, Medicine p2)
        {
                return p1.retName() < p2.retName();
        }
};

void WRITE()
{
   ofstream f("med1.dat",ios::binary | ios::app);
   Medicine m;
   char reply;
   do
   {
      m.indata();
      f.write((char*)&m,sizeof(m));
      cout<<"Want to enter more records";
      cin>>reply;
   }while(toupper(reply)=='Y');
   f.close();
}

void READ()
```

```
{
   ifstream f("med1.dat",ios::binary | ios::in);
   Medicine m;

   if(!f)
   {
      cout<<endl<<"FILE DOESN'T EXIST!!!!!";
      return;
   }
   int ctr=0;

   while(f.read((char*)&m,sizeof(m)))
   {
      cout<<endl<<"Record: "<<++ctr;
      m.outdata();
   }
   f.close();
}

void SEARCH_ID()
{
   ifstream f("med1.dat",ios::in | ios::binary);
   char found='N';
    Medicine m;
   if(!f)
   {
      cout<<endl<<"FILE DOESN'T EXIST!!!!!";
      return;
   }
   int id1;
   cout<<"Enter ID whose record is to be searched: ";
   cin>>id1;
   while(f.read((char*)&m,sizeof(m)))
   {
      if(m.retID()==id1)
      {
         cout<<"INFORMATION OF THE MEDICINE :";
         m.outdata();
         found='Y';
         break;
      }
   }
   if(found=='N')
   {
      cout<<"NO SUCH RECORD FOUND"<<endl;
   }
   f.close();
}

void SEARCH_NAME()
{
   ifstream f("med1.dat",ios::in | ios::binary);
   char found='N';
    Medicine m;
   if(!f)
   {
      cout<<endl<<"FILE DOESN'T EXIST!!!!!";
      return;
   }
   char nm[30];
    cin.ignore(numeric_limits<streamsize>::max(),'\n');
   cout<<"Enter name whose record is to be searched: ";
   gets(nm);
   while(f.read((char*)&m,sizeof(m)))
   {
      if(strcmp(m.retName(),nm)==0)
      {
         cout<<"INFORMATION OF THE MEDICINE :";
         m.outdata();
         found='Y';
         break;
      }
   }
   if(found=='N')
   {
      cout<<"NO SUCH RECORD FOUND"<<endl;
   }
   f.close();
}
```

```
void SEARCH_CMPNAME()
{
   ifstream f("med1.dat",ios::in | ios::binary);
   char found='N';
    Medicine m;
   if(!f)
   {
      cout<<endl<<"FILE DOESN'T EXIST!!!!!";
      return;
   }
   char company[30];
    cin.ignore(numeric_limits<streamsize>::max(),'\n');
   cout<<"Enter company name whose record is to be searched: ";
   gets(company);
   while(f.read((char*)&m,sizeof(m)))
   {
      if(strcmp(m.retCmpname(),company)==0)
      {
         cout<<"INFORMATION OF THE MEDICINE :";
         m.outdata();
         found='Y';
         break;
      }
   }
   if(found=='N')
   {
      cout<<"NO SUCH RECORD FOUND"<<endl;
   }
   f.close();
}

void SEARCH_SUPNAME()
{
   ifstream f("med1.dat",ios::in | ios::binary);
   char found='N';
    Medicine m;
   if(!f)
   {
      cout<<endl<<"FILE DOESN'T EXIST!!!!!";
      return;
   }
   char supply[30];
    cin.ignore(numeric_limits<streamsize>::max(),'\n');
   cout<<"Enter company name whose record is to be searched: ";
   gets(supply);
   while(f.read((char*)&m,sizeof(m)))
   {
      if(strcmp(m.retSupname(),supply)==0)
      {
         cout<<"INFORMATION OF THE MEDICINE :";
         m.outdata();
         found='Y';
         break;
      }
   }
   if(found=='N')
   {
      cout<<"NO SUCH RECORD FOUND"<<endl;
   }
   f.close();
}

void COUNT_REC()
{
   Medicine m;
   ifstream fs("med1.dat",ios::in|ios::binary);
    if(!fs)
   {
      cout<<"File Reading Error"<<endl;
      return;
   }
   int num=0;
   while(fs.read((char*)&m,sizeof(m)))
   {
      num++;
   }
   cout<<"Records in the file are-- "<<num;
   fs.close();
}
```

```
void SEARCH()
{
   int s;
   cout<<"\n Search on the basis of :- ";
   cout<<"\n 1.ID";
   cout<<"\n 2.Name";
   cout<<"\n 3.Company Name";
   cout<<"\n 4.Supplier Name";
   cin>>s;
   switch(s)
   {
   case 1:
      SEARCH_ID();
      break;
   case 2:
      SEARCH_NAME();
      break;
   case 3:
      SEARCH_CMPNAME();
      break;
   case 4:
      SEARCH_SUPNAME();
      break;
   }
}

void MODIFY_ID()
{
   fstream f("med1.dat", ios::in | ios::out | ios::binary);
   Medicine m;
   int id1;
   cout<<"\n Enter ID whose record is to be modified: ";
   cin>>id1;
   int rec_count=0;
   char found='N';
   while(f.read((char*)&m,sizeof(m)))
   {
      if(m.retID()==id1)
      {
         cout<<"\n Enter New Information";
```

```
            m.indata();
            f.seekg(rec_count*sizeof(Medicine),ios::beg);
            f.write((char*)&m,sizeof(Medicine));
            found='Y';
            break;
        }
        rec_count++;
    }
    if(found=='Y')
    {
        cout<<"\n Medicine Details Updated";
    }
    else
    {
        cout<<"\n Record Not Found!!!!";
    }
    f.close();
}

void MODIFY_NAME()
{
    fstream f("med1.dat", ios::in | ios::out | ios::binary);
    Medicine m;

    cin.ignore(numeric_limits<streamsize>::max(),'\n');
    char nm[30];
    cout<<"\n Enter name whose record is to be modified: ";
    gets(nm);

    int rec_count=0;
    char found='N';
    while(f.read((char*)&m,sizeof(m)))
    {
        if(strcmp(nm,m.retName())==0)
        {
            cout<<"\n Enter New Information";
            m.indata();
            f.seekg(rec_count*sizeof(Medicine),ios::beg);
            f.write((char*)&m,sizeof(Medicine));
            found='Y';
            break;
        }
        rec_count++;
    }
    if(found=='Y')
    {
        cout<<"\n Medicine Details Updated";
    }
    else
    {
        cout<<"\n Record Not Found!!!!";
    }
    f.close();
}

void MODIFY()
{
    int s;
    cout<<"\n Modify on the basis of :- ";
    cout<<"\n 1.ID";
    cout<<"\n 2.Name";
    cin>>s;

    switch(s)
    {
    case 1:
        MODIFY_ID();
        break;

    case 2:
        MODIFY_NAME();
        break;
    }
}

void DELETION1()
{
    ifstream fmain("med1.dat", ios::binary | ios::in);
    ofstream ftemp("temp.dat", ios::binary | ios::out);
    Medicine m;
    int id1;
    cout<<"Enter ID whose record is to be deleted:";
    cin>>id1;
    char found='N';
    while(fmain.read((char*)&m,sizeof(m)))
    {
        if(m.retID()!=id1)
        {
            ftemp.write((char*)&m,sizeof(m));
        }
        else
        {
            found='Y';
        }
    }
    if(found=='N')
    {
        cout<<"\n RECORD NOT FOUND!!";
    }
    else
    {
        cout<<"\n RECORD DELETED";
    }
    fmain.close();
    ftemp.close();
    remove("med1.dat");
    rename("temp.dat","med1.dat");
}

void DELETION2()
{
    ifstream fmain("med1.dat", ios::binary | ios::in);
    ofstream ftemp("temp.dat", ios::binary | ios::out);
    Medicine m;
   // int id1;
    //cout<<"Enter ID whose record is to be deleted:";
    //cin>>id1;
    char nm[30];
    cin.ignore(numeric_limits<streamsize>::max(),'\n');
    cout<<"Enter Name whose record is to be deleted: ";
   gets(nm);
   char found='N';
   while(fmain.read((char*)&m,sizeof(m)))
   {
      if(strcmp(m.retName(),nm)!=0)
      {
         ftemp.write((char*)&m,sizeof(m));
      }
      else
      {
         found='Y';
      }
   }
   if(found=='N')
   {
      cout<<"\n RECORD NOT FOUND!!";
   }
   else
   {
      cout<<"\n RECORD DELETED";
   }
   fmain.close();
   ftemp.close();
   remove("med1.dat");
   rename("temp.dat","med1.dat");
}

void DELETION3()
{
   ifstream fmain("med1.dat", ios::binary | ios::in);
   ofstream ftemp("temp.dat", ios::binary | ios::out);
   Medicine m;
  // int id1;
   //cout<<"Enter ID whose record is to be deleted:";
   //cin>>id1;
   char nm[30];
   cin.ignore(numeric_limits<streamsize>::max(),'\n');
   cout<<"Enter Name whose record is to be deleted: ";
   gets(nm);
   char found='N';
   while(fmain.read((char*)&m,sizeof(m)))
   {
      if(strcmp(m.retCmpname(),nm)!=0)
      {
         ftemp.write((char*)&m,sizeof(m));
      }
      else
      {
         found='Y';
      }
   }
   if(found=='N')
   {
      cout<<"\n RECORD NOT FOUND!!";
   }
   else
   {
      cout<<"\n RECORD DELETED";
   }
   fmain.close();
   ftemp.close();
   remove("med1.dat");
   rename("temp.dat","med1.dat");
}

void DELETION4()
{
   ifstream fmain("med1.dat", ios::binary | ios::in);
   ofstream ftemp("temp.dat", ios::binary | ios::out);
   Medicine m;
  // int id1;
   //cout<<"Enter ID whose record is to be deleted:";
   //cin>>id1;
   char nm[30];
   cin.ignore(numeric_limits<streamsize>::max(),'\n');
   cout<<"Enter Name whose record is to be deleted: ";
   gets(nm);
   char found='N';
   while(fmain.read((char*)&m,sizeof(m)))
   {
        if(strcmp(m.retSupname(),nm)!=0)
        {
            ftemp.write((char*)&m,sizeof(m));
        }
        else
        {
            found='Y';
        }
    }
    if(found=='N')
    {
        cout<<"\n RECORD NOT FOUND!!";
    }
    else
    {
        cout<<"\n RECORD DELETED";
    }
    fmain.close();
    ftemp.close();
    remove("med1.dat");
    rename("temp.dat","med1.dat");
}

void DELETION()
{
    int s;
    cout<<"\n Delete on the basis of :- ";
    cout<<"\n 1.ID";
    cout<<"\n 2.Name";
    cout<<"\n 3.Company Name";
    cout<<"\n 4.Supplier Name";
    cin>>s;

    switch(s)
    {
    case 1:
        {
            DELETION1();
            break;
        }
    case 2:
        {
            DELETION2();
            break;
        }
    case 3:
        {
            DELETION3();
            break;
        }
    case 4:
        {
            DELETION4();
            break;
        }
    }
}

void merge1(Medicine arr[], int l, int m, int r)
{
    int n1 = m - l + 1;
    int n2 = r - m;

