Problem Solving by Computer – Logical Thinking

Tech Capsule 4 – String based problems

In programming, String is a sequence of characters. Strings are used for manipulating non-numeric information.

Declaring a string?
In almost all programming language (let us consider C language as an example), a string is represented as an array of ***chars*** (characters) which ends with a null character (ASCII code is '\0'). The size of the array should be 1 more than the maximum expected length of the string to accommodate the null character.

For Example:

- *char name[25]*
- *char subject[10]*

Initializing Strings:
Strings can be initialized in the same way as we initialize arrays.

Example:

- **Using array notation:**
  - *char name[25]={'W','i','p','r','o','\0'}*

  Here even though the size of the sting is 25 we are initializing only the first 6 elements in the array.

- **Using a constant string:**
  - *char name[25]="Wipro"*

  This is same as initializing in array notation, but simpler to use. The last character by default will be the NULL character.

  In both the methods of initializing strings where we initialize the string at the time of declaring it, we can leave out the size of the array. This can be rewritten as below.

  char name[]={'W','i','p','r','o','\0'}

  char name[]="Wipro"

String Operations:

There are various operations which may need to be performed on strings. Few of these operations are listed below.

- Calculate the length of a given string
- Concatenate 2 given strings
- Reverse a given string
- Count the number of vowels in a given string

Let us look at one of these and build an algorithm.

Problem-

Design an algorithm to accept a string from the user and calculate its length.

Solution-

First step will be to declare a character array of some size (may be 50, assuming user will enter a string with size less than 50). Also identify other variables required.

Step 1 –

```
char inputstr[50]
int i=0, str_length=0
```

Step 2 –

Now we read the string from the user

```
print "Enter a string of less than 50 characters"
read inputstr
```

Step 3 –

Now to calculate the length of the given string we must navigate from the first character till the last character and for each character we increment a count variable (which is initially initialized to 0). The end of the string can identified as all strings end with a NULL '\0' character. The entire process of checking if a character is '\0' or not and if not increment the count is repetitive and can be done using any of the iterative constructs (while or for).

```
while inputstr[i] is not equal to '\0'
do
        Increment value of str_length by 1
        Increment value of I by 1
done
```

Step 4 –

print ***str_length***

Try by yourself

Activity 1:
Design an algorithm to accept a string from the user. Reverse the string and display it.

Activity 2:
Design an algorithm to accept a string from the user. Replace all vowels ('a', 'e', 'I', 'o', 'u') with 'z'. If there are no vowels in the string just print the original word with the message "No vowels present".

Activity 3:
Design an algorithm to accept 2 strings (string1 and string2) from the user. Concatenate both the strings into a third string "string3" and print the result.
Example –

string1 = "Hello"
string2 = "World"

string3 must be "HelloWorld"

Activity 4:
Design an algorithm to accept a string from the user. Count and print the number of times each character occurs in the given string.
Example –

input string = "malayalam"
Output must be –
m – 2
a – 4
l – 2
y - 1

Activity 5:
Design an algorithm which accepts 2 strings from the user and performs the following operation on it.
Example –

string1 = "Hello"
string2 = "World"
string3 must be "HellodlroW"

Both strings must be concatenated but after reversing the second string (as shown in string3 above).