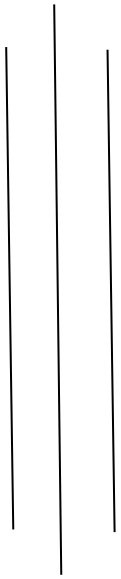
# TRIBHUVAN UNIVERSITY

## INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS



Subject: C-Programming

Experiment Number: 9

Title: **STRING**

Date: 14th August,2021


Submitted By:

Name: Susheel Thapa

Program: BCT

Section: CD

Roll No: 077BCT090

Submitted To:

Department of Electronics and Computer Engineering

Checked By:

# Background Theory:

## Introduction to string:

String is not a fundamental data type in C.
Like many other programming language like javaScript or python in c string is not a fundamental data type. In C string is a sequence of character ending in a \0 character sequence.
Thus a string with 10 character is contained in array of 11 character because there must be an \0 character at the end for an string to exist.
If there are other characters after the \0 character those character are ignored and only other character are displayed.

\0 character indicates the end of string.
char a[10]={'r','a',\0,'m'};
Here a is a string with only 2 characters and m is not part of the string although it is part of a the array

There are a lot of string handling function must of these function accepts the first argument as a pointer which represent the first element of array.

Strlrn():

Syntax:
strlen(stringVariable)
is a string handaling function which accepts the character pointer as its first element and returns the length of the string as output.
The important property about strlen() is that it looks for the first \0 character and does not proceed further.
strlen() function is defined in <string.h> header file.
int a= strlen(a);

strcat():
the function prototype for strcat() is char* strcat(char* dest,char * src)
This means that the function appends the string in the src at the end of the dest and returns the dest pointer too.
The important precaution to betaken is src should be large enough to include the both first and second string.
The function adds \0 character at the end of resut but if the dest is small \0 might not be included and it might endup being just array of characters.

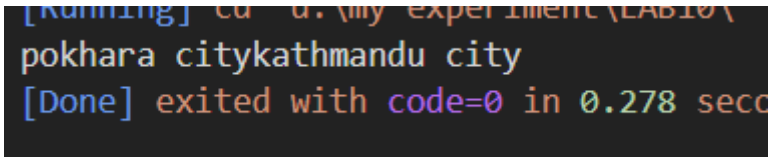
strcpy()
strcpy() is  a function defined in <string.h> to copy a string from one destination to another we can use the strcpy() function for this purpose,
Syntax
char *strcpy(char *dest, const char *src)
The function copies the string of the src to the string dest and returns the pointer to dest.

strcmp()
strcmp() function takes two function as argument and returns the
int strcmp(const char *str1, const char *str2)
the function is defined in <string.h> and is used to analyze the two string it returns 0 if str1 and str2 are completely identical
it returns a negative number if str2 is less than str2 and a positive number if str1 is greater than str2.

TWO DIMENSIONAL ARRAY OF STRINGS
A two dimensional array of string is an array of pointer which represents the string.
Thus the elements of two dimensional array represent the pointer to the string.

1.
```
#include <stdio.h>
#include <conio.h>
#include <string.h>
int main(){
   int i;
   char name1[]="pokhara city";
   char name2[]= {'k','a','t','h','m','a','n','d','u',' ','c','i','t','y','\0'};
   for(i=0;i<strlen(name1);i++){
      printf("%c",name1[i]);
   }
   for(i=0;i<strlen(name2);i++){
      printf("%c",name2[i]);
   }
}
```

