1.  Write a program that reads a text file and **prints its odd lines** on the console.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

class Steve

{

public static int count = 0;

public static String[] steveTalk;

public static void Main()

{

using (StreamReader r = new StreamReader("Steve.txt"))

{

string line;

while ((line = r.ReadLine()) != null)

{

count++;

}

}

using (StreamReader sr = new StreamReader("Steve.txt"))

{

int i = 0;

steveTalk = new String[count];

String line;

while ((line = sr.ReadLine()) != null)

{

steveTalk[i] = line;

Console.WriteLine(steveTalk[i]);

i++;

}

}

while (true)

{

string input = Console.ReadLine();

int chooseLine = Convert.ToInt32(input);

try

{

Console.WriteLine(steveTalk[chooseLine]);

}

catch

{

Console.WriteLine("Error! Not a number or array index out of bounds");

}

Console.ReadLine();

}

}

}

2. Write a program that **joins two text files** and records the results in a third file.

public void positionCure(string afile,string bfile,string dfile)

{

string alphaFilePath = afile;

List<string> alphaFileContent = new List<string>();

using (FileStream fs = new FileStream(alphaFilePath, FileMode.Open))

using(StreamReader rdr = new StreamReader(fs))

{

while(!rdr.EndOfStream)

{

alphaFileContent.Add(rdr.ReadLine());

}

}

string betaFilePath = bfile;

StringBuilder sb = new StringBuilder();

using (FileStream fs = new FileStream(betaFilePath, FileMode.Open))

using (StreamReader rdr = new StreamReader(fs))

{

while(! rdr.EndOfStream)

{

string[] betaFileLine = rdr.ReadLine().Split(Convert.ToChar(","));

foreach (string alphaline in alphaFileContent)

{

string[] alphaFileLine = alphaline.Split(Convert.ToChar(","));

if (alphaFileLine[0].Equals(betaFileLine[0].ToString()))

{

sb.AppendLine(String.Format("{0}, {1}, {2}", betaFileLine[0], betaFileLine[1], alphaline.Substring(alphaline.IndexOf(Convert.ToChar(","))+1)));

}

}

}

}

using (FileStream fs = new FileStream(dfile, FileMode.Create))

using (StreamWriter writer = new StreamWriter(fs))

{

writer.Write(sb.ToString());

}

}

}

3.  Write a program that reads the contents of a text file and **inserts the line numbers** at the beginning of each line, then rewrites the file contents.

try

{

StreamReader sr = new StreamReader("C:\\Sample.txt");

line = sr.ReadLine();

while (line != null)

{

Console.WriteLine(line);

line = sr.ReadLine();

}

sr.Close();

Console.ReadLine();

}

catch(Exception e)

{

Console.WriteLine("Exception: " + e.Message);

}

finally

{

Console.WriteLine("Executing finally block.");

}

4. Write a program that **compares two text files** with the same number of rows line by line and prints the number of equal and the number of different lines.

private bool FileCompare(string file1, string file2)

{

int file1byte;

int file2byte;

FileStream fs1;

FileStream fs2;

if (file1 == file2)

{

return true;

}

fs1 = new FileStream(file1, FileMode.Open);

fs2 = new FileStream(file2, FileMode.Open);

if (fs1.Length != fs2.Length)

{

fs1.Close();

fs2.Close();

return false;

}

do

{

// Read one byte from each file.

file1byte = fs1.ReadByte();

file2byte = fs2.ReadByte();

}

while ((file1byte == file2byte) && (file1byte != -1));

fs1.Close();

fs2.Close();

return ((file1byte - file2byte) == 0);

}

5.   Write a program that **reads a list of names** from a text file, arranges them in **alphabetical order**, and writes them to another file. The lines are written one per row.

List<string> list = new List<string>();

using (StreamReader reader = new StreamReader("Documents/inputName.txt"))

{

string line;

while ((line = reader.ReadLine()) != null)

{

var parts = line.Split(',');

Array.Reverse(parts);

list.Add(string.Join(" ", parts));

}

}

list.Sort(new NameComparer());

using (StreamWriter writer = new StreamWriter("outputfile.txt"))

{

foreach(var line in list)

writer.WriteLine(line);

}