**Agnim Gupta**

**2028083**

**A23, CSSE**

**Question 1**

**Two files named ‘Source1’ and ‘Source2’ contains sorted list of integers . Write a program that reads the contents of both the files and stores the merged list in sorted form in a new file named ‘Target’**

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

    const int size = 80;

    char line[size];

    ifstream fin1, fin2;

    fin1.open("Source1.txt");

    fin2.open("Source2.txt");

    for (int i = 1; i <= 10; i++)

    {

        if (fin1.eof() != 0)

        {

            cout << "End of files." << endl;

            exit(1);

        }

        fin1.getline(line, size);

        cout << line << "\n";

        if (fin2.eof() != 0)

        {

            cout << "Exit from the Capital.";

            exit(1);

        }

        fin2.getline(line, size);

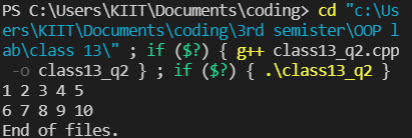
        cout << line << "\n";

    }

    return 0;

}

**Output**

****

**Question 2**

#include <iostream>

#include <iomanip>

#include <fstream>

using namespace std;

void mergeFiles( ifstream& fin1, ifstream& fin2, ofstream& fout );

void showFile( const char\* fname );

int main()

{

    ifstream fin1( "Source1.txt" );

    ifstream fin2( "Source2.txt" );

    ofstream fout( "Target.txt" );

    mergeFiles( fin1, fin2, fout );

    fout.close();

    showFile( "Source1.txt" );

    showFile( "Source2.txt" );

    showFile( "Target.txt" );

    return 0;

}

void mergeFiles( ifstream& fin1, ifstream& fin2, ofstream& fout )

{

    int i1, i2, countIn = 0, countOut = 0;

    if( fin1 >> i1 ) ++ countIn;

    if( fin1 )

    {

        if( fin2 >> i2 )

        {

            ++ countIn;

            while( fin1 && fin2 )

            {

                if( i1 <= i2 )

                {

                    fout << i1 << ' ';

                    ++countOut;

                    if( !(fin1 >> i1) )

                    {

                        fout << i2 << ' ';

                        ++countOut;

                    }

                    else ++countIn;

                }

                else

                {

                    fout << i2 << ' ';

                    ++countOut;

                    if( !(fin2 >> i2) )

                    {

                        fout << i1 << ' ';

                        ++countOut;

                    }

                    else ++countIn;

                }

            }

        }

        else

        {

            fout << i1 << ' ';

            ++countOut;

        }

    }

    while( fin1 >> i1 ) { fout << i1 << ' '; ++countIn; ++countOut; }

    while( fin2 >> i2 ) { fout << i2 << ' '; ++countIn; ++countOut; }

}

void showFile( const char\* fname )

{

    ifstream fin( fname );

    if( fin )

    {

        cout << "\nShowing numbers in file " << fname << " : ";

        int tmp;

        while( fin >> tmp ) cout << setw( 1 )<< tmp << ' ';

        fin.close();

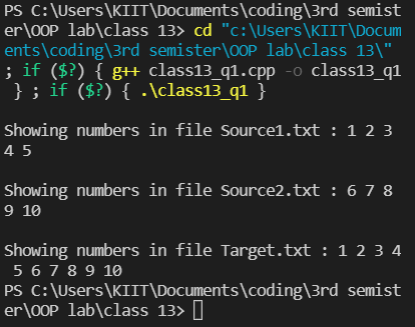
        cout << endl;

    }

    else cout << "\nThere was a problem opening file " << fname << endl;

}

**Output**

****

**Question 3**

#include <iostream>

#include <fstream>

#include <string>

#include <algorithm>

using namespace std;

class student

{

    public:

    string StudentName;

    int RollNumber;

    string Gender;

    int Weight;

    int Height;

};

bool compare(const student S1, const student S2)

{

    return S1.StudentName < S2.StudentName;

}

int main()

{

    ifstream x

    {

        "record.txt"

    };

    if (!x)

    {

        cerr << "The File could not be opened due to an error." << endl;

        exit(0);

    }

    student s[3];

    int i = 0;

    while (!x.eof())

    {

        x >> s[i].StudentName;

        x >> s[i].RollNumber;

        x >> s[i].Gender;

        x >> s[i].Height;

        x >> s[i].Weight;

        i++;

    }

    sort(s, s + 3, compare);

    for (int i = 0; i < 3; i++)

    {

        cout << s[i].StudentName << endl;

        cout << s[i].RollNumber << endl;

        cout << s[i].Gender << endl;

        cout << s[i].Height << endl;

        cout << s[i].Weight << endl;

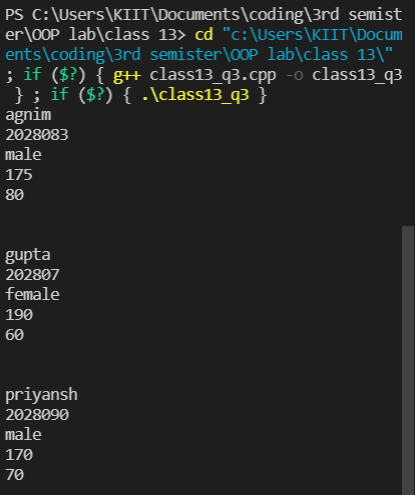
        cout << "\n" << endl;

    }

    return 0;

}

**Output**

****