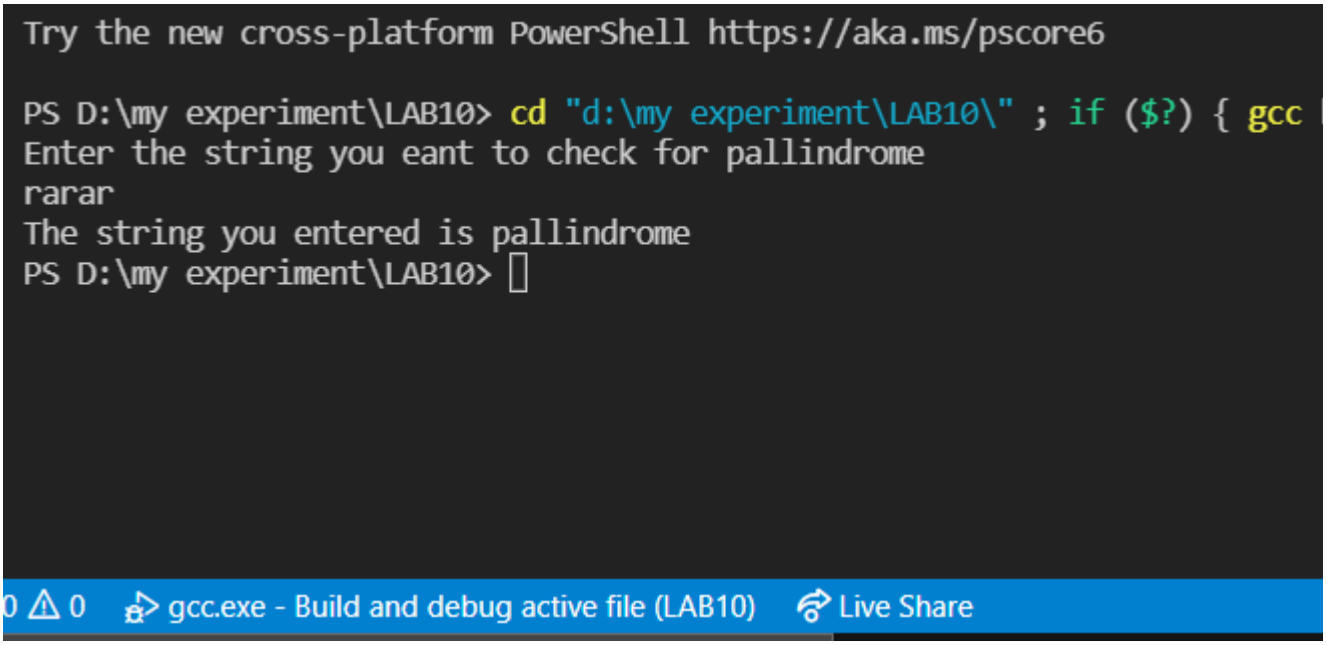
OUTPUT:



2.Write a program to check whether a string given by user is pallindrome or not.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
char* reverse(char *s){
    int l=strlen(s);
     char*r = malloc(l+1);
     for(int i=l-1;i>=0;i--){
         r[l-1-i]=s[i];
     }
     r[l]='\0';
     return r;
}
int main(){
    char* a = malloc(sizeof(char)*20);
    char* b=NULL;
    printf("Enter the string you eant to check for pallindrome\n");
    fflush(stdin);
    gets(a);
    b= reverse(a);
    if(strcmp(a,b)==0){
        printf("The string you entered is pallindrome\n");
    }
    else{
        printf("The string you entered is not pallindrome\n");
    }
}
```

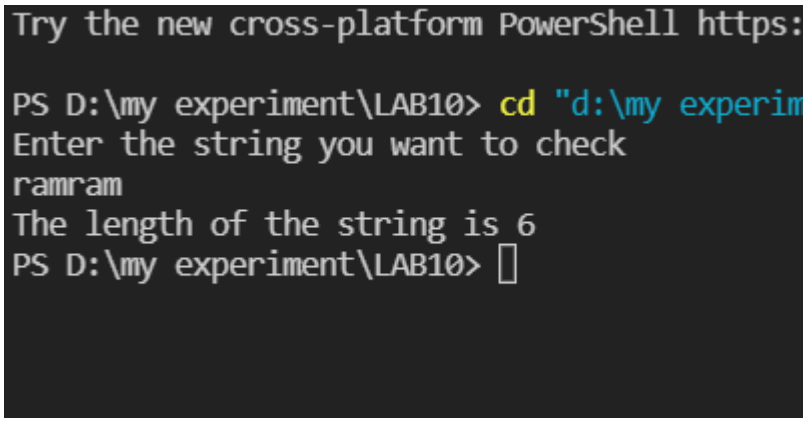
OUTPUT:




2.Write a program to return the number of characters in a string.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
int stringLength(char *a){
   int r=0;
   char b=*a;
   while(b!='\0'){
      r++;
      b=*(++a);
   }
   return r;
}
int main(){
   char ch[100];
   printf("Enter the string you want to check\n");
   fflush(stdin);
   gets(ch);
   int n=stringLength(ch);
   printf("The length of the string is %d",n);
}
```

