CSC180 Assignment #08

**#1** Exercise 7.12 (textbook 85-86)  
1) The Captain Crunch decoder ring works by taking each letter in a string and adding 13 to it. For example, 'a' becomes 'n' and 'b' becomes 'o'. The letters \wrap around" at the end, so 'z' becomes 'm'. Write a method that takes a string and that returns a new string containing the encoded version. You should assume that the string contains upper and lower case letters, and spaces, but no other punctuation. Lower case letters should be transformed into other lower case letters; upper into upper. You should not encode the spaces.

2) Generalize the Captain Crunch method so that instead of adding 13 to the letters, it  
   adds any given amount. Now you should be able to encode things by adding 13 and  
   decode them by adding -13. Test it.

using System;

using System.IO;

using System.Diagnostics;

namespace OMG

{

public class Program

{

public static string EncodeDecode(string s, int n)

{

string converted = "";

n = n % 26;

int index = 0;

while (index < s.Length)

{

char curChar = s[index];

if ('a' <= curChar && curChar <= 'z')

{

int convertedChar = curChar + n;

if (convertedChar > 'z')

{

convertedChar = (convertedChar - 'z') + 'a' - 1;

}

else if (convertedChar < 'a')

{

convertedChar = 'z' - ('a' - convertedChar) + 1;

}

converted += (char)convertedChar;

}

else if ('A' <= curChar && curChar <= 'Z')

{

int convertedChar = curChar + n;

if (convertedChar > 'Z')

{

convertedChar = (convertedChar - 'Z') + 'A' - 1;

}

else if (convertedChar < 'A')

{

convertedChar = 'Z' - ('A' - convertedChar) + 1;

}

converted += (char)convertedChar;

}

else

{

converted += (char)curChar;

}

index++;

}

return converted;

}

public static void Main(string[] args)

{

Console.Write("Enter a word to code/decode:");

string s = Console.ReadLine();

Console.Write("Enter a positive number to Encode or negative to Decode:");

int n = int.Parse(Console.ReadLine());

Console.WriteLine(EncodeDecode(s, n));

}

}

}

**#2** A word is said to be a "doubloon" if every letter that appears in the word appears exactly twice.

For example, the following are all the doubloons:

Abba, Anna, appall, appearer, appeases, arraigning, beriberi, bilabial, boob,

Caucasus, coco, Dada, deed, Emmett, Hannah, horseshoer, intestines, Isis,

mama, Mimi, murmur, noon, Otto, papa, peep, reappear, redder, sees, Shanghaiings, Toto

Write a method called IsDoubloon that returns true if the given word is a doubloon and false otherwise. (hint: use nested loops to count the number of letters)

using System;

namespace Doubloon

{

public class Program

{

static void Main(string[] args)

{

Console.WriteLine(IsDoubloon("abba"));

}

public static bool IsDoubloon(string s)

{

int[] hist = new int[26];

for (int i = 0; i < s.Length; i++)

{

char curChar = s[i];

if ('a' <= curChar && curChar <= 'z')

{

hist[curChar - 'a']++;

}

}

for (int i = 0; i < hist.Length; i++)

{

if (hist[i] != 0 && hist[i] != 2)

{

return false;

}

}

return true;

}

}

}

**#3** – Refer to the sample program below and answer the following questions:  
1) Explain Method01 and insert necessary comments to the source code.

Method 1 writes random numbers within a given range to the text file, numbers.txt.

2) Explain Method02 and insert necessary comments to the source code.

Method 2 sorts the array of integers in ascending order, starting at the position minimum.

3) What does code segment (lines 22-27) accomplish? (note that the following is Line 22: string[] lines = File.ReadAllLines(fileName);)

Lines 22-27 reads the lines from numbers.txt as an array of strings. The next line counts the lines and reads the string array then converts them to integer arrays.

/\*

\*CSC180 Week 8 Programmming Assignment - Problem #3

\* https://docs.microsoft.com/en-us/dotnet/standard/io/how-to-write-text-to-a-file

\* https://dotnetcoretutorials.com/2020/05/10/basic-sorting-algorithms-in-c/

\* https://medium.com/engineering-hub/https-medium-com-engineering-hub-sorting-algorithms-in-csharp-and-java-4615f6f87696

\*/

using System;

using System.IO;

using System.Diagnostics;

namespace CSC205Week08

{

class Program

{

static void Main(string[] args)

{

string fileName = "numbers.txt"; //sets filename to numbers.txt

Stopwatch stopwatch = new Stopwatch(); //creates a new stopwatch object

Method01(fileName, 1000, 1, 1001); //uses Method 1 with numbers.txt to create a random array of 1000 integers between 1 and 1001

string[] lines = File.ReadAllLines(fileName);//reads numbers.txt as a string array per line

int[] values = new int[lines.Length];//sets the integer array values as the lines from the previous string array

for (int i = 0; i < values.Length; i++)//increments the integer array

{

values[i] = Convert.ToInt32(lines[i]);//sets the integer array values to the integer array lines

}

stopwatch.Start();//starts the stopwatch

Console.WriteLine("starting ... ");//prints starting to the console

Method02(values);//sorts the values from Method 1

Console.WriteLine("done! ... ");//prints done

stopwatch.Stop();//stops the stopwatch

Console.WriteLine("time measured: {0} ms", stopwatch.ElapsedMilliseconds);//prints the stopwatch time elapsed during the execution of the previous 5 lines, to the console

foreach (int value in values)

Console.Write(value + " ");//writes the sorted random value and a space to the console

Console.WriteLine();//moves the cursor to the next line

}

static void Method01(string fileName, int total, int lowerRange, int upperRange)//creates random values using given parameters and writes the numbers.txt

{

using (var writer = new StreamWriter(fileName))

{

Random r = new Random();

int number = 0;

{

for (int i = 1; i < total; i++)

{

number = r.Next(lowerRange, upperRange);

writer.WriteLine(number);

}

}

}

}

static void Method02(int[] arr)//sorts the array of values

{

for (int start = 0; start < arr.Length - 1; start++)

{

int posMin = start;

for (int i = start + 1; i < arr.Length; i++)

{

if (arr[i] < arr[posMin])

{

posMin = i;

}

}

int tmp = arr[start];

arr[start] = arr[posMin];

arr[posMin] = tmp;

}

}

}

}