    Medicine L[n1], R[n2];

    for (int i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (int j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

    int i = 0;
    int j = 0;
    int k = l;

    while (i < n1 && j < n2) {
        if (L[i].retID() <= R[j].retID()) {
            arr[k] = L[i];
            i++;
        }
        else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }

    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}

void mergeSort(Medicine arr[], int l, int r)
{
    if (l < r) {

        int m = (l + r - l) / 2;
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);

        merge1(arr, l, m, r);
    }
}

void SORT_BY_ID()
{
    fstream f;
    f.open("med1.dat",ios::binary | ios::in);
    Medicine M[1000];

    if(!f)
    {
        cout<<"FILE DOESN'T EXISTS!!!!! ";
```

```
   return;
}

int n=0;
while(f.read((char*)&M[n],sizeof(Medicine)))
 {
    n++;
 }

Medicine temp;
int choice;
cout<<"Sort"<<endl;
cout<<"1. Ascending"<<endl;
cout<<"2. Descending"<<endl;
cin>>choice;

if(choice==1)
{
    priority_queue<Medicine,vector<Medicine>,CompareId1>Q;
   for(int i=0;i<n;i++)
   {
      Q.push(M[i]);
   }
   for(int i=0;i<n;i++)
   {
      M[i]=Q.top();
      Q.pop();
   }
}
else
{
   priority_queue<Medicine,vector<Medicine>,CompareId>Q;
   for(int i=0;i<n;i++)
   {
      Q.push(M[i]);
   }
   for(int i=0;i<n;i++)
   {
      M[i]=Q.top();
      Q.pop();
```

```
        }
    }

    f.close();
    f.open("med1.dat",ios::binary | ios::out);
    int i=0;
    while(i<n)
    {
        f.write((char*)&M[i],sizeof(Medicine));
        ++i;
    }
    cout<<"\n FILE SORTED";
    f.close();
}

void SORT_BY_NAME()
{
    fstream f;
    f.open("med1.dat",ios::binary | ios::in);
    Medicine M[1000];

    if(!f)
    {
        cout<<"FILE DOESN'T EXISTS!!!!! ";
        return;
    }

    int n=0;
    while(f.read((char*)&M[n],sizeof(Medicine)))
     {
         n++;
     }

    Medicine temp;
    int choice;
    cout<<"Sort"<<endl;
    cout<<"1. Ascending"<<endl;
    cout<<"2. Descending"<<endl;
    cin>>choice;
    if(choice==1)
    {
        for(int i=1;i<n;i++)
        {
            for(int j=0;j<n-i;j++)
            {
                if(strcmp(M[j].retName(),M[j+1].retName())>0)
                {
                    temp=M[j];
                    M[j]=M[j+1];
                    M[j+1]=temp;
                }
            }
        }
    }
    else
    {
        priority_queue<Medicine,vector<Medicine>,CompareName>Q;
        for(int i=0;i<n;i++)
        {
            Q.push(M[i]);
        }
        for(int i=0;i<n;i++)
        {
            M[i]=Q.top();
            Q.pop();
        }
    }

    f.close();
    f.open("med1.dat",ios::binary | ios::out);
    int i=0;
    while(i<n)
    {
        f.write((char*)&M[i],sizeof(Medicine));
        ++i;
    }
    cout<<"\n FILE SORTED";
    f.close();
}
```

```
void SORT_BY_CMPNAME()
{
   fstream f;
   f.open("med1.dat",ios::binary | ios::in);
   Medicine M[1000];

   if(!f)
   {
      cout<<"FILE DOESN'T EXISTS!!!!! ";
      return;
   }

   int n=0;
   while(f.read((char*)&M[n],sizeof(Medicine)))
    {
       n++;
    }

   Medicine temp;
   int choice;
   cout<<"Sort"<<endl;
   cout<<"1. Ascending"<<endl;
   cout<<"2. Descending"<<endl;
   cin>>choice;

   if(choice==1)
   {
      for(int i=1;i<n;i++)
      {
         for(int j=0;j<n-i;j++)
         {
            if(strcmp(M[j].retCmpname(),M[j+1].retCmpname())>0)
            {
               temp=M[j];
               M[j]=M[j+1];
               M[j+1]=temp;
            }
         }
      }
    }

    else
    {
        for(int i=1;i<n;i++)
        {
            for(int j=0;j<n-i;j++)
            {
                if(strcmp(M[j].retCmpname(),M[j+1].retCmpname())<0)
                {
                    temp=M[j];
                    M[j]=M[j+1];
                    M[j+1]=temp;
                }
            }
        }
    }

    f.close();

    f.open("med1.dat",ios::binary | ios::out);
    int i=0;
    while(i<n)
    {
        f.write((char*)&M[i],sizeof(Medicine));
        ++i;
    }
    cout<<"\n FILE SORTED";

    f.close();
}

void SORT_BY_SMPNAME()
{
    fstream f;
    f.open("med1.dat",ios::binary | ios::in);
    Medicine M[1000];

    if(!f)
    {
```

```
   cout<<"FILE DOESN'T EXISTS!!!!! ";
   return;
}

int n=0;
while(f.read((char*)&M[n],sizeof(Medicine)))
 {
    n++;
 }

Medicine temp;
int choice;
cout<<"Sort"<<endl;
cout<<"1. Ascending"<<endl;
cout<<"2. Descending"<<endl;
cin>>choice;

if(choice==1)
{
   for(int i=1;i<n;i++)
   {
      for(int j=0;j<n-i;j++)
      {
         if(strcmp(M[j].retSupname(),M[j+1].retSupname())>0)
         {
            temp=M[j];
            M[j]=M[j+1];
            M[j+1]=temp;
         }
      }
   }
}

else
{
   for(int i=1;i<n;i++)
   {
      for(int j=0;j<n-i;j++)
      {
         if(strcmp(M[j].retSupname(),M[j+1].retSupname())<0)
```

```
                {
                    temp=M[j];
                    M[j]=M[j+1];
                    M[j+1]=temp;
                }
            }
        }
    }

    f.close();
    f.open("med1.dat",ios::binary | ios::out);

    int i=0;
    while(i<n)
    {
        f.write((char*)&M[i],sizeof(Medicine));
        ++i;
    }

    cout<<"\n FILE SORTED";
    f.close();
}

void SORT()
{
    int s;
    cout<<"\n Sort on the basis of :- ";
    cout<<"\n 1.ID";
    cout<<"\n 2.Name";
    cout<<"\n 3.Company Name";
    cout<<"\n 4.Supplier Name";
    cin>>s;

    switch(s)
    {
    case 1:
        {
            SORT_BY_ID();
            break;
        }
    case 2:
        {
            SORT_BY_NAME();
            break;
        }
    case 3:
        {
            SORT_BY_CMPNAME();
            break;
        }
    case 4:
        {
            SORT_BY_SMPNAME();
            break;
        }
    }
}

void ADMIN()
{
    char ch1;
    int ch;

    do{
        cout<<"\n WELCOME TO MEDICAL STORE";
        cout<<"\n 1.Create a file";
        cout<<"\n 2.Read a file";
        cout<<"\n 3.Count total records in a file";
        cout<<"\n 4.Search some Record";
        cout<<"\n 5.Modify a record";
        cout<<"\n 6.Delete a record ";
        cout<<"\n 7.Sort all records";
        cout<<"\n 8.Exit";
        cin>>ch;

        switch(ch)
        {
        case 1:
            WRITE();
            break;
        case 2:
            READ();
            break;
        case 3:
            COUNT_REC();
            break;
        case 4:
            SEARCH();
            break;
        case 5:
            MODIFY();
            break;
        case 6:
            DELETION();
            break;
        case 7:
            SORT();
            break;
        case 8:
            break;

        }
        cout<<endl<<"\n Want to continue as ADMIN(y/n)--";
        cin>>ch1;

    }while(ch1=='y' || ch1=='Y');

}

bool LOGIN()
{
    char username[30];
    char password[30];
    cin.ignore(numeric_limits<streamsize>::max(),'\n');

    cout<<"\n Enter Username  ";
    gets(username);
    cout<<"\n Enter Password  ";
    gets(password);
    if((strcmp(username,"admin")==0) && (strcmp(password,"Hello")==0))
    {
        return true;
    }

    return false;
}

void DISPLAY_FOR_CUSTOMER()
{
    ifstream f("med1.dat",ios::binary | ios::in);
    Medicine m;

    if(!f)
    {
        cout<<endl<<"FILE DOESN'T EXIST!!!!!";
        return;
    }
    int ctr=0;

    while(f.read((char*)&m,sizeof(m)))
    {
        cout<<endl<<++ctr<<")";
        m.OutDataForCustomer();
    }
    f.close();
}

void SEARCH_NAME_FOR_CUSTOMER()
{
    ifstream f("med1.dat",ios::in | ios::binary);
    char found='N';
     Medicine m;
    if(!f)
    {
        cout<<endl<<"FILE DOESN'T EXIST!!!!!";
        return;
    }
    char nm[30];
     cin.ignore(numeric_limits<streamsize>::max(),'\n');
    cout<<"Enter name whose record is to be searched: ";
    gets(nm);
    while(f.read((char*)&m,sizeof(m)))
    {
        if(strcmp(m.retName(),nm)==0)
        {
            cout<<"INFORMATION OF THE MEDICINE : \n";
            cout<<"  ";
            m.OutDataForCustomer();
            found='Y';
            break;
        }
    }
    if(found=='N')
    {
        cout<<"NO SUCH RECORD FOUND"<<endl;
    }
    f.close();
}

void PURCHASE_FOR_CUSTOMER()
{
    fstream f("med1.dat", ios::in | ios::out | ios::binary);
    Medicine m;

    cin.ignore(numeric_limits<streamsize>::max(),'\n');
    char nm[30];
    int quantity;

    cout<<"\n Enter Medicine name you want to buy ";
    gets(nm);

    int rec_count=0;
    bool b;
    char found='N';

    while(f.read((char*)&m,sizeof(m)))
    {
        if(strcmp(nm,m.retName())==0)
        {
            cout<<"\n Enter quantity";
            cin>>quantity;

            b=m.purchase(quantity);
            if(!b)
            {
               cout<<"SORRY! your required quantity was not present in our stock \n";
               cout<<"Please enter a valid quantity \n";
            }

