```
#include <fstream>
#include <iostream>
#include <string>
#include <cstring>
#include <conio.h>
#include <ctype.h>
#include <stdlib.h>
#include <sstream>

        public :
void Create(string col[],int n){
                        ofstream file;
                        char filename[198];
                        string input;
                        int i=0;
                        char exit;
                        system("cls");
                        cout <<">> Enter filename with extension: ";
                        cin >> filename;


                    file.open (filename);
                    char c;
                    while(i<n){
                                file<< trim(col[i++])<<",";
```

```
                }
                file<<"\n";
                system("cls");
                cout <<">> Enter information\n";
                cin.ignore();
        do{

                for(i=0;i<n;++i){
                cout << trim(col[i])<<": ";
                        getline(cin,input);


                        file<<trim(input)<<",";
                }
                cout<<"\nEnter [ESC]key to exit or [Enter]key to continue";
                file<<"\n";


                exit=getch();
                        if(exit == 27)
                                break;
                        else if (exit=13)
                                system("cls");
                cout<<"\n";
        }while(exit!=27);
    file.close();
}
```

```
void IsSameStucture(){
	string file1;
	char filename1[198];
	string file2;
	char filename2[198];
	char * pch;
	char str[32767];
	string fileCol1[32767];
	string fileCol2[32767];
	int n1=0,n2=0;
	string cell,line1,line2;
	int i,j;
	cout<<">> Enter filename1: ";
	cin >>file1;
	if (invalidFilename(file1)){
		cout <<">> Invalid filename: do not use these characters \n \\ / : * ? \" < > |\n";
		return;
	}
	strcpy(filename1,file1.c_str());
	cout <<">> Enter filename2: ";
	cin >>file2;
	strcpy(filename2,file2.c_str());

	  ifstream fileOne ( filename1 );
```

```
   // Always check to see if file opening succeeded


      while(fileOne.good()){
       getline(fileOne,line1);
                    break;
      }
      fileOne.close();


          ifstream fileTwo ( filename2 );


      while(fileTwo.good()){
       getline(fileTwo,line2);
                    break;
      }
      fileTwo.close();


strcpy(str,line1.c_str());
  pch = strtok (str," ,");
  while (pch != NULL){
          fileCol1[n1++] = pch;
          pch = strtok (NULL, ",");
}


strcpy(str,line2.c_str());
  pch = strtok (str," ,");
```

```
        while (pch != NULL){
                fileCol2[n2++] = pch;
                pch = strtok (NULL, ",");
        }

        if(n1==n2){
        for(i=0;i<n1;++i)
                if(fileCol1[i]!= fileCol2[i]){
                        cout <<"\n>> "<<filename1<<" and "<<filename2<<" do not have same
structure!\n";
                        break;
                }
cout <<"\n>> "<<filename1<<" and "<<filename2<<" have same structure!\n";
        }
        else
                cout <<"\n>> "<<filename1<<" and "<<filename2<<" do not have same structure!\n";
system("pause");
system("cls");
}
        void IsDuplicate(string s3){
                string str1="";
                string str2="";
                char filename[198];
                string line;
                 int lines=0;
                 int i=0,j=0,c;
```

```
        bool isAny=false;


       strcpy(filename,s3.c_str());
ifstream file ( filename );
// Always check to see if file opening succeeded
if ( !file.is_open() )
  cout<<">> Error: Could not open file '"<<filename<<"'\n";
   while(file.good()){
    getline(file,str1);
    ++lines;
   }
   file.close();


   if(lines<=1)
       cout<< ">> nothing to compare!";
   else {
       ifstream file ( filename );
          for(i=0;i<lines;++i){
                        getline(file,str1);
                        c=1;
                        ifstream file1 ( filename );
                                getline(file1,str2);


                for(j=0;j<lines;++j){
                        if(str1.compare(str2) == 0)
```

```
                                        if(i<j)
                                                ++c;
                                        else if (i>j)
                                                break;
                                getline(file1,str2);
                        }
                        file1.close();
                        if(c>1){
                                cout <<">> "<<str1<<" occured "<<c<<" times\n";
                                isAny=true;
                        }
                }
                file.close();
                }
                        cout <<"There are no duplicate entries in file: "<< filename;
                        cout<<endl;
        system("pause");
        system("cls");
        }
void RemoveDuplicate(string s3){
                        string str1="";
                        string str2="";
                        char filename[198];
                        char filenameTemp[193];
