

1. Write C# Sharp program to demonstrate that the Compare(String, String, Boolean) method is equivalent to using ToUpper or ToLower when comparing strings.

Expected Output :

Comparing 'QRS' and 'qrs':

The Strings are equal when capitalized? true

The Strings are equal when case is ignored? true

2. Write a C# Sharp program to demonstrate how culture affects comparisons.

Note : In Czech – Czech Republic culture, "ch" is a single character that is greater than "d". However, in English - United States culture, "ch" consists of two characters, and "c" is less than "d".

Expected Output :

For en-US: change < dollar

For cs-CZ: change > dollar

3. Write a C# Sharp program to compare two strings in following three different ways produce three different results.

a. using linguistic comparison for the en-US culture;

b. using linguistic case-sensitive comparison for the en-US culture;

c. using an ordinal comparison. It illustrates how the three methods of comparison

Expected Output :

'sister' comes before 'Sister'.

'sister' is the same as 'Sister'.

'sister' comes after 'Sister'.

4. Write a C# Sharp program to compare three versions of the letter "I". The results are affected by culture choice, whether the case is ignored, and whether an ordinal comparison is performed.

Expected Output :

Compare three versions of the letter I using different values of StringComparison.

The current culture is en-US.

StringComparison.CurrentCulture:

LATIN SMALL LETTER I (U+0069) is less than LATIN SMALL LETTER DOTLESS I (U+0131)

LATIN SMALL LETTER I (U+0069) is less than LATIN CAPITAL LETTER I (U+0049)

LATIN SMALL LETTER DOTLESS I (U+0131) is greater than LATIN CAPITAL LETTER I (U+0049)

5. Write a C# Sharp program to demonstrate that you compare ordinals and compare using different sort orders.

Expected Output :

```
CompareOrdinal("xyz"[1], "XYZ"[1]):
```

```
'y' is greater than 'Y'
```

```
Compare("xyz"[1], "XYZ"[1]):
```

```
'y' is less than 'Y'
```

6. Write a C# Sharp program to perform an ordinal comparison of two strings that only differ in case.

Expected Output :

Compare the numeric values of the corresponding Char objects in each string.

```
str1 = 'JAVA', str2 = 'python'
```

```
String 'JAVA' is less than String 'python'.
```

7. Write a C# Sharp program to compare a given string with a set of strings.

Expected Output :

```
Bad argument: TestClass (type TestClass)
```

```
Comparing 'some text' with 'TestClass': -1
```

```
Bad argument: 123 (type Int32)
```

```
Comparing 'some text' with '123': 1
```

```
Comparing 'some text' with 'some text': 0
```

```
Comparing 'some text' with 'Some Text': -1
```

8. Write a C# Sharp program to compare the current string instance with another string.

Expected Output :

The strings occur in the same position in the sort order.

The first string follows the second in the sort order.

The first string precedes the second in the sort order.

The first string precedes the second in the sort order.

The first string follows the second in the sort order.

9. Write a C# Sharp program to concatenate three objects, objects with a variable and a 3-element object array.

Expected Output :

Search for the target string "Å" in the string "abcЙ□".

Using the English (United Kingdom) - "en-GB" culture:

Case sensitive:

The string to search ends with the target string: False

.....

10. Write a C# Sharp program to concatenate a list of variable parameters.

Expected Output :

abcd --> cadb

efgh --> hgef

ijkl --> lkij

mnpq --> nmpo

11. Write a C# Sharp program to concatenate three strings and display the result.

Expected Output :

Don't count your chickens, before the eggs, have hatched.

12. Write a C# Sharp program to concatenate string array values.

Expected Output :

hello welcome to C# Sharp create Windows client applications

hello applicationsC# Sharp client create to welcome Windows

13. Write a C# Sharp program to determine whether the string "birds" is a substring of a familiar string.

Note : Quotation 'two birds with one stone'.

Expected Output :

'birds' is in the string 'Kill two birds with one stone': True

'birds begins at character position 10

14. Write a C# Sharp program to create two string objects with different values. When it calls the Copy method to assign the first value to the second string, the output indicates that the strings represent different object references. However, their values are now equal. On the other hand, when the first string is assigned to the second string, the two strings have identical values. This is because they represent the same object reference.

Expected Output :

```
s1 = 'JAVA'
s2 = 'Python'
```

15. Write a C# Sharp program to indicate whether each string in an array ends with a period (".").

Expected Output :

```
'Actions speak louder than words' ends in a period: False
'Hello!' ends in a period: False
'Python.' ends in a period: True
'PHP.' ends in a period: True
'random' ends in a period: False
```

16. Write C# Sharp program to check whether a string occurs at the end of another string.

Expected Output :

Search for the target string "Å" in the string "abcЙ□".