            //Place content
            f.seekg(rec_count*sizeof(Medicine),ios::beg);
            f.write((char*)&m,sizeof(Medicine));
            found='Y';
            break;
        }
        rec_count++;
    }

    if(!b && found=='Y')
    {
    }
    else if(found=='Y')
    {
        m.generate_bill(quantity);
        cout<<"\n Your Purchase was successfull";
    }
    else
    {
        cout<<"\n Please Enter a valid Medicine Name";
    }
    f.close();
}

void CUSTOMER()
{
    //cout<<"Hello"<<endl;
    char ch1;
    int ch;
```

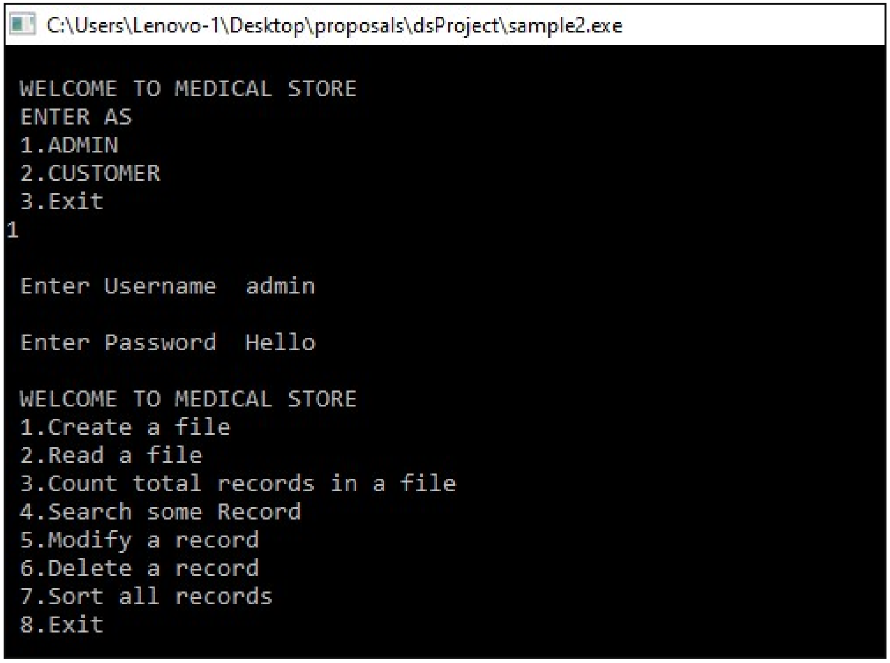
```
    do{
        cout<<"\n WELCOME TO MEDICAL STORE";
        cout<<"\n 1.Explore All Available Medicines";
        cout<<"\n 2.Search some Medicine name";
        cout<<"\n 3.Purchase some Medicine";
        cout<<"\n 4.Exit";
        cin>>ch;

        switch(ch)
        {
        case 1:
            DISPLAY_FOR_CUSTOMER();
            break;
        case 2:
            SEARCH_NAME_FOR_CUSTOMER();
            break;
        case 3:
            PURCHASE_FOR_CUSTOMER();
            break;
        case 4:
            break;
        }

        cout<<endl<<"Want to continue as Customer(y/n)--";
        cin>>ch1;
    }while(ch1=='y' || ch1=='Y');
}

int main()
{
    char ch1;
    int ch;
    do{
        cout<<"\n WELCOME TO MEDICAL STORE";
        cout<<"\n ENTER AS";
        cout<<"\n 1.ADMIN";
        cout<<"\n 2.CUSTOMER";
        cout<<"\n 3.Exit \n";
        cin>>ch;
        switch(ch)
        {
            case 1:
                {
                    bool b=LOGIN();
                    if(b)
                    {
                        ADMIN();
                    }
                    else
                    {
                        cout<<"\n Wrong Username or Password \n";
                    }
                    break;
                }
            case 2:
                {
                    CUSTOMER();
                    break;
                }
            case 3:
                {
                    break;
                }
        }
        cout<<endl<<"Want to continue in Medical Store(y/n)--";
        cin>>ch1;

    }while(ch1=='y' || ch1=='Y');
    return 0;
}
```

OUTPUT

OUTPUT

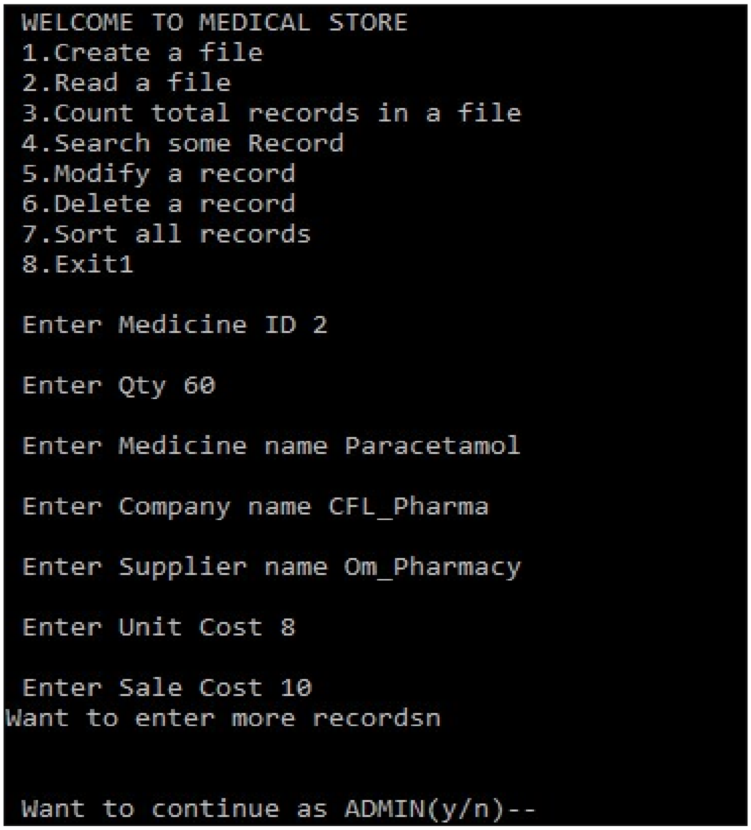
Welcome to medical store

Admin Login

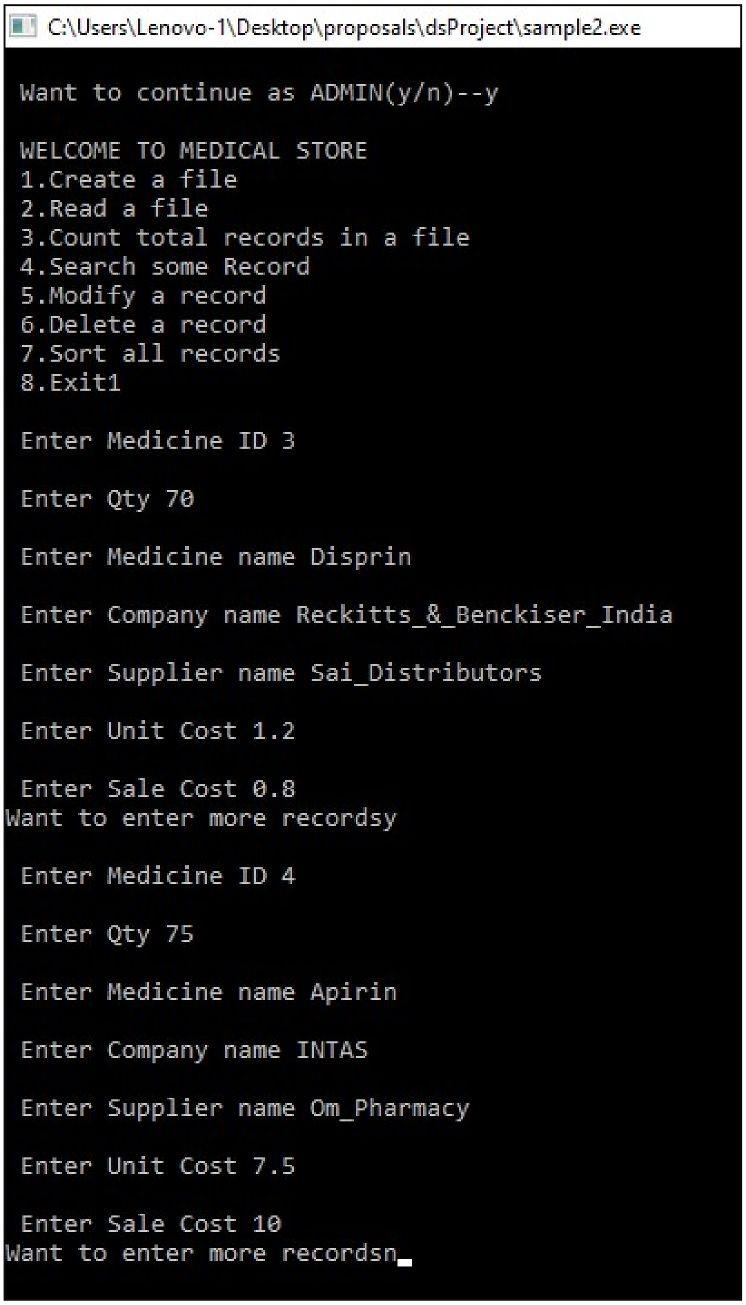