                        string line;
```

```
                int lines=0;
                int i=0,j=0,c;
                bool isAny=false;
                bool write=true;
               strcpy(filename,s3.c_str());
strcpy (filenameTemp, filename) ;
               strcat (filenameTemp, ".temp") ;
               ifstream file(filename);

               while(file.good()){
                        getline(file,str1);
                        ifstream file1(filenameTemp);
                        write=true;

                        while(file1.good()){
                                getline(file1,str2);
                                if(str1.compare(str2)==0){

                                        write=false;
                                        if(str1.length()>1){
                                                isAny=true;
                                        }
                                        break;
                                }
                        }
```

```
                        file1.close();
                        if(write){
                                ofstream file2(filenameTemp,ios::app);
                                file2<<str1<<endl;
                                file2.close();
                        }

                }

            file.close();

                cout <<">> updated data saved to: "<<filename<<endl;
                        remove (filename);
                        rename(filenameTemp,filename);
        system("pause");
        system("cls");
        }
void csvSort(string filename){
         ifstream file(filename.c_str());
   string str;
   int n1=0,i;
   string colName;
   string col[10000];
```

```
getline(file,str);
   replace( str.begin(), str.end(), ',', ' ' );
   stringstream ssin1(str);
   while (ssin1.good() && n1 < 10000){
ssin1 >> col[n1];
++n1;
}
if(n1>=10000)
        cout << ">> Maximum column limit 10000 reached!";
cout<<">> Choose column name from "<<filename<<endl;
cin >>colName;
for(i=0;i<n1;i++)
                if(strcmp(trim(col[i]).c_str(),trim(colName).c_str())==0)
                        break;
                if(i==n1){
                        cout<<">> "<<colName<<" does not exists in "<<filename<<endl;
                                system("pause");
                                system("cls");
                return;
                }
                ofstream file3;
                file3.open ("fun5.temp");
   string line;int k;
      while(file.good()){
          for(k=0;k<=i;++k){
```

```
                    getline(file,line,',');
                        //cout <<"\nk="<<k<<line1;
                  }
                  if(line.length()>1)
                     file3 <<line<<endl;
                  getline(file,line);
                }
file3.close();

  ifstream file5(filename.c_str());
  vector<string> rows;
  getline(file5, line);
    while(file5.good())
    {
       getline(file5, line);
       if(line.length()>1)
       rows.push_back(line);
    }
 std::sort(rows.begin(), rows.end());
 ofstream outFile("sortFun5.csv");
    std::vector<std::string>::iterator iterator = rows.begin();
    for(; iterator != rows.end(); ++iterator)
       outFile << *iterator << std::endl;
outFile.close();
remove ("fun5.temp");
```

```
cout <<">> sorted column saved  to sortFun5.csv"<<endl;
        system("pause");
        system("cls");
}
void Union(){

                char filename1[198];
                char filename2[198];
                string f1,f2;
                string str1,str2;
                string colName;
                 int n1=0,n2=0,i=0,j=0;
                 string col1[10000];
                string col2[10000];
                cout << "Enetr file 1: ";
                cin >>f1;
                cout << "Enter file 2: ";
                cin >>f2;
                strcpy(filename1,f1.c_str());
                strcpy(filename2,f2.c_str());

            ifstream file1 ( filename1 );
            ifstream file2 ( filename2 );

            getline(file1,str1);
```

```
   getline(file2,str2);
   replace( str1.begin(), str1.end(), ',', ' ' );
   stringstream ssin1(str1);
   while (ssin1.good() && n1 < 10000){
ssin1 >> col1[n1];
++n1;
}
if(n1>=10000)
        cout << ">> Maximum column limit 10000 reached!";
cout<<">> Choose column name from "<<filename1<<endl;

cout<<endl<<">> Or from "<<filename2<<endl;

   replace( str2.begin(), str2.end(), ',', ' ' );

   stringstream ssin2(str2);

   while (ssin2.good() && n2 < 10000){
ssin2 >> col2[n2];
++n2;
}
if(n2>=10000)
        cout << ">> Maximum column limit 10000 reached!";
        cout<<endl<<"Enter column name: ";
        cin >>colName;
```

```
for(i=0;i<n1;i++)
        if(strcmp(trim(col1[i]).c_str(),trim(colName).c_str())==0)
                break;
for(j=0;j<n2;j++)
        if(strcmp(trim(col2[j]).c_str(),trim(colName).c_str())==0)
                break;


        if(i==n1){
                cout<<">> "<<colName<<" does not exists in "<<filename1<<endl;
                        system("pause");
                        system("cls");
        return;
        }