Using the English (United Kingdom) - "en-GB" culture:

Case sensitive:

The string to search ends with the target string: False

Case insensitive:

The string to search ends with the target string: False

Using the English (Australia) - "en-AU" culture:

Case sensitive:

The string to search ends with the target string: False

Case insensitive:

The string to search ends with the target string: False

17. Write a C# Sharp program to determine whether a string ends with a particular substring.

Note : The results are affected by the choice of culture, whether case is ignored, and whether an ordinal comparison is performed.

Expected Output :

Determine whether a string ends with another string, using different values of StringComparison.

The current culture is en-US.

StringComparison.CurrentCulture:

"xyzPQR" ends with "PQR".

"xyzPQR" ends with "PQR".

.....

18. Write a C# Sharp program to get the longest Palindromic substring from a given string.

From Wikipedia:

In computer science, the longest palindromic substring or longest symmetric factor problem is the problem of finding a maximum-length contiguous substring of a given string that is also a palindrome. For example, the longest palindromic substring of "bananas" is "anana". The longest palindromic substring is not guaranteed to be unique; for example, in the string "abracadabra", there is no palindromic substring with length greater than three, but there are two palindromic substrings with length three, namely, "aca" and "ada".

Expected Output :

Original String: aaaaaabbbbccc

Length of the longest substring without repeating characters of the said string:

aaaaa

Original String: BDEFGAABEF

Length of the longest substring without repeating characters of the said string:

AA

Original String: Python

Length of the longest substring without repeating characters of the said string:

P

Original String: Java

Length of the longest substring without repeating characters of the said string:

av

19. Write a C# Sharp program to reverse a given string in uppercase.

Expected Output :

Original string: php
Said string in uppercase: PHP
Original string: java
Said string in uppercase: AVAJ
Original string: abcd
Said string in uppercase: DCBA

20. Write a C# Sharp program to remove duplicate characters from a given string.

Expected Output :

Original String: aaaaaabbbbccc
After removing duplicates characters from the said string:
abc
Original String: Python
After removing duplicates characters from the said string:
Python
Original String: Java
After removing duplicates characters from the said string:
Jav

21. Write a C# Sharp program to find the length of the longest substring without repeating characters in a given string.

Expected Output :

Original String: aaaaaabbbbccc
Length of the longest substring without repeating characters of the said string:
2
Original String: BDEFGAABEF
Length of the longest substring without repeating characters of the said string:
6
Original String: Python
Length of the longest substring without repeating characters of the said string:
6
Original String: Java
Length of the longest substring without repeating characters of the said string:
3

22. Write a C# Sharp program to reverse the case (upper->lower, lower->upper) of all the characters in a given string.

Expected Output :

Original string: PHP

After reversing the case of all characters of the said string: php

Original string: JavaScript

After reversing the case of all characters of the said string: jAVAsCRIPT

Original string: Python 3.0

After reversing the case of all characters of the said string: pYTHON 3.0

23. Write a C# Sharp program to find the central character(s) in a given string. Return the middle character if the string length is odd and return two middle characters if the string length is even.

Expected Output :

Original string: Python

Middle character(s) of the said string: th

Original string: PHP

Middle character(s) of the said string: H

Original string: C#

Middle character(s) of the said string: C#

24. Write a C# Sharp program to find the maximum and minimum number from a given string of numbers separated by a single space.

Expected Output :

Original string of numbers: 3 4 8 9 0 2 1

Maximum and minimum number of the said string: 9, 0

Original string of numbers: -2 -1 0 4 10

Maximum and minimum number of the said string: 10, -2

25. Write a C# Sharp program to check whether a given string is an "isograms" or not. Return True or False.

From Wikipedia,

A heterogram (from hetero-, meaning 'different', + -gram, meaning 'written') is a word, phrase, or sentence in which no letter of the alphabet occurs more than once. The terms isogram and nonpattern word have also been used to mean the same thing.

Expected Output :

Original string: Python
Check the said string is an 'isograms' or not! True

Original string: JavaScript
Check the said string is an 'isograms' or not! False

Original string: PHP
Check the said string is an 'isograms' or not! False

Original string: C#
Check the said string is an 'isograms' or not! True

26. Write a C# Sharp program to convert the first character of each word in a given string to uppercase.

Expected Output :

Original string: python exercises
After converting the first character of each word of the said string:
Python Exercises
Original string: The quick brown Fox jumps over the little lazy Dog
After converting the first character of each word of the said string:
The Quick Brown Fox Jumps Over The Little Lazy Dog

27. Write a C# Sharp program to find the position of a specified word in a given string.

Sample Example:

Text: The quick brown fox jumps over the lazy dog.
Position: 1 2 3 4 5 6 7 8 9
Expected Output :

Original string: The quick brown fox jumps over the lazy dog.
Position of the word 'fox' in the said string: 4
Position of the word 'The' in the said string: 1
Position of the word 'lazy' in the said string: 8

28. Write a C# Sharp program to alternate the case of each letter in a given string. The first letter of this string must be uppercase.