```
 WELCOME TO MEDICAL STORE
 ENTER AS
 1.ADMIN
 2.CUSTOMER
 3.Exit
1

 Enter Username  admin

 Enter Password  Hello

 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit
```

Create medicine records

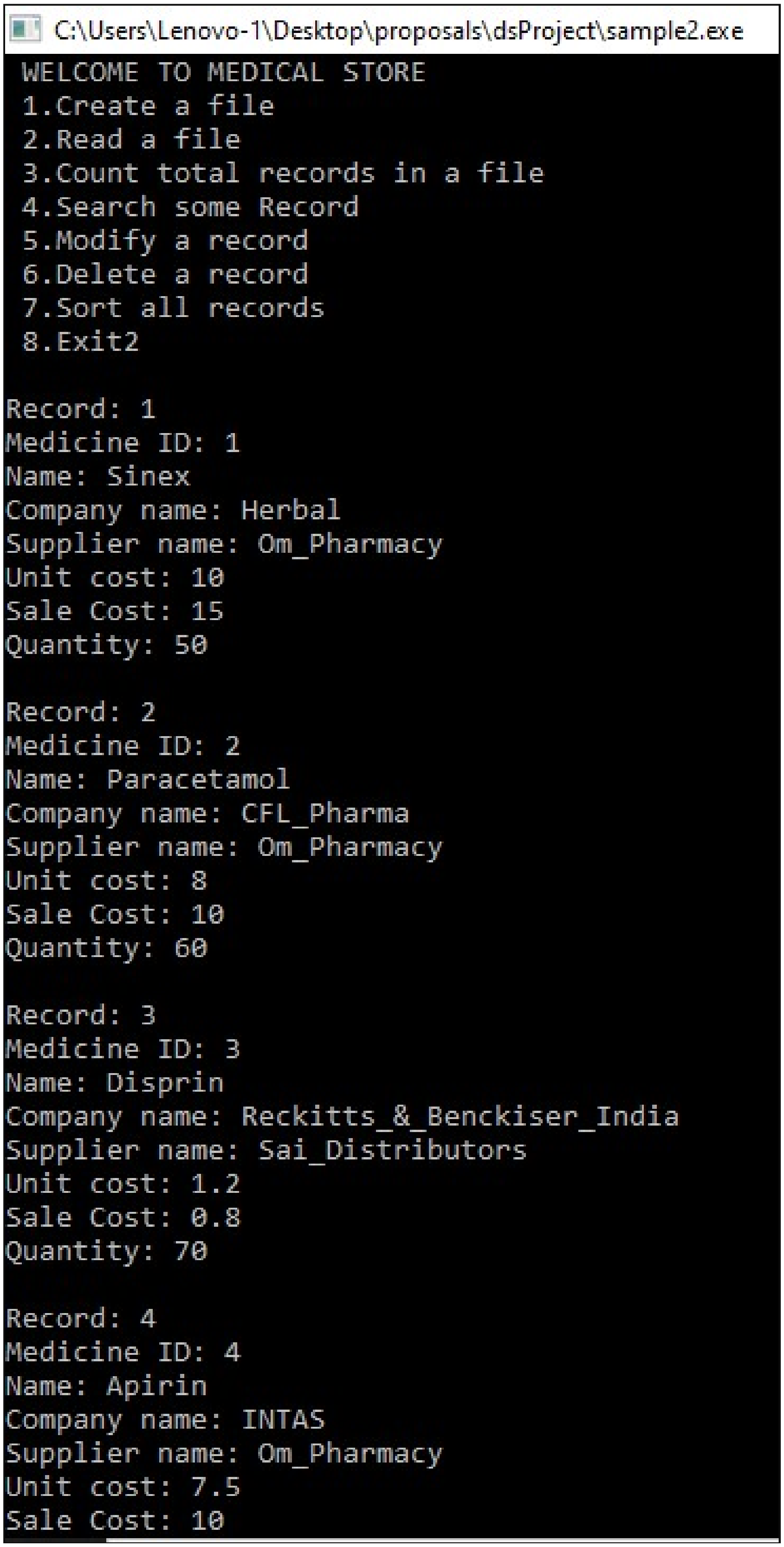


```
 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit1

 Enter Medicine ID 2

 Enter Qty 60

 Enter Medicine name Paracetamol

 Enter Company name CFL_Pharma

 Enter Supplier name Om_Pharmacy

 Enter Unit Cost 8

 Enter Sale Cost 10
Want to enter more recordsn


 Want to continue as ADMIN(y/n)--
```



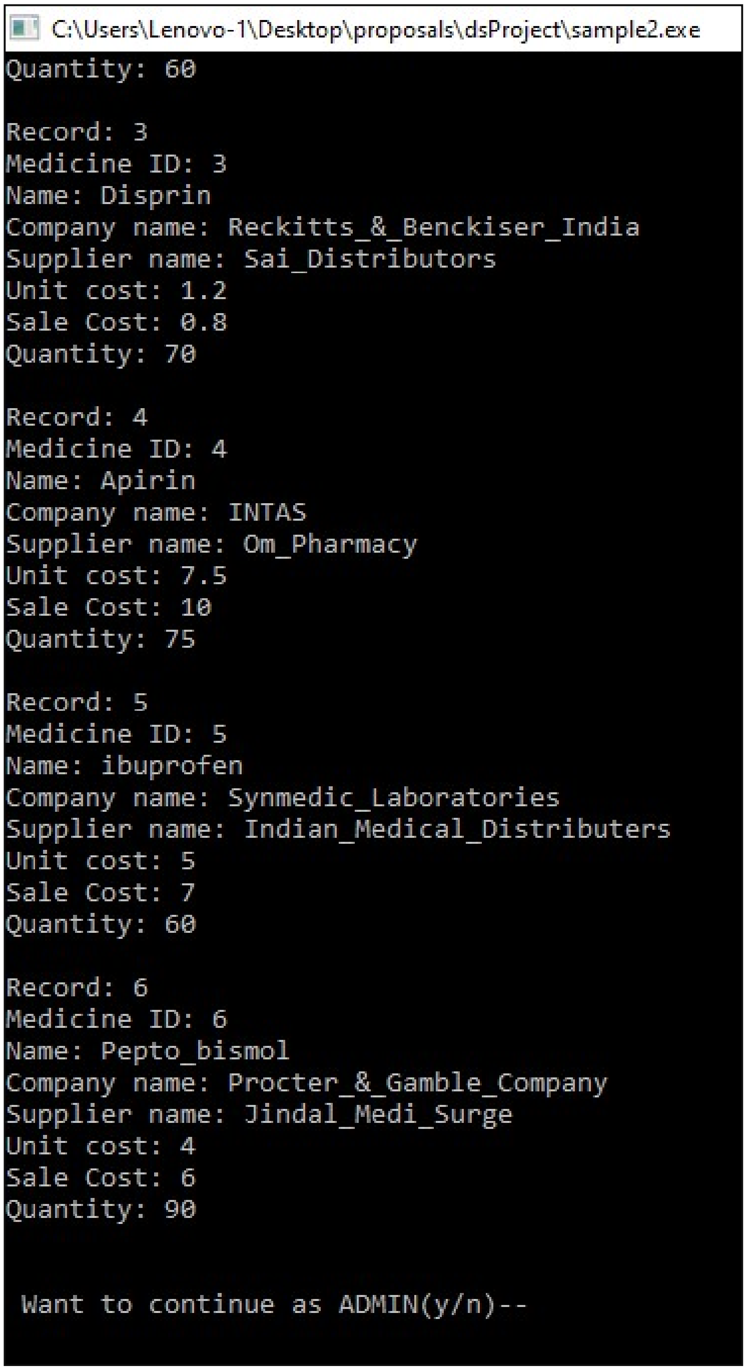
```
 Want to continue as ADMIN(y/n)--y

 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit1

 Enter Medicine ID 3

 Enter Qty 70

 Enter Medicine name Disprin

 Enter Company name Reckitts_&_Benckiser_India

 Enter Supplier name Sai_Distributors

 Enter Unit Cost 1.2

 Enter Sale Cost 0.8
Want to enter more recordsy

 Enter Medicine ID 4

 Enter Qty 75

 Enter Medicine name Apirin

 Enter Company name INTAS

 Enter Supplier name Om_Pharmacy

 Enter Unit Cost 7.5

 Enter Sale Cost 10
Want to enter more recordsn
```

Read All Records



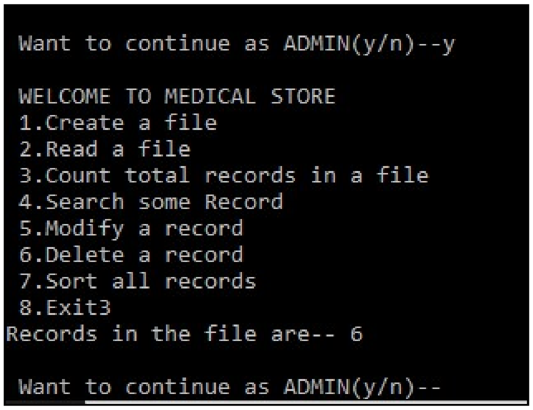
C:\Users\Lenovo-1\Desktop\proposals\dsProject\sample2.exe

```
 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit2

Record: 1
Medicine ID: 1
Name: Sinex
Company name: Herbal
Supplier name: Om_Pharmacy
Unit cost: 10
Sale Cost: 15
Quantity: 50

Record: 2
Medicine ID: 2
Name: Paracetamol
Company name: CFL_Pharma
Supplier name: Om_Pharmacy
Unit cost: 8
Sale Cost: 10
Quantity: 60

Record: 3
Medicine ID: 3
Name: Disprin
Company name: Reckitts_&_Benckiser_India
Supplier name: Sai_Distributors
Unit cost: 1.2
Sale Cost: 0.8
Quantity: 70

Record: 4
Medicine ID: 4
Name: Apirin
Company name: INTAS
Supplier name: Om_Pharmacy
Unit cost: 7.5
Sale Cost: 10
```



```
Quantity: 60

Record: 3
Medicine ID: 3
Name: Disprin
Company name: Reckitts_&_Benckiser_India
Supplier name: Sai_Distributors
Unit cost: 1.2
Sale Cost: 0.8
Quantity: 70

Record: 4
Medicine ID: 4
Name: Apirin
Company name: INTAS
Supplier name: Om_Pharmacy
Unit cost: 7.5
Sale Cost: 10
Quantity: 75

Record: 5
Medicine ID: 5
Name: ibuprofen
Company name: Synmedic_Laboratories
Supplier name: Indian_Medical_Distributers
Unit cost: 5
Sale Cost: 7
Quantity: 60

Record: 6
Medicine ID: 6
Name: Pepto_bismol
Company name: Procter_&_Gamble_Company
Supplier name: Jindal_Medi_Surge
Unit cost: 4
Sale Cost: 6
Quantity: 90