        if(j==n2){
                cout<<">> "<<colName<<" does not exists in "<<filename2<<endl;
                        system("pause");
                        system("cls");
        return;
        }


        ofstream file3;
        file3.open ("file3");
        int k=0,l=0;
```

```
                        file3<<"+"<<colName<<"+"<<endl;


                string line1,line2;
                  while(file1.good()){
                        for(k=0;k<=i;++k){
                    getline(file1,line1,',');
                        //cout <<"\nk="<<k<<line1;
                }
                if(line1.length()>1)
                    file3 <<line1<<endl;
                 getline(file1,line1);
               }


    while(file2.good()){
                        for(k=0;k<=j;++k){
                    getline(file2,line1,',');
                        //cout <<"\nk="<<k<<line1;
                 }
                    file3 <<line1<<endl;
                     getline(file2,line1);
               }
file3.close();
bool write=true;
bool isAny=false;
ifstream file("file3");
```

```
                while(file.good()){
                        getline(file,str1);
                        ifstream file1("file3.temp");
                        write=true;

                        while(file1.good()){
                                getline(file1,str2);
                                if(str1.compare(str2)==0){

                                        break;
                                }
                        }

                        file1.close();
                        if(write){
                                ofstream file2("file3.temp",ios::app);
                                file2<<str1<<endl;
                                file2.close();
                        }

                }
        file.close();
                        cout <<">> Union of "<< filename1<<" and "<<filename2<<" saved to
file3"<<endl;
                        remove ("file3");
```

```
                        rename("file3.temp","file3");
        system("pause");
        system("cls");
}
void Intesection(){
                char filename1[198];
                char filename2[198];
                string f1,f2;
                string str1,str2;
                string colName;
                 int n1=0,n2=0,i=0,j=0;
                 string col1[10000];
                string col2[10000];
                cout << "Enetr file 1: ";
                cin >>f1;
                cout << "Enter file 2: ";
                cin >>f2;
                strcpy(filename1,f1.c_str());
                strcpy(filename2,f2.c_str());
                bool isAny=false;
                 bool write=false;

           ifstream file1 ( filename1 );
           ifstream file2 ( filename2 );
```

```
   // Always check to see if file opening succeeded
   if ( !file1.is_open() ){
     cout<<">> Error: Could not open file '"<<filename1<<"'\n";
     return;
   }
   if ( !file2.is_open() ){
     cout<<">> Error: Could not open file '"<<filename2<<"'\n";
     return;
   }

   getline(file1,str1);
   getline(file2,str2);
   replace( str1.begin(), str1.end(), ',', ' ' );
   stringstream ssin1(str1);
   while (ssin1.good() && n1 < 10000){
ssin1 >> col1[n1];
++n1;
}
if(n1>=10000)
            cout << ">> Maximum column limit 10000 reached!";
cout<<">> Choose column name from "<<filename1<<endl;

cout<<endl<<">> Or from "<<filename2<<endl;

   replace( str2.begin(), str2.end(), ',', ' ' );
```

```
   stringstream ssin2(str2);


   while (ssin2.good() && n2 < 10000){
ssin2 >> col2[n2];
++n2;
}
if(n2>=10000)
        cout << ">> Maximum column limit 10000 reached!";


        cout<<endl<<"Enter column name: ";
        cin >>colName;


        for(i=0;i<n1;i++)
                if(strcmp(trim(col1[i]).c_str(),trim(colName).c_str())==0)
                        break;
        for(j=0;j<n2;j++)
                if(strcmp(trim(col2[j]).c_str(),trim(colName).c_str())==0)
                        break;


                if(i==n1){
                        cout<<">> "<<colName<<" does not exists in "<<filename1<<endl;
                                system("pause");
                                system("cls");
                return;
```

```
                    }
                    if(j==n2){
                            cout<<">> "<<colName<<" does not exists in "<<filename2<<endl;
                                    system("pause");
                                    system("cls");
                    return;
                    }
                    file1.close();
                    file2.close();
string line1,line2;int k;
ifstream fileR1(filename1);
getline(fileR1,line1);
remove("file3");
            while(fileR1.good()){
                    write=false;

     for(k=0;k<=i;++k){
       getline(fileR1,line1,',');
          //cout <<"\nk="<<k<<line1;
    }

                    ifstream fileR2(filename2);
                            getline(fileR2,line2);

    while(fileR2.good()){
```

```
     for(k=0;k<=j;++k){
        getline(fileR2,line2,',');
            //cout <<"\nk="<<k<<line1;
