Birla Institute of Technology & Science, Pilani
Computer Programming (CF F111)
Second Semester 2014-2015
Lab-10

Objectives

1. Strings
2. Exercises

Strings in C: A string in C is defined as an array of characters terminated by a NULL (`'\0'`) character. Strings can be read by using the inbuilt function **`gets(string_variable)`** and can be displayed by the function **`puts(string_variable)`.** To store a string of N characters N+1 locations are required due to default NULL character at the end of string. Usually in all string manipulation problems strings are read character by character in a loop until '\0' occurs. The ASCII code of the escape sequence '\0' is 0.

There are some inbuilt string manipulation functions available and defined in the header file **string.h** like strcmp(s1,s2), strlen(s1), strcpy(s1,s2). Here s1 and s2 are string variables declared as character arrays.

1. The following C program uses array of pointers to print the address locations of two strings and the strings themselves. Run and test the program. Change the array of pointers col to a 2D array of size 2 by 8. See the variation in the memory spaces allocated to each string.

```
char *col[] = {"Red", "Blue"};
int i,j;
for(i=0;i<2;i++)
{
    printf("\n%d ", *(col+i));
    j=0;
    while(*(*(col+i)+j) != '\0')
    {
        printf("%c ", *(*(col+i)+j));
        j++;
    }

}
```

2. The following C program takes a sentence and a word as input from the user and finds the number of occurrence of the word in the sentence. The words of a sentence may be separated by more than one space. Run and test the program for various possible inputs.

```c
int main()
{
    char sentence[100],word[20],temp[20]; // string declarations
    int i=0,j=0,occurences=0;// i points current position within sentence
    // j points current position within word printf("Enter a sentence:\n");
    gets(sentence);
    printf("Enter the word to be search:\n");
    gets(word);
    while(sentence[i]!='\0')
    {
        while(sentence[i]!=' '&& sentence[i]!='\0') /*Extract words from sentence in
            temp*/
        {
            temp[j++]=sentence[i];
            i++;
        }
        temp[j]='\0';

        if((strcmp(temp,word))==0) occurences++;

        if(sentence[i]== ' ')   i++; /* skip    additional  spaces  in  the sentences */
        j=0;
    }
    printf("Number of Occurences of word are %d",occurences);
    return 0;
}
```

In the above program, remove the function call strcmp and implement your own function for the same.

More exercises on strings.

1. Write a C function that takes a string as input and modifies the string by removing all consecutive duplicate characters. For example, if the input string is Commonness, the output should be Comones

2. Write a C program that reads multiple lines of input and displays only those lines that contain a specific word entered by the user.

3. Write a C program that takes as input a string containing a phone number prefixed with a "+91" and prints the city code (for landline numbers only) and the actual phone number. Example of some inputs: "+91-1596-515100", "+91-9223345690" etc.

4. Write a C program that replaces all occurrences of a substring with another string of equal size in the main string.

5. Write a program that takes data a line at a time and reverses the words of the line. For example, if the input is Computer Programming is Easy", the output should be "Easy is Programming Computer".

6. Implement the following functions by:

    a) isdigit()
    b) islower()
    c) isalpha()
    d) isspace()
    e) isupper()
    f) toupper()
    g) tolower()
    h) strcpy()
    i) strcat()
    j) strlen()