 Want to continue as ADMIN(y/n)--
```

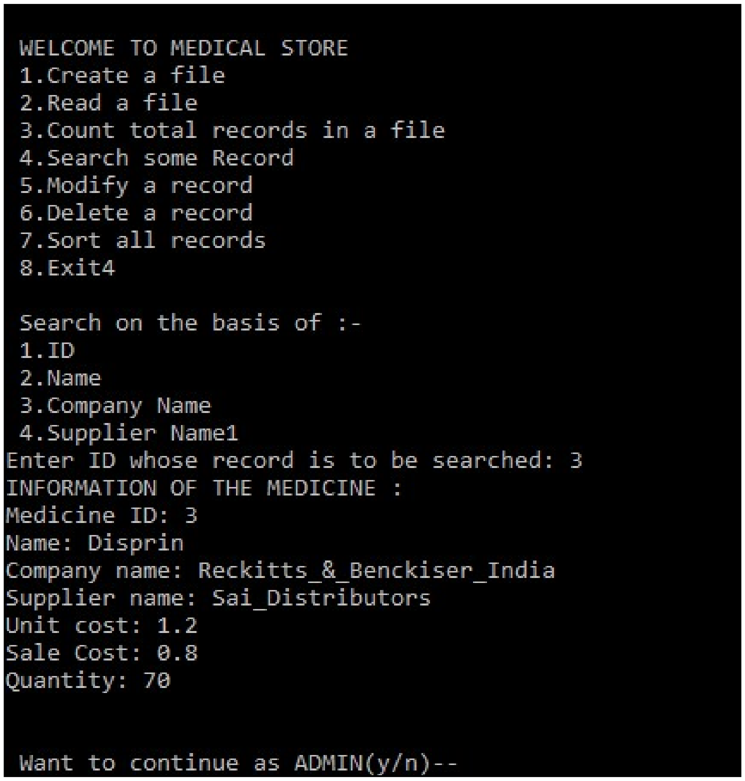
Count Records



```
 Want to continue as ADMIN(y/n)--y

 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit3
Records in the file are-- 6

 Want to continue as ADMIN(y/n)--
```

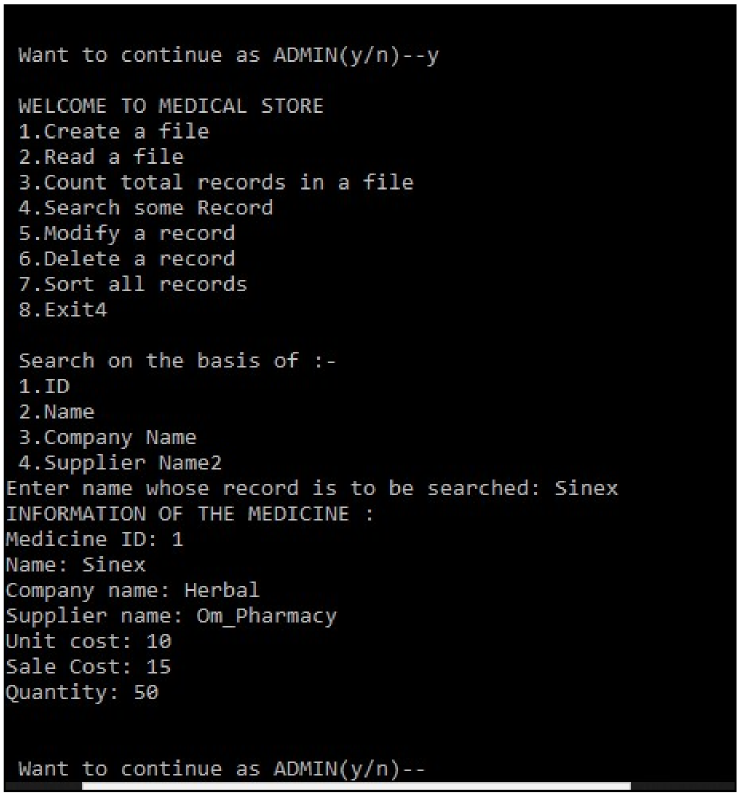
Search records

On ID basis



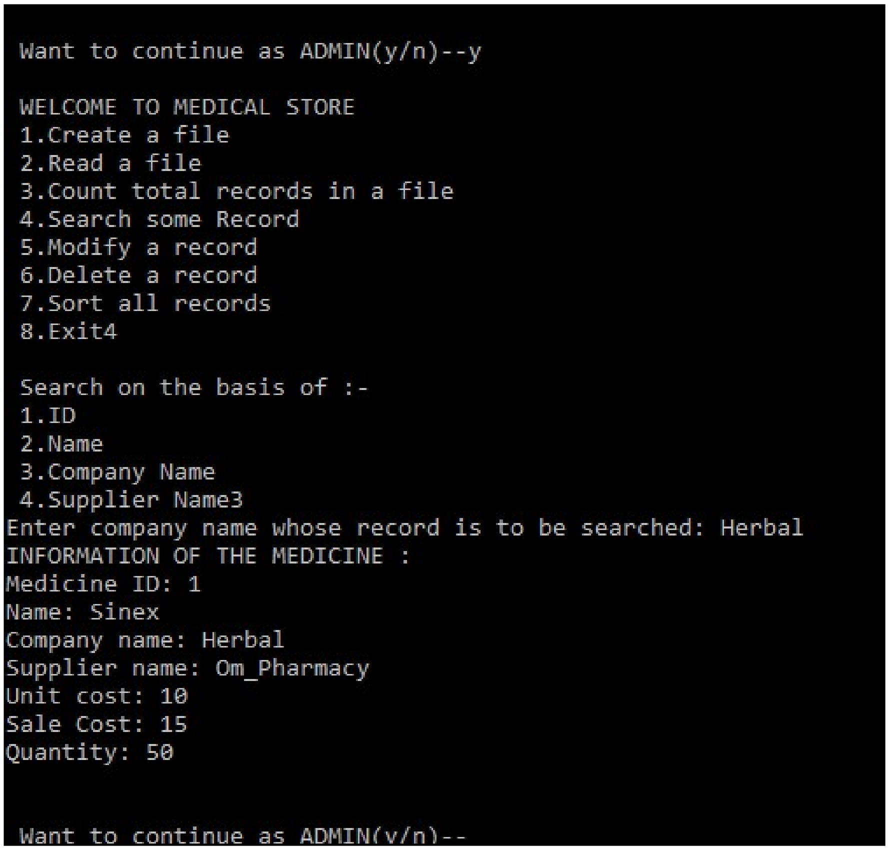
```
 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit4

 Search on the basis of :-
 1.ID
 2.Name
 3.Company Name
 4.Supplier Name1
Enter ID whose record is to be searched: 3
INFORMATION OF THE MEDICINE :
Medicine ID: 3
Name: Disprin
Company name: Reckitts_&_Benckiser_India
Supplier name: Sai_Distributors
Unit cost: 1.2
Sale Cost: 0.8
Quantity: 70


 Want to continue as ADMIN(y/n)--
```

On Name basis



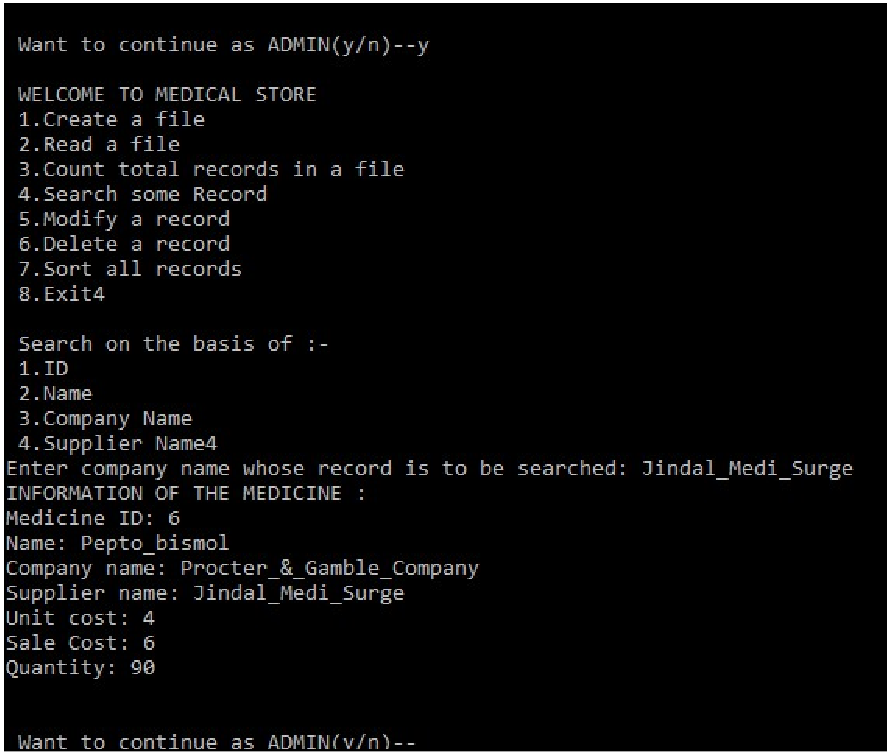
```
 Want to continue as ADMIN(y/n)--y

 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit4

 Search on the basis of :-
 1.ID
 2.Name
 3.Company Name
 4.Supplier Name2
Enter name whose record is to be searched: Sinex
INFORMATION OF THE MEDICINE :
Medicine ID: 1
Name: Sinex
Company name: Herbal
Supplier name: Om_Pharmacy
Unit cost: 10
Sale Cost: 15
Quantity: 50


 Want to continue as ADMIN(y/n)--
```

On Company Name basis



```
 Want to continue as ADMIN(y/n)--y

 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit4

 Search on the basis of :-
 1.ID
 2.Name
 3.Company Name
 4.Supplier Name3
Enter company name whose record is to be searched: Herbal
INFORMATION OF THE MEDICINE :
Medicine ID: 1
Name: Sinex
Company name: Herbal
Supplier name: Om_Pharmacy
Unit cost: 10
Sale Cost: 15
Quantity: 50


