CSE 115L: Programming Language I Lab (Section: 06)

Spring 2020

Lab-09 Tasks (Strings)

Tasks:

1. Declare two strings A and B of size 100 and 50, respectively. Then take user input of both strings. Concatenate (join) B at the end of A using loop. Display the concatenated string.

    Enter first string: Bangla
    Enter second string: desh
    After joining, first string: Bangladesh.

    **hints: if you use fgets() for string input, then do the following:
    str[strlen(str) - 1] = '\0';

```
#include <stdio.h>
#include <string.h>

int main(void)
{
        char str[100], str2[100];

        gets(str);
        gets(str2);

        strcat(str, str2);
        printf("After joining, first string:
%s", str);
        return 0;
}
```

```
#include <stdio.h>
#include <string.h>

int main(void)
{
        char str[100], str2[100];

        fgets(str, sizeof(str), stdin);
        fgets(str2, sizeof(str2), stdin);

        str[strlen(str)-1] = '\0';

        strcat(str, str2);
        printf("After joining, first string:
%s", str);
        return 0;
}
```

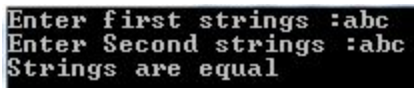
2. Write a program that deletes vowels from a string.

Enter a string: Bangladesh
String after deleting vowels: Bngldsh

```
#include <stdio.h>
#include <string.h>

int check_v(char c)
{
	if(c == 'a' || c == 'A' || c == 'e'|| c == 'E' || c == 'i' || c == 'I' || c == 'o' || c == 'O' || c
== 'u' || c == 'U')
	return 1;
	else
	return 0;
}

int main(void)
{
	char str[100], str_c[100];
	int i, j=0;

	printf("enter first string: ");
	fgets(str, sizeof(str), stdin);

	str[strlen(str)-1] = '\0';

	for(i=0; str[i]!='\0'; i++)
	{
	if(check_v(str[i]) == 0)
	{
	str_c[j] = str[i];
	j++;
	}
	}
	str_c[j] = '\0';
	puts(str_c);

	return 0;
}
```

Solutions of previous tasks:

1. Write a program to compare two strings without using C library function.




```
#include <stdio.h>

int main(void)
{
	char str[100], name[100], i, flag=1;

	printf("enter first string: ");
	fgets(str, sizeof(str), stdin);
	printf("enter second string: ");
	fgets(name, sizeof(name), stdin);

	for(i=0; (str[i]!='\0') || (name[i]!='\0'); i++)
	{
	if(str[i] != name[i])
	{
	flag = 0;
	break;
	}
	}

	if(flag == 1)
	printf("same");
	else
	printf("no");
	return 0;
}
```

2. Check whether an input string is palindrome or not. A string is a palindrome if it remains the same after you reverse it.
   For example, “racecar”, “level”, “12321”, “madam” etc.

   Sample:
   Enter a string: racecar
   It’s a palindrome

```
#include <stdio.h>
#include <string.h>

int main(void)
{
        char str[100], i, flag=1;

        printf("enter first string: ");
        fgets(str, sizeof(str), stdin);

        int len = strlen(str);

        for(i=0; i<len/2; i++)
        {
        if(str[i] != str[len-2-i])
        {
        flag = 0;
        break;
        }
        }

        if(flag == 1)
        printf("palindrome");
        else
        printf("not palindrome");
        return 0;
}
```

```
#include <stdio.h>
#include <string.h>

int main(void)
{
        char str[100], rev[100];

        printf("enter first string: ");
        fgets(str, sizeof(str), stdin);

        str[strlen(str)-1] = '\0';

        strcpy(rev, str);
        strrev(rev);

        printf("%d", strcmp(str, rev));

        if(strcmp(str, rev) == 0)
        printf("palindrome");
        else
        printf("not palindrome");

        return 0;
}
```