**Assignment 2 C#**

Enze Zhang

**Knowledge**

1. When we don't change the string, we can use string. If we need to modify the string a lot, string builder is better. While working with the String class, every time you perform some operations on your string, you recreate the entire string in the memory over and over again, whereas StringBuilder allocates some buffer space in the memory and applies modifications into that buffer space. As the StringBuilder object is mutable, it provides better performance as compared to the String object when heavy string manipulations are involved. String operations use more memory as compared to StringBuilder because String creates intermediate garbage instances after each operation. String is in System namespace but StringBuilder is in System.Text namespace
2. The Array class is the base class for all the arrays in C#.
3. We can use Array.Sort() method to sort array.
4. We can use Length property to get the total number of elements in an array.
5. Yes, we can use object [] to create array with different types of elements.
6. System.Array.CopyTo() copies the elements from the original array to the destination array starting at the specified destination array index. System.Array.Clone() method makes a clone of the original array. It returns an exact length array.

**Practice Arrays**

1.

using System;

namespace Copying\_an\_Array

{

class Program

{

static void Main(string[] args)

{

int[] oriArr = new int[] {10,9,8,7,6,5,4,3,2,1};

int[] newArr = new int[oriArr.Length];

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

{

newArr[i] = oriArr[i];

}

Console.WriteLine("Original Array:");

foreach(var j in oriArr)

{

Console.Write($"{j}, ");

}

Console.WriteLine("\nNew Array:");

foreach(var k in newArr)

{

Console.Write($"{k}, ");

}

}

}

}

2.

using System;

using System.Collections.Generic;

namespace ToPlayList

{

class ToPlayList

{

static void Main(string[] args)

{

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

Console.WriteLine("This is a play list. You can add, remove or clear games.");

while (true)

{

Console.WriteLine("Enter command (+ game, - game, or -- to clear)):");

string cmd = Console.ReadLine();

if (cmd == "--")

{

Plist.Clear();

Console.WriteLine("Play list cleared.");

}

else if (cmd.Substring(0,1) == "+"){

Plist.Add(cmd.Substring(2));

Console.WriteLine($"Game {cmd.Substring(2)} added.");

}

else if (cmd.Substring(0,1) == "-")

{

Plist.Remove($"{cmd.Substring(2)}");

Console.WriteLine($"Game {cmd.Substring(2)} removed.");

}

else

{

Console.WriteLine("Invalid Command");

}

Console.Write($"Current play list: ");

foreach(string i in Plist)

{

Console.Write($"{i}, ");

}

Console.WriteLine("");

}

}

}

}

3.

using System;

using System.Collections.Generic;

namespace Prime\_Numbers

{

class Program

{

static int[] FindPrimesInRange(int startNum,int endNum)

{

if(startNum < 2)

{

startNum = 2;

}

List<int> AList = new List<int>();

bool Pchecker = true;

for(int i = startNum; i <= endNum; i++)

{

for(int j = 2; j < i; j++)

{

if (i % j == 0)

{

Pchecker = false;

break;

}

}

if(Pchecker == true)

{

AList.Add(i);

}

Pchecker = true;

}

return AList.ToArray();

}

static void Main(string[] args)

{

Console.WriteLine("This is a prime number printer.");

Console.WriteLine("Please enter the starting number:");

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

Console.WriteLine("Please enter the ending number:");

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

int[] AnswerArr = FindPrimesInRange(st, ed);

Console.WriteLine($"The prime list will be:");

foreach(int k in AnswerArr)

{

Console.Write($"{k}, ");

}

}

}

}

4.

using System;

using System.Linq;

using System.Collections.Generic;

namespace Rotate\_Array

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Please enter an array seperated by space : ");

List<string> MyList = Console.ReadLine().Split(' ').ToList();

List<int> NumList = MyList.Select(int.Parse).ToList();

int n = NumList.Count;

Console.WriteLine("Please enter how many times you want to rotate:");

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

List<int> SumList = new List<int>();

for (int m = 0; m < n; m++)

{

SumList.Add(0);

}

for(int r = 1; r <= k; r++)

{

for(int i = 0; i < n; i++)

{

SumList[i] += NumList[(n+(i-r))%n];

}

}

Console.WriteLine("The sum array will be:");

foreach(int i in SumList)

{

Console.Write($"{i} ");

}

}

}

}

5.

using System;

using System.Linq;

using System.Collections.Generic;

namespace Longest\_Sequence

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Please input an array seperated by space:");

List<string> MyList = Console.ReadLine().Split(' ').ToList();

List<int> NumList = MyList.Select(int.Parse).ToList();

Dictionary<int, int> MyDic = new Dictionary<int, int>();

int curMax = 0;

int MaxVal = NumList[0];

int curVal = NumList[0];

int curCnt = 0;

for(int i = 0; i < NumList.Count; i++)

{

if (NumList[i] == curVal)

{

curCnt++;

}

else

{

if (curCnt > curMax)

{

curMax = curCnt;

MaxVal = curVal;

}

curVal = NumList[i];

curCnt = 1;

}

}

if (curCnt > curMax)

{

curMax = curCnt;

MaxVal = curVal;

}

Console.WriteLine("The longest sequence of equal elements is:");

for(int k = 0; k < curMax; k++)

{

Console.Write($"{MaxVal} ");

}

}

}

}

7.