 Want to continue as ADMIN(y/n)--
```

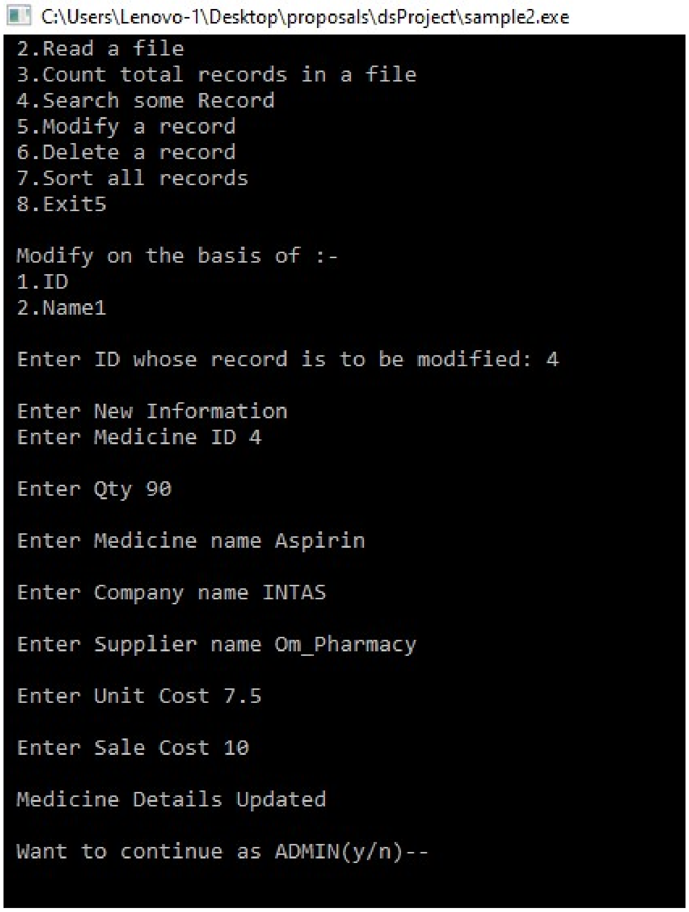
On Supplier Name basis



```
 Want to continue as ADMIN(y/n)--y

 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit4

 Search on the basis of :-
 1.ID
 2.Name
 3.Company Name
 4.Supplier Name4
Enter company name whose record is to be searched: Jindal_Medi_Surge
INFORMATION OF THE MEDICINE :
Medicine ID: 6
Name: Pepto_bismol
Company name: Procter_&_Gamble_Company
Supplier name: Jindal_Medi_Surge
Unit cost: 4
Sale Cost: 6
Quantity: 90


 Want to continue as ADMIN(y/n)--
```

Modify Records(On ID and name basis) (Here I will show only on the basis of ID, else it works perfectly when we search on the basis of Name etc.)

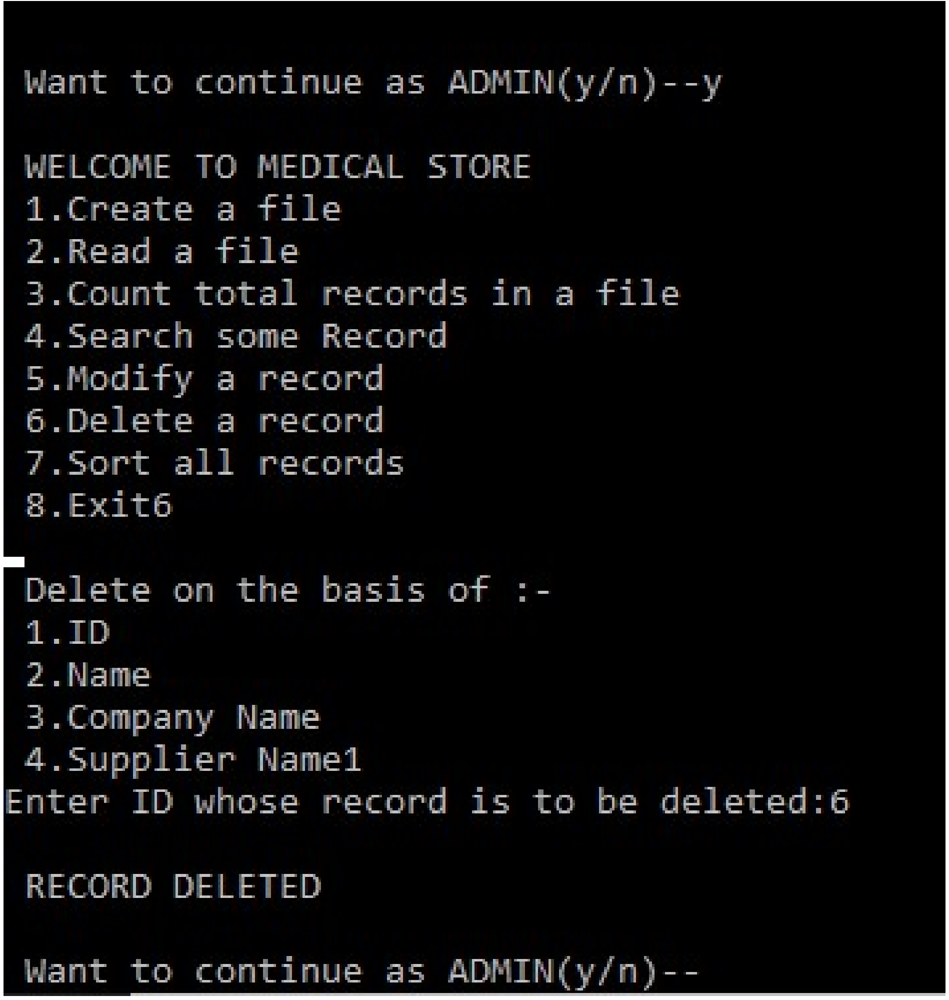
On ID basis



C:\Users\Lenovo-1\Desktop\proposals\dsProject\sample2.exe

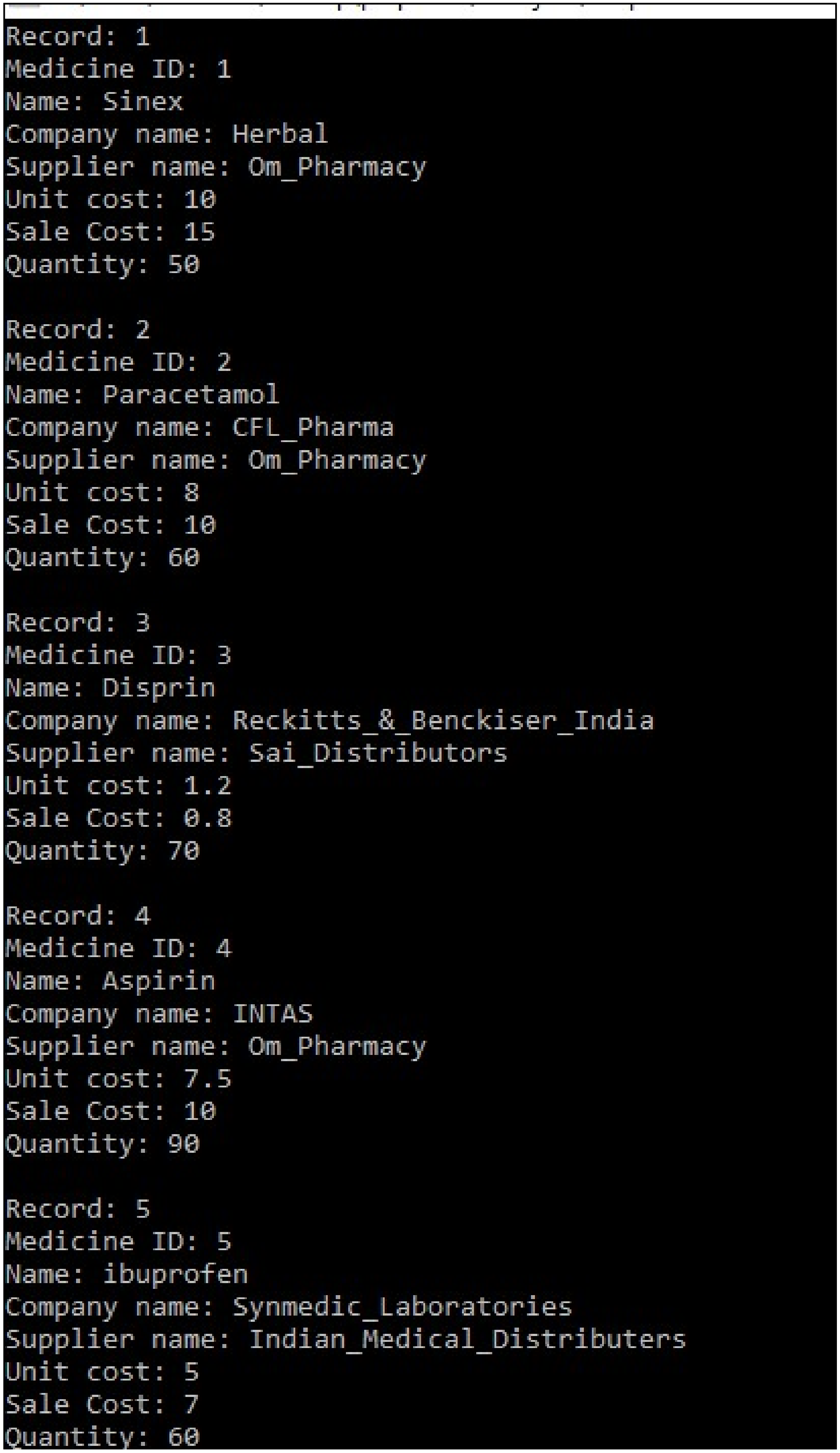
```
2.Read a file
3.Count total records in a file
4.Search some Record
5.Modify a record
6.Delete a record
7.Sort all records
8.Exit5

