Birla Institute of Technology & Science, Pilani, Hyderabad Campus

First Semester 2020-2021
Computer Programming [CS F111] Lab 10

Practice Programs:

1. Write a C program to sort the strings in a dictionary order.

Hint: apply Bubble sort on strings.

Sample Input/Output:

```
Enter the number of strings: 5

Enter the strings:
abc
aabc
zhgas
zhg
ysrghk

OUTPUT:
Strings in ascending order:
aabc
abc
ysrghk
zhg
zhgas
```

Code:

```c
#include<stdio.h>
#include<string.h>
int main()
{
    int i,j,k,n;
    char str[50], t[50], arr[10][50];
    printf("\nEnter the number of strings: ");
    scanf("%d", &n);
    printf("\nEnter the strings:\n");
    for(i=0;i<n;i++)
    {
        scanf("%s",str);
        strcpy(arr[i],str);
    }
    for(i=1;i<n;i++)
    {
        for(j=0;j<n-i;j++)
        {
            if(strcmp(arr[j],arr[j+1])>0)
            //if current string is greater
            {
                //swap
                strcpy(t,arr[j]);
                strcpy(arr[j],arr[j+1]);
                strcpy(arr[j+1],t);
            }
        }
    }
    printf("\nOUTPUT:\nStrings in ascending order:\n");
    for(i=0;i<n;i++)
        printf("%s\n",arr[i]);
}
```

2. Write a C-Program to reverse a string without using any string library functions.

Code:

```c
#include<stdio.h>
void reverse(char* s,int n)
{
    char t;
    int i;
    for(i=0;i<n/2;i++)
    {
        t=s[i];
        s[i]=s[n-i-1];
        s[n-i-1]=t;
    }
    for(i=0;i<n;i++)
        printf("%c",s[i]);
    printf("\n");
}
int main()
{
    int length;
    char ch, str[50000];
    printf("\nEnter your string: ");
    scanf("%c",&ch);
    length=0;
    while(ch!='\n')
    {
        str[length]=ch;
        length++;
        scanf("%c",&ch);
    }
    printf("\nOUTPUT\nReversed String: ");
    reverse(str,length);
}
```

Sample Input/Output:

```
Enter your string: Hello World

OUTPUT
Reversed String: dlroW olleH
```

3. string.h library provides various functionalities to apply on the strings. Please observe the following program which mimics some of those functionalities.

Note: The code is pasted here instead of screenshots towards your benefit of quickly practicing them.

Code:

```
#include<stdio.h>
char* my_strcpy(char* destination, char* source)
{
	int i;
	for(i=0;source[i]!='\0';i++)
	{
		destination[i]=source[i];
	}
	destination[i]='\0'; //copies the null char
	return destination;
}
int my_strlen(const char*s)
{
	int n=0;
	for(int i=0;s[i]!='\0';i++)
		n++;
	return n;
}
char* my_strcat(char* destination, char* source)
{
	int i;
	int n=my_strlen(destination);
	for(i=0;i<my_strlen(source);i++)
		destination[n+i]=source[i];
	destination[n+i]='\0';
	return destination;
}
int my_strcmp(const char* s1, const char* s2)
{
	int n1=my_strlen(s1);
	int n2=my_strlen(s2);
	int flag=0;
	for(int i=0;i<n1 && i<n2;i++)
	{
		if(s1[i]>s2[i]){
			flag=1;
```

```
                break;
            }
            else if(s1[i]<s2[i]){
                flag=-1;
                break;
            }
        }
        if(flag==0){
            if(n1==n2)
                return 0;
            else if(n1<n2)
                return -1;
            else
                return 1;
        }
        return flag;
}
int main()
{
        char str1[500];
        char str2[500];
        char str3[500];
        printf("\nEnter first string: ");
        scanf("%s",str1);
        printf("\nEnter second string: ");
        scanf("%s",str2);
        printf("\nOUTPUT\n");
        //print length of both strings
        printf("lstr1=%d lstr2=%d\n",my_strlen(str1),my_strlen(str2));
        //copy str1 to str3 and print
        my_strcpy(str3,str1); //catching the returned pointer not necessary
        //as change is made to str3
        printf("%s\n",str3);
        //compare str1 and str2
        printf("strcmp=%d\n",my_strcmp(str1,str2));
        //concatenate str1 and str2 and print
        my_strcat(str1,str2); //catching the returned pointer not necessary
        //as change is made to str1
        printf("%s\n",str1);
}
```

Sample Input/Output:

```
Enter first string: BITS

Enter second string: PILANI

OUTPUT
lstr1=4 lstr2=6
BITS
strcmp=-1
BITSPILANI
```

Exercise Problems:

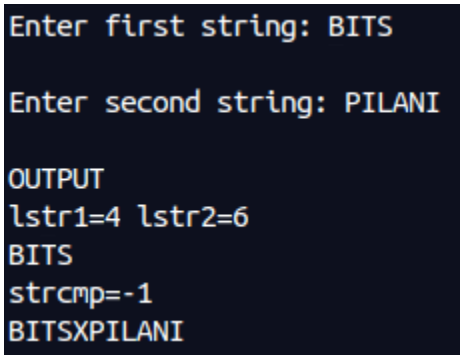
1. In sample Input/Output of practice program-1, if some string (example: zhg) is prefix of some other string (example: zhgas), then the prefix is printed prior to the other in ascending order. Now, modify the practice program-1 to print the other way round. That is lengthier string is printed prior to the prefix as shown below.
   Hint: you may make use of **strncmp()** from string.h.

```
Enter the number of strings: 5

Enter the strings:
aabc
abc
zhg
zhgas
ysrghk

OUTPUT:
Strings in ascending order:
aabc
abc
ysrghk
zhgas
zhg
```

2. Modify the practice program-2 to reverse each word of a given string without uitlizing any string library functions. The example Input/Output is as follows:

```
Enter your string: Hello World

OUTPUT
Reversed String: olleH dlroW
```

3. In the practice program-3, modify the **my_strcat()** function to insert character ‘X’ between strings to be concatenated. The example input/output is as follows:



*************ALL THE BEST*************

NOTE: Upload the screenshots of the Practice programs and Exercise programs along with the displayed results into your corresponding Google Classroom.

PATH to Submit the Screenshots:
Google Classroom --> Classwork --> View Assignment --> Create/Upload