using System;

using System.Linq;

using System.Collections.Generic;

namespace Most\_Frequent\_Number

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Please input an array seperated by space:");

List<string> MyList = Console.ReadLine().Split(' ').ToList();

List<int> NumList = MyList.Select(int.Parse).ToList();

Dictionary<int, int> MyDic = new Dictionary<int, int>();

foreach(int i in NumList)

{

if (MyDic.ContainsKey(i))

{

MyDic[i] += 1;

}

else

{

MyDic[i] = 1;

}

}

int curMax = 0;

List<int> AnsList = new List<int>();

foreach(KeyValuePair<int,int> k in MyDic)

{

if (k.Value > curMax)

{

curMax = k.Value;

AnsList.Clear();

AnsList.Add(k.Key);

}

else if (k.Value == curMax)

{

AnsList.Add(k.Key);

}

}

int LeftMost = AnsList[0];

AnsList.Sort();

if (AnsList.Count == 1)

{

Console.WriteLine($"The number {AnsList[0]} is the most frequent (occurs {curMax} times)");

}

else if (AnsList.Count > 1)

{

Console.Write("The numbers ");

if(AnsList.Count > 2)

{

for (int j = 0; j < AnsList.Count - 2; j++)

{

Console.Write($"{AnsList[j]}, ");

}

}

Console.Write($"{AnsList[AnsList.Count-2]} and {AnsList[AnsList.Count-1]} have the same maximal frequence (each occurs {curMax} times). ");

Console.Write($"The leftmost if them is {LeftMost}.\n");

}

}

}

}

**Practice Strings**

1.

using System;

namespace Reverse\_String

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Please input the string to be reversed: ");

string str = Console.ReadLine();

// First Way

char[] chr = str.ToCharArray();

Array.Reverse(chr);

Console.WriteLine("First method:");

foreach(char c in chr)

{

Console.Write($"{c}");

}

//Second Way

Console.WriteLine("\nSecond method:");

for(int i = str.Length-1; i >= 0; i--)

{

Console.Write(str[i]);

}

}

}

}

2.

using System;

using System.Linq;

using System.Collections.Generic;

using System.Text.RegularExpressions;

namespace Reverse\_Words\_in\_Sentence

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Please enter the text: ");

string ipt = Console.ReadLine();

List<string> WordList = ipt.Split(new Char[] { '.', ',', ':', ';', '=', '(', ')', '&', '[', ']', '"', '\'', '\\', '/', '!', '?', ' ' }, StringSplitOptions.RemoveEmptyEntries).ToList();

WordList.Reverse();

List<string> PList = new List<string> { ".", ",", ":", ";", "=", "(", ")", "&", "[", "]", "\"", "\'", "\\", "/", "!", "?", " " };

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

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

int curLen = 0;

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

{

if (PList.Contains(Char.ToString(ipt[i])))

{

curLen++;

}

else

{

if (curLen > 0)

{

TList.Add(ipt.Substring(i-curLen,curLen));

curLen = 0;

}

}

}

if (curLen > 0)

{

TList.Add(ipt.Substring(ipt.Length - curLen, curLen));

curLen = 0;

}

for (int j = 0; j < WordList.Count; j++)

{

Console.Write($"{WordList[j]}");

Console.Write($"{TList[j]}");

}

}

}

}

3.

using System;

using System.Linq;

using System.Collections.Generic;

namespace Extract\_Palindromes

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Please enter the text: ");

List<string> MyList = Console.ReadLine().Split(new Char[] { '.',',',':',';','=', '(', ')', '&', '[', ']' ,'"', '\'', '\\','/','!','?',' '}, StringSplitOptions.RemoveEmptyEntries).ToList();

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

foreach(string s in MyList)

{

int l, r;

bool Pckr = true;

if(s.Length % 2 == 1)

{

l = s.Length / 2;

r = s.Length / 2;

}

else

{

l = s.Length / 2 - 1;

r = s.Length / 2;

}

while(l >= 0)

{

if (s[l] != s[r])

{

Pckr = false;

break;

}

l--;

r++;

}

if(Pckr == true)

{

if (AnsList.Contains(s)==false)

{

AnsList.Add(s);

}

}

}

AnsList.Sort();

var ans = String.Join(", ", AnsList.ToArray());

Console.WriteLine($"Palindromes are the followings: \n{ans}");

}

}

}

4.

using System;

using System.Linq;

using System.Collections.Generic;

namespace Parse\_URL

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Please enter the URL:");

string ipt = Console.ReadLine();

List<string> MyList = ipt.Split(new Char[] { ':', '/' }, StringSplitOptions.RemoveEmptyEntries).ToList();

List<string> AnsList = new List<string>() {"","",""};

if (ipt.Contains("://"))

{

AnsList[0] = MyList[0];

AnsList[1] = MyList[1];

if (MyList.Count >= 3)

{

AnsList[2] = ipt.Substring(AnsList[0].Length + AnsList[1].Length+4);

}

}

else

{

AnsList[1] = MyList[0];

if(MyList.Count >= 2)

{

AnsList[2] = ipt.Substring(AnsList[1].Length + 1);

}

}

Console.WriteLine($"[protocol] = \"{AnsList[0]}\"");

Console.WriteLine($"[server] = \"{AnsList[1]}\"");

Console.WriteLine($"[resource] = \"{AnsList[2]}\"");

}

}

}