Modify on the basis of :-
1.ID
2.Name1

Enter ID whose record is to be modified: 4

Enter New Information
Enter Medicine ID 4

Enter Qty 90

Enter Medicine name Aspirin

Enter Company name INTAS

Enter Supplier name Om_Pharmacy

Enter Unit Cost 7.5

Enter Sale Cost 10

Medicine Details Updated

Want to continue as ADMIN(y/n)--
```

Delete Records (On ID, name, company name, supplier name basis) (Here I will show only on the basis of ID, else it works perfectly when we search on other basis etc.)



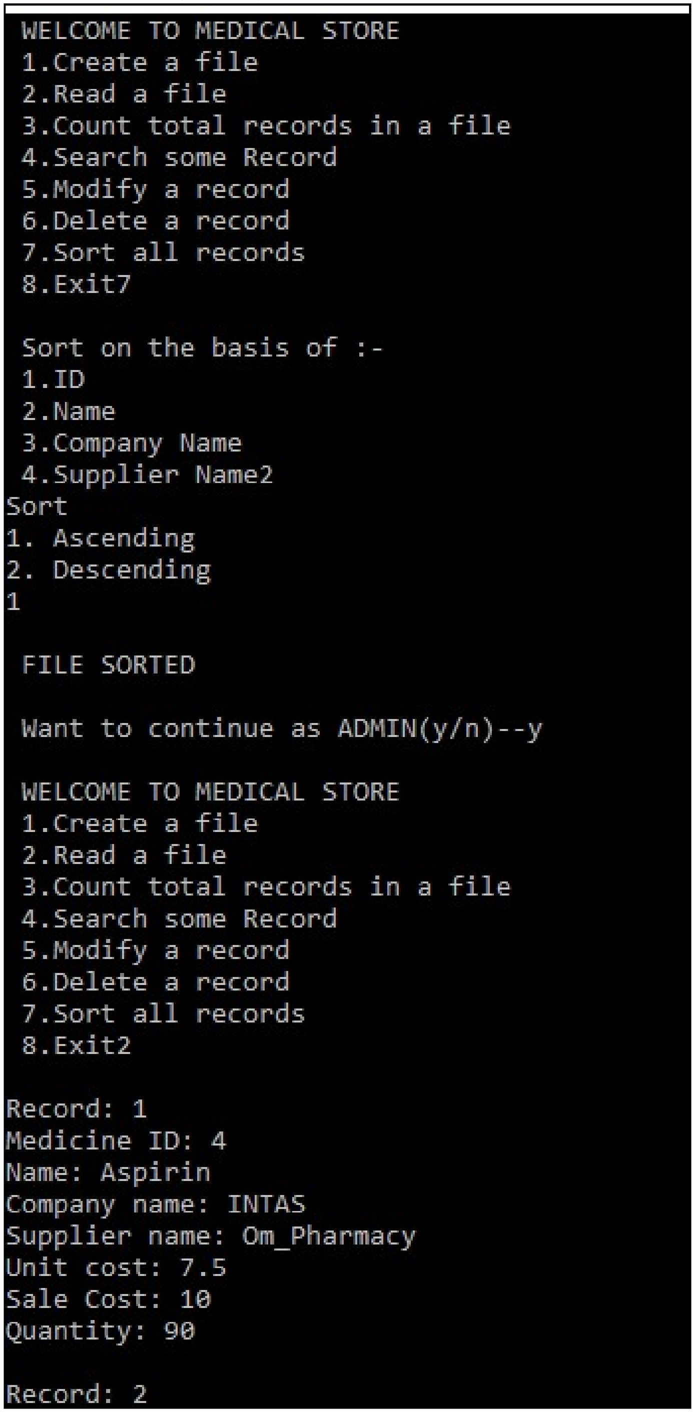
```
 Want to continue as ADMIN(y/n)--y

 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit6

 Delete on the basis of :-
 1.ID
 2.Name
 3.Company Name
 4.Supplier Name1
Enter ID whose record is to be deleted:6

 RECORD DELETED

 Want to continue as ADMIN(y/n)--
```

Here we can we record get deleted on reading all records

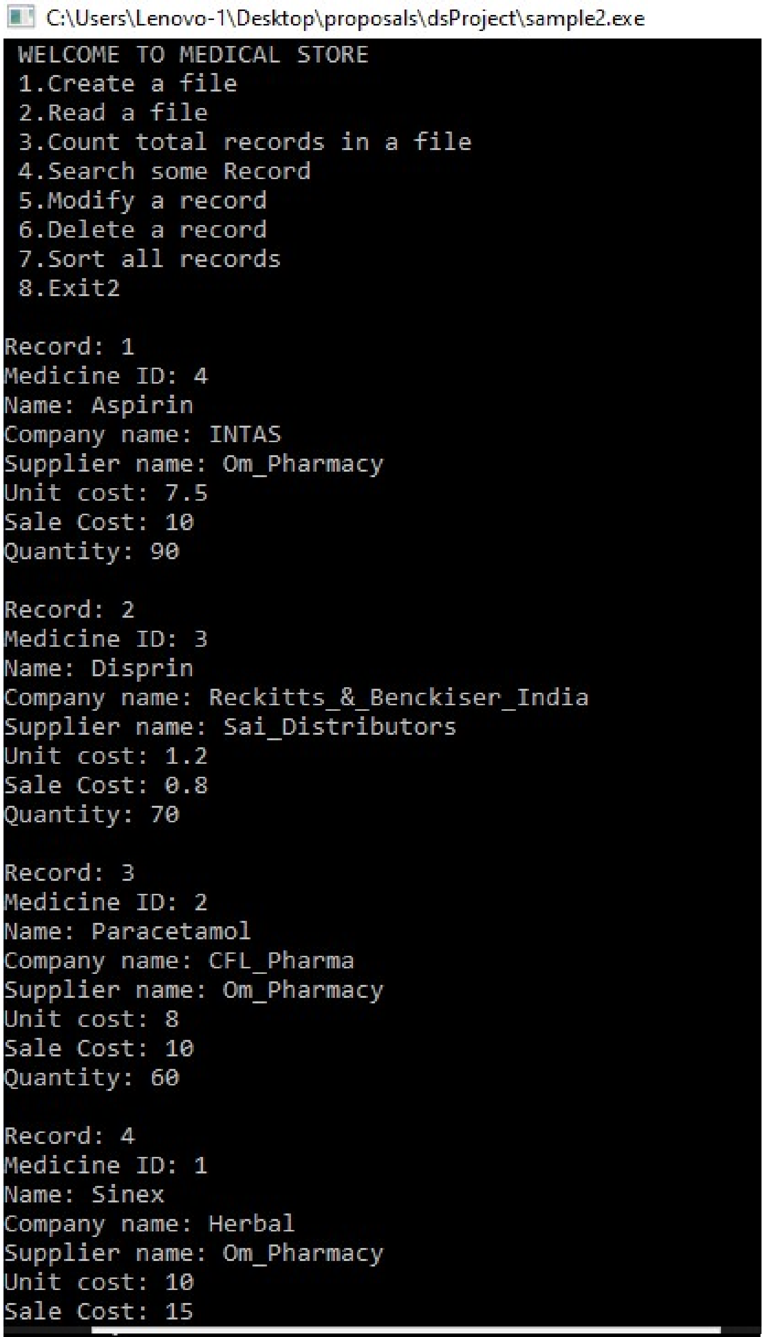


```
Record: 1
Medicine ID: 1
Name: Sinex
Company name: Herbal
Supplier name: Om_Pharmacy
Unit cost: 10
Sale Cost: 15
Quantity: 50

Record: 2
Medicine ID: 2
Name: Paracetamol
Company name: CFL_Pharma
Supplier name: Om_Pharmacy
Unit cost: 8
Sale Cost: 10
Quantity: 60

Record: 3
Medicine ID: 3
Name: Disprin
Company name: Reckitts_&_Benckiser_India
Supplier name: Sai_Distributors
Unit cost: 1.2
Sale Cost: 0.8
Quantity: 70

Record: 4
Medicine ID: 4
Name: Aspirin
Company name: INTAS
Supplier name: Om_Pharmacy
Unit cost: 7.5
Sale Cost: 10
Quantity: 90

Record: 5
Medicine ID: 5
Name: ibuprofen
Company name: Synmedic_Laboratories
Supplier name: Indian_Medical_Distributers
Unit cost: 5
Sale Cost: 7
Quantity: 60
```

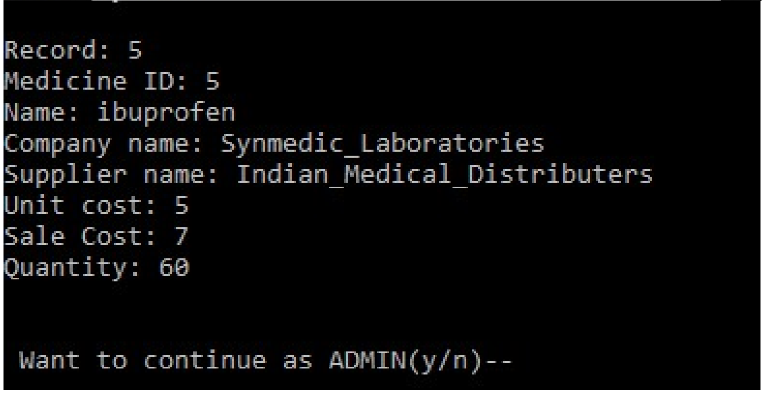
SORT all records
(On ID, name, company name, supplier name basis)(both ascending and descending) (Here I will show only on the basis of name, else it works perfectly when we search on other basis etc. in both ascending and descending ways)



```
 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit7

 Sort on the basis of :-
 1.ID
 2.Name
 3.Company Name
 4.Supplier Name2
Sort
1. Ascending
2. Descending
1

 FILE SORTED

 Want to continue as ADMIN(y/n)--y

 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit2

Record: 1
Medicine ID: 4
Name: Aspirin
Company name: INTAS
Supplier name: Om_Pharmacy
Unit cost: 7.5
Sale Cost: 10
Quantity: 90