Expected Output:

Original string: c# Exercises

After alternating the case of each letter of the said string:

C# ExErCiSeS

Original string: C# is used to develop web apps, desktop apps, mobile apps, games and much more.

After alternating the case of each letter of the said string:

C# Is UsEd To DeVeLoP wEb ApPs, dEsKtOp ApPs, mObIlE aPpS, GaMeS aNd MuCh MoRe.

29. Write a C# Sharp program to reverse all the words of a given string with even length.

Expected Output:

Original string: C# Exercises

Reverse all the words of the said string which have even length.:

#C Exercises

Original string: C# is used to develop web apps , desktop apps , mobile apps , games and much more.

Reverse all the words of the said string which have even length.:

#C si desu ot develop web sppa , desktop sppa , elibom sppa , games and hcum more.

30. Write a C# Sharp program to find the longest common ending between two given strings.

Expected Output:

Original strings: running ruminating

Common ending between said two strings:ing

Original strings: thisisatest testing123testing

Common ending between said two strings:

31. Write a C# Sharp program that takes a string and reverses the words of three or more lengths in a string. Return the updated string. As input characters, only letters and spaces are permitted.

Sample Data:

("The quick brown fox jumps over the lazy dog") -> "ehT kciuq nworb xof spmuj revo eht yzal god"

("Reverse the words of three or more") -> "esreveR eht sdrow of eerht or erom"

("ABcDef") -> "feDcBA"

32. Write a C# Sharp program to check if two given strings are anagrams or not.

Sample Data:

("wxyz", "zyxw") -> True
("pears", "spare") -> True
("stone", "tones") -> True
("cat", "rat") -> False

33. Write a C# Sharp program to reverse a positive integer and add the reversed number to the original number.

Sample Data:

(456) -> 456654
(-123) -> "Input a positive number."
(10000) -> 1000000001

34. Write a C# Sharp program to count the number of duplicate characters (case sensitive) including spaces in a given string. If there are no duplicates, return 0.

Sample Data:

("Red Green WHITE") -> 2
("ppqrrstuuu") -> 4
("dow jones industrial average") -> 9

35. Write a C# Sharp program to count a specific string (case sensitive) in given text.

36. Write a C# Sharp program that displays all the characters with their respective numbers from a string.

Sample Data:

("C Sharp") -> " 3 19 8 1 18 16"
("The quick brown fox jumps over the lazy dog.") -> " 20 8 5 17 21 9 3 11 2 18 15 23 14 6 15
24
10 21 13 16 19 15 22 5 18 20 8 5 12 1 26 25 4 15 7"
("HTML Tutorial") -> " 8 20 13 12 20 21 20 15 18 9 1 12 "

37. Write a C# Sharp program that takes a string that repeats a substring. Count the number of times the substring appears.

Sample Data:

("aaaaaa") -> 6

("abababab") -> 4

("abcdabcdabcd") -> 3

38. Write a C# Sharp program to find the longest abecedarian word in a given array of words.

Sample Data:

({"abc", "abcd", "abcdef", "pqrstuv"}) -> "pqrstuv"

({"abc", "abcd", "abcdef", "pqrs"}) -> "abcdef"

({}) -> "Empty array!"

39. Write a C# Sharp program to find the century of a given year.

Sample Data:

(1435) -> "15th century"

(1567) -> "16th century"

(1921) -> "20th century"

(2014) -> "21st century"

40. Write a C# Sharp program to calculate the highest frequency of the character(s) in the words of a given string.

Sample Data:

("aaaaaa") -> "a"

("Century of the said year") -> ""

("CPP Exercises") -> "s, e, p"

41. Write a C# Sharp program that converts ASCII characters to hexadecimal values.

Sample Data:

("Python") -> "50 79 74 68 6f 6e"

("Century of the year") -> "43 65 6e 74 75 72 79 20 6f 66 20 74 68 65 20 79 65 61 72"

("CPP Exercises") -> "43 50 50 20 45 78 65 72 63 69 73 65 73"

42. Write a C# Sharp program to check whether all characters in a given string are unique. If all the characters are distinct, return "There is no similarity between any of the characters!" otherwise if not unique, then:.

Display a message stating this.

Display the duplicated character.

It is only necessary to display the first non-unique character.

Display where "both" duplicated characters appears in the string.

All of the above messages can be incorporated into a single message.