MODULE 13
String Comparison with same lenghts

Introduction

String functions are used in computer programming languages to manipulate a string or query information about a string. Most computer programming languages that have a string data type will have some string functions. String functions are often implemented as properties and methods of string objects. In functional and list based languages a string is represented as a list (of characters), therefore all list-manipulation procedures could be considered string functions.

There are two ways to compare the given strings. The first method would be as in general like finding the lengths of the given strings and compare.

In python there is an in built function called '***len()*** ', used to find the length of the given string.

len() function

This function takes one string values as an argument and returns the number of characters in that string.

Note: length of a string = the number of characters in the string+ white spaces.

Worked out Examples:

```
>>> str = "hello world"
>>> length = len(str)
>>> print length
11
```

Note: Here total characters are (Hello+World) = 10 but there is a space character between these two words which is also counted as the length of the string.

Points to remember

1. White spaces are counted as the length of the string. Ex: 'space', 'tab'.
2. Special and Escape characters are not counted as the length of the string. Ex: '\', '\n', '\r'.

Comparison Strings

We hope you have got an idea about finding the length of the given strings. So when you want to compare two strings, you can find the lengths of the given strings and compare based on their lengths.

Steps to Follow

1. Read and store the first string.
2. Find the length of the first string. (Use len()).
3. Read and store the second string.
4. Follow step 2.
5. Compare the lengths. (Use ***if***)

Compare strings which have same lengths.

It is ok when you have two strings of different lengths. For instance str1 = “ramesh” has length of ‘6’ and str2= “ram” has length of ‘3’. So you can easily compare both. But when the two strings have same length such as str1 = “ravi” and str2 = “raju” where both strings have same length = ‘4’.

To solve above problem we need to compare the strings based on their characters instead of length of the string. Computer can compare integer values but how does computer compare characteristics. Solution is pretty simple; when computer is comparing characteristics it is actually comparing the “ASCII” values of those characters. Before we know the comparison of “ASCII” values let us know what it is?

What is ASCII?

“American Standard Code for Information Interchange ”. Pronounced *ask-ee*, ASCII is a code for representing English characters as numbers, with each letter assigned a number from 0 to 127. For example, the ASCII code for uppercase ‘*M’* is 77. Most computers use ASCII codes to represent text, which makes it possible to transfer data from one computer to another.

Using ***ord()*** method we can find the ASCII value of any character.

Character to ASCII value:

```
>>> ord ('a')
97
```

ASCII value to character:

```
>>> chr(66)
'B'
```

Comparing two characters:

```
>>> s1 = 'a'
>>> s2 = 'b'
>>> if s1<s2:
>>>     print "True"
>>> else:
>>>     print "False"

>>> True.
```

Notice that the program is printing the output based on the ASCII values only. We know that ASCII value of 'a' is 97 and ASCII value of 'b' is 98 so it will produce the out is "True".

It is easy to compare if you are given two characters to compare. Then for strings, which have a set of characters. In python there is an in built function called ***cmp(x,y)***.

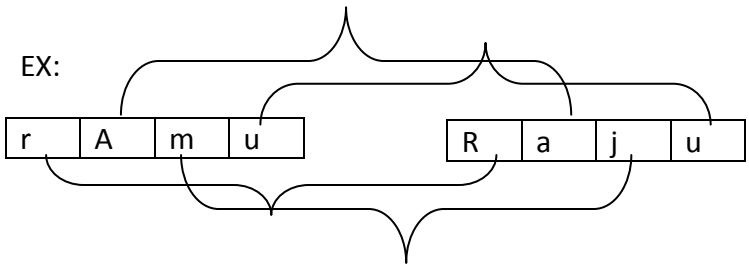
cmp() function

This function takes two strings as arguments and will compare two given strings based their ASCII values. This function takes two arguments as an input and starts comparing as follows.

Compares the two objects *x* and *y* and return an integer according to the outcome. The return value is negative (-1) if `x < y`, zero if `x == y` and strictly positive (+1) if `x > y`.

What goes in cmp() function

1. Checks the first character of the first string with the first character of the second string.
2. This will continue till the end of the string is encountered.



In above example the ASCII values of first two characters and the last character is same but the ASCII value of third character is different.

Worked out Example:

```
>>> s1 = "ravi"
>>> s2 = "RAVI"
>>> print cmp(s1,s2)
1
>>>
```

Note: Though both words are same but it checks the lower case and upper case of the words and then compare. ASCII of 'R' is 82 and ASCII of 'r' is 114.

Practice Problems:

1. What is the output when you run the following program?

```
>>> a=10
>>> b=12
>>> print cmp(a,b)
```

A. 1 B. -1 C. 0 D. Error

2. If you get a value 1 what is the condition for the two strings (S1,S2).

A. S1>S2 B. S1<S2 C.S1==S2 D. S1>=S2

3. What are the ASCII values for 'A' and 'z'?

A. 122, 65 B. 65,122 C. 97,90 D. 90,97

4. Write a program to print all ASCII values of lower case alphabets (a to z).

5. Take two string from the user and then use cmp function to compare.

Exercise Problem

Compare the given two strings which have same lengths.

1. "Acer", "dell".
2. "Hello World", "hello world".
3. "xyxabc", "abcxyx".
4. "abcabc", "ababab".
5. "xyxxyx", "xzyxyz".

Hints to follow

1. Take two string inputs and store them in two variables.
2. Use cmp() function to compare.