**02 Arrays and Strings**

Test your Knowledge

1. When to use String vs. StringBuilder in C# ?

String is to declare string value and it is immutable.

StringBuilder is to change the value of a string object.

2. What is the base class for all arrays in C#?

C# provides an array class to deal with array related operations. It provides methods for creating, manipulating, searching, and sorting elements of an array. This class works as the base class for all arrays in the .Net programming environment.

3. How do you sort an array in C#?

Sort the array using the Sort() method. Array. Sort(list); You can try to run the following code to to sort an array in ascending order

4. What property of an array object can be used to get the total number of elements in

an array?

Length Property is used to get the total number of elements in all the dimensions of the Array.

5. Can you store multiple data types in System.Array?

We cannot store multiple data types to array. we can store this to arraylist.

6. What’s the difference between the System.Array.CopyTo() and System.Array.Clone()?

Clone() method returns a new array object containing all the elements in the original array. This method creates a copy of an array as an object, therefore needs to be cast to the actual array type before it can be used to do very much. The clone is of the same type as the original array.

The copyto() method copies the elements into another existing array. It copies the elements of one array to another pre-existing array starting from a given index(usually 0).

**Practice Array**

1.

using System;

namespace kim {

class days {

public static void Main()

{ string[] Days;

Days = new string[] {"Sun", "Mon", "Tue", "Wed",

"Thu", "Fri", "Sat",”weekend”,”Holiday”,”anniversary”};

foreach(string day in Days)

Console.Write(day + " ");

Console.Write("\nTotal Number of Elements: ");

Console.Write(weekDays.Length);

}

}

}

3. using System;

class HelloWorld

{

static int[] FindPrimesInRange(int strtRange, int endRange)

{

int[] primes = new int[endRange-strtRange+1];

int i, j;

int arrayIndex = 0;

for (i = strtRange; i <= endRange; i++)

{

if (i == 1 || i == 0)

{

continue;

}

int flag = 1;

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

{

if (i % j == 0)

{

flag = 0;

break;

}

}

if (flag == 1)

{

primes[arrayIndex++] = i;

}

}

int[] primesNumber=new int[arrayIndex];

for ( i = 0; i<arrayIndex; i++ ) {

primesNumber[i] = primes[i];

}

return primesNumber;

}

static void Main ()

{

Console. WriteLine ("Enter starting range and ending range of prime numbers");

int strtRange = Convert.ToInt32 (Console.ReadLine ());

int endRange = Convert.ToInt32 (Console.ReadLine ());

int[] primes=FindPrimesInRange(strtRange,endRange);

Console.WriteLine(string.Join(" ", primes));

}

}

5. using System;

using System.Collections.Generic;

class GFG{

static int findLongestConseqSubseq(int[] arr,

int n)

{

// Sort the array

Array.Sort(arr);

int ans = 0, count = 0;

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

v.Add(10);

// Insert repeated elements

// only once in the vector

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

{

if (arr[i] != arr[i - 1])

v.Add(arr[i]);

}

// Find the maximum length

// by traversing the array

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

{

// Check if the current element is

// equal to previous element +1

if (i > 0 && v[i] == v[i - 1] + 1)

count++;

else

count = 1;

// Update the maximum

ans = Math.Max(ans, count);

}

return ans;

}

// Driver code

static void Main()

{

int[] arr = { 1, 9, 3, 10, 4, 20, 2 };

int n = arr.Length;

Console.WriteLine("Length of the Longest " +

"contiguous subsequence is " +

findLongestConseqSubseq(arr, n));

}

}

**Practice String**

1. ReverseStringWithoutInbuiltMethod(string stringInput){

char[] stringArray = stringInput.ToCharArray();

string reverse = String.Empty;

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

{reverse+=stringArray[i];}

Console.WriteLine(reverse);

Console.ReadLIne();

1. var resultstring = new

string(stringInput.ToCharArray().Reverse().ToArray());

console.WriteLine(resultstring);

Console.ReadLine();

}

1. static void Main(string[] args)

{

string \_inputstr, \_reversestr = string.Empty;

Console.Write("Enter a string : ");

\_inputstr = Console.ReadLine();

if (\_inputstr != null)

{

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

{

\_reversestr += \_inputstr[i].ToString();

}

if (\_reversestr == \_inputstr)

{

Console.WriteLine("String is Palindrome Input = {0} and Output= {1}", \_inputstr, \_reversestr);

}

else

{

Console.WriteLine("String is not Palindrome Input = {0} and Output= {1}", \_inputstr, \_reversestr);

}

} Console.ReadLine(); }