Record: 2
```



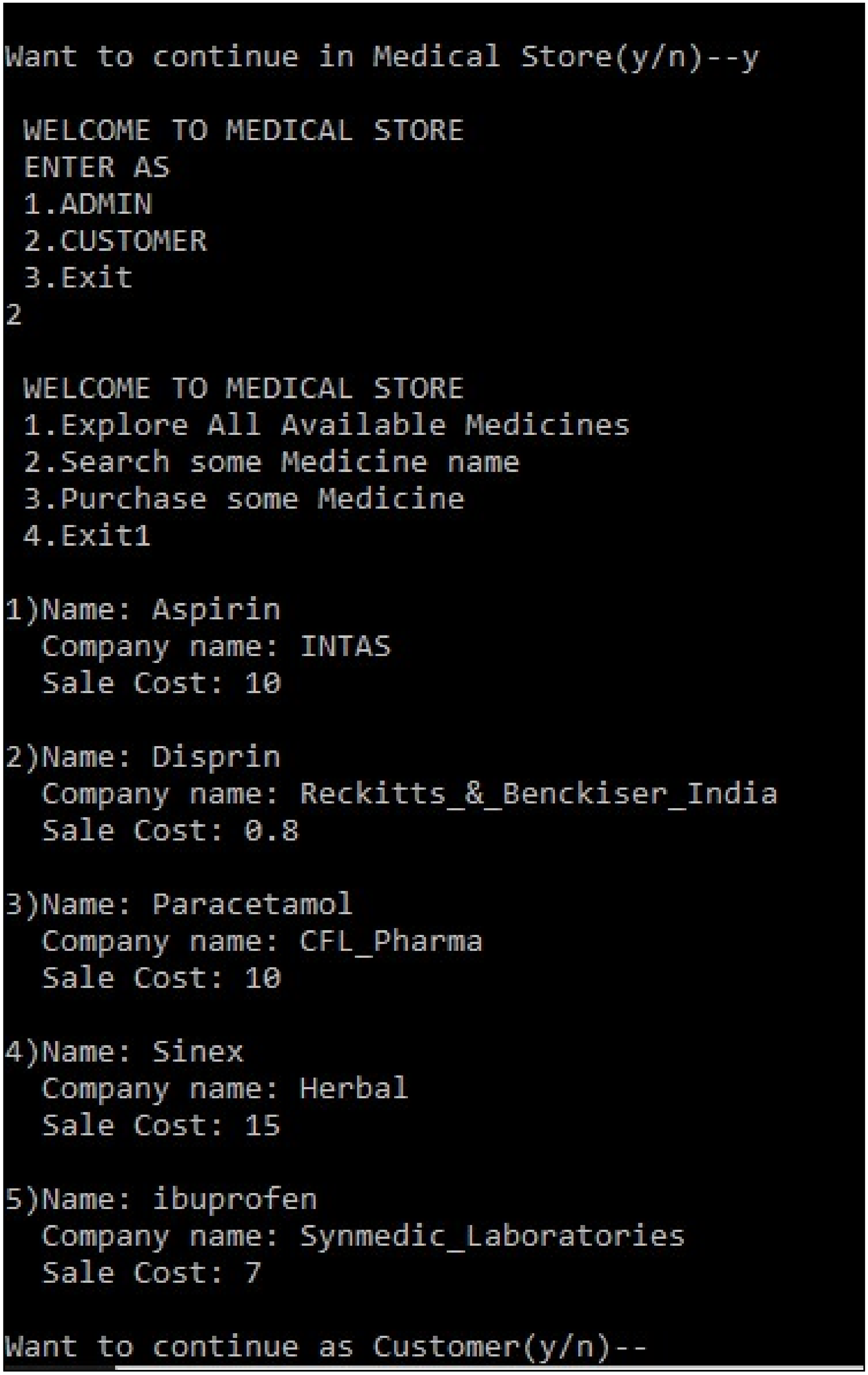
```
 WELCOME TO MEDICAL STORE
 1.Create a file
 2.Read a file
 3.Count total records in a file
 4.Search some Record
 5.Modify a record
 6.Delete a record
 7.Sort all records
 8.Exit2

Record: 1
Medicine ID: 4
Name: Aspirin
Company name: INTAS
Supplier name: Om_Pharmacy
Unit cost: 7.5
Sale Cost: 10
Quantity: 90

Record: 2
Medicine ID: 3
Name: Disprin
Company name: Reckitts_&_Benckiser_India
Supplier name: Sai_Distributors
Unit cost: 1.2
Sale Cost: 0.8
Quantity: 70

Record: 3
Medicine ID: 2
Name: Paracetamol
Company name: CFL_Pharma
Supplier name: Om_Pharmacy
Unit cost: 8
Sale Cost: 10
Quantity: 60

Record: 4
Medicine ID: 1
Name: Sinex
Company name: Herbal
Supplier name: Om_Pharmacy
Unit cost: 10
Sale Cost: 15
```



```
Record: 5
Medicine ID: 5
Name: ibuprofen
Company name: Synmedic_Laboratories
Supplier name: Indian_Medical_Distributers
Unit cost: 5
Sale Cost: 7
Quantity: 60


 Want to continue as ADMIN(y/n)--
```

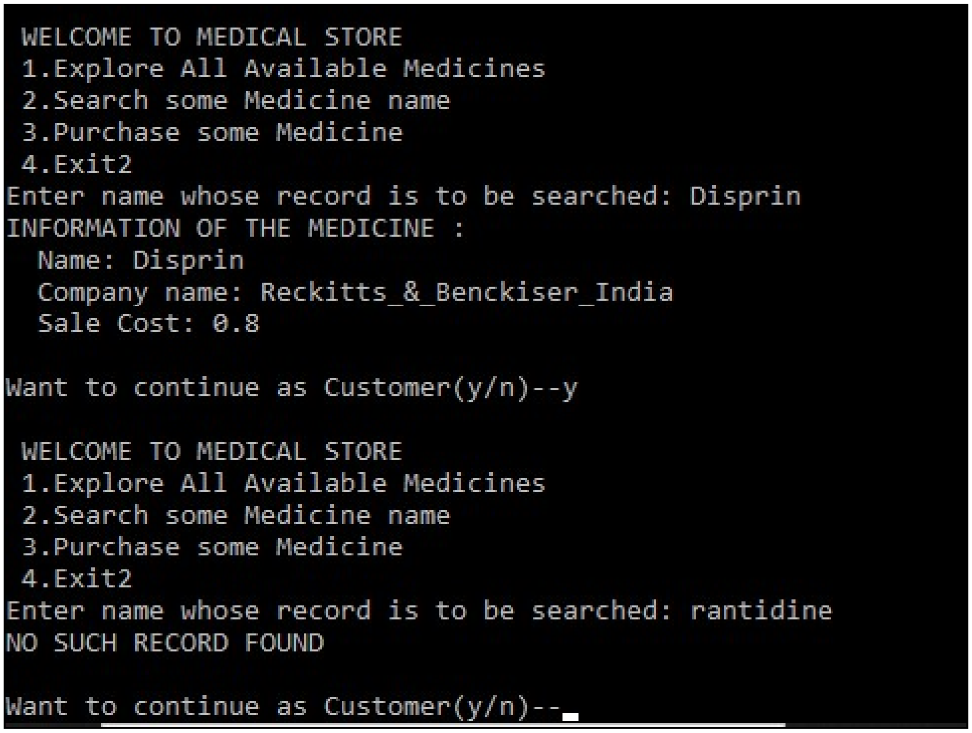
Customer Panel

Explore All medicines



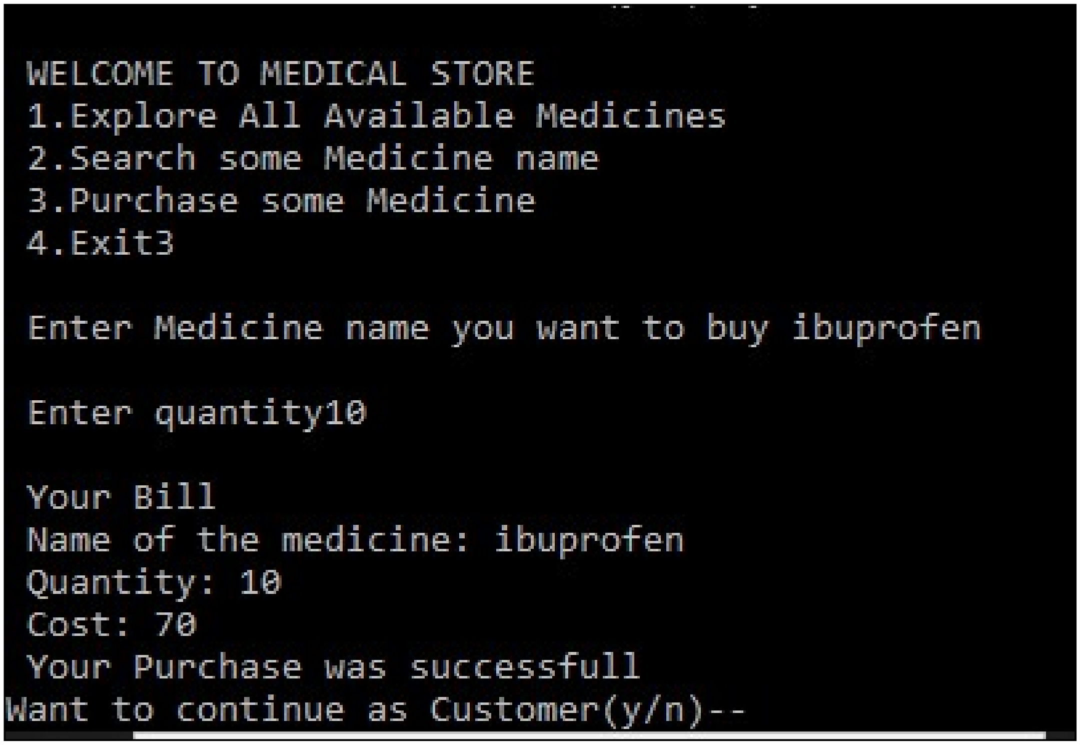
```
Want to continue in Medical Store(y/n)--y

 WELCOME TO MEDICAL STORE
 ENTER AS
 1.ADMIN
 2.CUSTOMER
 3.Exit
2

 WELCOME TO MEDICAL STORE
 1.Explore All Available Medicines
 2.Search some Medicine name
 3.Purchase some Medicine
 4.Exit1

1)Name: Aspirin
  Company name: INTAS
  Sale Cost: 10

2)Name: Disprin
  Company name: Reckitts_&_Benckiser_India
  Sale Cost: 0.8

3)Name: Paracetamol
  Company name: CFL_Pharma
  Sale Cost: 10

4)Name: Sinex
  Company name: Herbal
  Sale Cost: 15

5)Name: ibuprofen
  Company name: Synmedic_Laboratories
  Sale Cost: 7

Want to continue as Customer(y/n)--
```

Search some medicine



```
 WELCOME TO MEDICAL STORE
 1.Explore All Available Medicines
 2.Search some Medicine name
 3.Purchase some Medicine
 4.Exit2
Enter name whose record is to be searched: Disprin
INFORMATION OF THE MEDICINE :
  Name: Disprin
  Company name: Reckitts_&_Benckiser_India
  Sale Cost: 0.8

Want to continue as Customer(y/n)--y

 WELCOME TO MEDICAL STORE
 1.Explore All Available Medicines
 2.Search some Medicine name
 3.Purchase some Medicine
 4.Exit2
Enter name whose record is to be searched: rantidine
NO SUCH RECORD FOUND

Want to continue as Customer(y/n)--
```

Purchase some medicine



```
 WELCOME TO MEDICAL STORE
 1.Explore All Available Medicines
 2.Search some Medicine name
 3.Purchase some Medicine
 4.Exit3

 Enter Medicine name you want to buy ibuprofen

 Enter quantity10

 Your Bill
 Name of the medicine: ibuprofen
 Quantity: 10
 Cost: 70
 Your Purchase was successfull
Want to continue as Customer(y/n)--
```