Q.1: How to reverse a string?

**Ans.:**The user will input a string and the method should return the reverse of that string

* input: hello, output: olleh
* input: hello world, output: dlrow olleh

**internal** **static** **void** ReverseString(**string** str)

{

**char**[] charArray = str.ToCharArray();

**for** (**int** i = 0, j = str.Length - 1; i < j; i++, j--)

{

charArray[i] = str[j];

charArray[j] = str[i];

}

**string** reversedstring = new **string**(charArray);

Console.WriteLine(reversedstring);

}

**Q.2: How to find if the given string is a palindrome or not?**

**Ans.:**The user will input a string and we need to print “*Palindrome*” or “*Not Palindrome*” based on whether the input string is a palindrome or not.

* input: madam, output: Palindrome
* input: step on no pets, output: Palindrome
* input: book, output: Not Palindrome

if we pass an integer as a string parameter then also this method will give the correct output

* input: 1221, output: Palindrome

**internal** **static** **void** chkPalindrome(**string** str)

{

**bool** flag = **false**;

**for** (**int** i = 0, j = str.Length - 1; i < str.Length / 2; i++, j--)

{

**if** (str[i] != str[j])

{

flag = **false**;

**break**;

}

**else**

flag = **true**;

}

**if** (flag)

{

Console.WriteLine("Palindrome");

}

**else**

Console.WriteLine("Not Palindrome");

}

**Q.3: How to reverse the order of words in a given string?**

**Ans.:**The user will input a sentence and we need to reverse the sequence of words in the sentence.

* input: Welcome to Csharp corner, output: corner Csharp to Welcome

**internal** **static** **void** ReverseWordOrder(**string** str)

{

**int** i;

StringBuilder reverseSentence = new StringBuilder();

**int** Start = str.Length - 1;

**int** End = str.Length - 1;

**while** (Start > 0)

{

**if** (str[Start] == ' ')

{

i = Start + 1;

**while** (i <= End)

{

reverseSentence.Append(str[i]);

i++;

}

reverseSentence.Append(' ');

End = Start - 1;

}

Start--;

}

**for** (i = 0; i <= End; i++)

{

reverseSentence.Append(str[i]);

}

Console.WriteLine(reverseSentence.ToString());

}

**Q.4: How to reverse each word in a given string?**

**Ans.:**The user will input a sentence and we need to reverse each word individually without changing its position in the sentence.

* input: Welcome to Csharp corner, output: emocleW ot prahsC renroc

**internal** **static** **void** ReverseWords (**string** str)

{

StringBuilder output = new StringBuilder ();

List<**char**> charlist = new List<**char**> ();

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

{

**if** (str[i] == ' ' || i == str.Length - 1)

{

**if** (i == str.Length - 1)

charlist.Add (str[i]);

**for** (**int** j = charlist.Count - 1; j >= 0; j--)

output.Append(charlist[j]);

output.Append(' ');

charlist = new List<**char**>();

}

**else**

charlist.Add(str[i]);

}

Console.WriteLine(output.ToString());

}

**Q.5: How to count the occurrence of each character in a string?**

**Ans.:**The user will input a string and we need to find the count of each character of the string and display it on console. We won’t be counting space character.

* input: hello world;

output:

h – 1

e – 1

l – 3

o – 2

w – 1

r – 1

d – 1

**internal** **static** **void** Countcharacter (**string** str)

{

Dictionary<**char**, **int**> characterCount = new Dictionary<**char**, **int**>();

**foreach** (**var** character in str)

{

**if** (character != ' ')

{

**if** (!characterCount.ContainsKey(character))

{

characterCount.Add (character, 1);

}

**else**

{

characterCount[character]++;

}

}

}

**foreach** (**var** character in characterCount)

{

Console.WriteLine("{0} - {1}", character.Key , character.Value);

}

}

**Q.6: How to remove duplicate characters from a string?**

**Ans.:**The user will input a string and the method should remove multiple occurrences of characters in the string

* input: csharpcorner, output: csharpone

**internal** **static** **void** removeduplicate (**string** str)

{

**string** result = **string**.Empty;

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

{

**if** (!result.Contains(str[i]))

{

result += str[i];

}

}

Console.WriteLine(result);

}

**Q.7: How to find all possible substring of a given string?**

**Ans.:**This is a very frequent interview question. Here we need to form all the possible substrings from input string, varying from length 1 to the input string length. The output will include the input string also.

* input: abcd , output : a ab abc abcd b bc bcd c cd d

**internal** **static** **void** findallsubstring(**string** str)

{

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

{

StringBuilder subString = new StringBuilder(str.Length - i);

**for** (**int** j = i; j < str.Length; ++j)

{

subString.Append(str[j]);

Console.Write(subString + " ");

}

}

}

**Q.8: How to perform Left circular rotation of an array?**

**Ans.:**The user will input an integer array and the method should shift each element of input array to its Left by one position in circular fashion. The logic is to iterate loop from Length-1 to 0 and swap each element with last element.

* input: 1 2 3 4 5, output: 2 3 4 5 1

**internal** **static** **void** RotateLeft(**int**[] array)

{

**int** size = array.Length;

**int** temp;

**for** (**int** j = size - 1; j > 0; j--)

{

temp = array[size - 1];

array[array.Length - 1] = array[j - 1];

array[j - 1] = temp;

}

**foreach** (**int** num in array)

{

Console.Write(num + " ");

}

}

**Q.9: How to perform Right circular rotation of an array?**

**Ans:** The user will input an integer array and the method should shift each element of input array to its Right by one position in circular fashion. The logic is to iterate loop from 0 to Length-1 and swap each element with first element

* input: 1 2 3 4 5, output: 5 1 2 3 4

**internal** **static** **void** RotateRight(**int**[] array)

{

**int** size = array.Length;

**int** temp;

**for** (**int** j = 0; j < size - 1; j++)

{

temp = array[0];

array[0] = array[j + 1];

array[j + 1] = temp;

}

**foreach** (**int** num in array)

{

Console.Write(num + " ");

}

}