       }


                                if(line1.compare(line2)==0){
                                        if(line1.length()>1){
                                                write=true;
                                                isAny=true;
                                        }
                                        break;
                                }
                                getline(fileR2,line2);
            }
                        fileR2.close();


            if(write==true){
                                ofstream file3("file3",ios::app);
                                file3<<line1<<endl;
                                file3.close();
                        }
getline(fileR1,line1);
            }
```

```
        fileR1.close();


        if(isAny==false)
            cout<<">> There were no duplicate entries!"<<endl;
        else
            cout <<">> updated data saved to: file3"<<endl;
        system("pause");
        system("cls");


}
//-------------------------------------------------------------------------------------
//              READ FUNCTION
//-------------------------------------------------------------------------------------


    void read (string s3){
            string str="";
            char filename[198];
            string line;
            if ( s3.length()>197 ){
         cout<<">> Error: Could not open file \n";
         return;
            }
            strcpy(filename,s3.c_str());
        ifstream file ( filename );
        // Always check to see if file opening succeeded
```

```
            if ( !file.is_open() )
              cout<<">> Error: Could not open file '"<<filename<<"'\n";
            else {
               while(file.good()){
                getline(file,str,',');
                std::cout <<str<<"\t\t";
               }
               file.close();
            }
                    cout<<endl;
        system("pause");
        system("cls");
        }
};

//---------------------------------------------------------------------------------------
//                    MAIN FUNCTION
//---------------------------------------------------------------------------------------
int main ( int argc, char *argv[] )
{ Mini ob;
char choice='1';
string c1,c2,input;char exit;
int i=0;
string col[sizeof(int)];
  if ( argc != 3 ) // argc should be 2 for correct execution
```

```
  // We print argv[0] assuming it is the program name
  cout<<">> usage: "<< argv[0] <<" <filename1>.csv <filename2>.csv\n";
 else {
        system("cls");
        while(choice != '0'){
                cout <<"\nEnter\n1 to create a *.csv file \n2 whether they are same structure or not "<<"\n3 to find duplicate row entries\n4 to Remove Duplicate raw entry in CSV file\n5 to sort column contentn\n6 to find union of two *.csv files\n7 to find intersection of two files based on column name \n\n0 to Exit";
                cout<<"\n[Data in files are case senstive]\n------------------------------------------------------------\n>> ";
                cin >> choice;
                if (choice =='1'){
                        system("cls");
                        cout <<"\nEnter column names one by one..[ESC]key to go to main menu\n";
                        i=0;
                        cout <<">> Enter column name: ";
                        exit = getche();
                        cin.ignore();
        if(exit!=27){
                do{

                        getline(cin , input);
                        col[i++]=string(1,exit)+input;
                        cout <<">> Enter column name: ";
                        exit = getche();
```

```
                    }
                    while(exit!=27);
}else{
        system("cls");
        cout<< ">>Error: Enter atleast one column name\n";
        continue;
        }
                ob.Create(col,i);
}// choice if 1


else if (choice=='2'){
        ob.IsSameStucture();
}
else if (choice=='3'){
                    system("cls");
                    cout << ">> Enter filename to scan: ";
                    cin >>input;
                    ob.IsDuplicate(input);
}
else if(choice=='4'){
        system("cls");
                    cout << ">> Enter filename to search: ";
                    cin >>input;
                    ob.RemoveDuplicate(input);
}
```

```
            else if(choice=='5'){
            system("cls");
                        cout << ">> Enter filename to sort: ";
                        cin >>input;
                        ob.csvSort(input);
}
            else if(choice=='6'){
            system("cls");
                        ob.Union();
}
else if(choice=='7'){
            system("cls");
            ob.Intesection();
}
 else if (choice == '9'){
                        system("cls");
                        cout << ">> Enter filename to read: ";
                        cin >>input;
                        ob.read(input);
              }
              else if (choice!=0){
                        system("cls");
                        cout<<">>Error: Invalid input\n";
                      }
 }
```

```
  }

}
```