OUTPUT:




4.Write a program to read a string using gets(). Pass it to a function that finds the longest words of string count the number of vowels and consonant from the string and return the counts and the word from the main().

```
#include <stdio.h>
#include <stdlib.h>
void longe(char *word,char ** s,int *n,int *v,int *co){
    *s=(char*)word;
    char *f=(char*)word;
    char *l=NULL;
    *n=0;
    int n1=0;
    if(*word=='\0'){
        *s=NULL;
        return;
    }
    char c=*f;
    while(c!='\0'){
        c=*(++word);
        if(c==' '){
            l=word;
            n1=l-f;
            f=word+1;
            if(n1>*n){
                *s=f;
                *n=n1;
```

```
            }
        }
        if(c=='\0'){
            l=word;
            n1=l-f;
            if(n1>*n){
                *s=f;
                *n=n1;
            }
        }
    }
    for(int i=0;i<*n;i++){
        if(*((*s)+i)=='a'|| *((*s)+i)=='e'||*((*s)+i)=='i'||*((*s)+i)=='o'||*((*s)+i)=='u'||*((*s)
+i)=='A'|| *((*s)+i)=='E'||*((*s)+i)=='I'||*((*s)+i)=='O'||*((*s)+i)=='U'){
            (*v)++;
        }
        else{
            (*co)++;
        }
    }
    return;
}
int main(){
    char word[100];
    printf("Enter the sentence you want to enter\n");
    fflush(stdin);
    gets(word);
    char*a =NULL;
    int n,v=0,c=0;
    longe(word,&a,&n,&v,&c);
    printf("The longest word in the sentence was found to be ");
    for(int i=0;i<n;i++){
        printf("%c",*(a+i));
    }
    printf("\nThe number of vowels and consonants in the word was found to be %d and
%d respectively" ,v,c);
    return 0;
}
```

OUTPUT:

```
  ^~
PS D:\my experiment\LAB10> cd "d:\my experiment\LAB10\" ; if ($?) { gcc hellowworld.c -o hellowwor
Enter the sentence you want to enter
Big Bad BULLL
The longest word in the sentence was found to be BULLL
The number of vowels and consonants in the word was found to be 1 and 4 respectively
PS D:\my experiment\LAB10>
```

5.Write a program to reverse a word using recursive function.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
void rever(char *words,int first,int last,char*re){
   if(first>last) {
      return;
   }
   if(first==last){
      re[first]=words[first];
      return;
   }
   *(re+first)=words[last];
   *(re+last)=words[first];
   rever(words,++first,--last,re);
   return;
}
int main(){
   char words[100];
   printf("Enter the word.");
   fflush(stdin);
   gets(words);
   char *reverse = malloc(sizeof(char)*(strlen(words)+1));
   rever(words,0,strlen(words)-1,reverse);
   reverse[strlen(words)]='\0';
   fflush(stdout);
   puts(reverse);
   return 0;
}
```

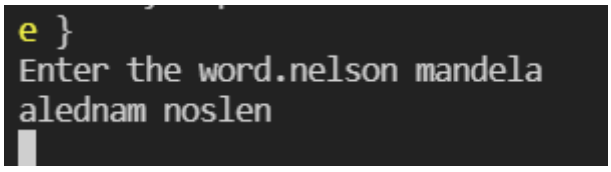
OUTPUT:

```
PS D:\my experiment\LAB10> cd "d:\my exper
Enter the word.nelson mandela
alednam noslen
PS D:\my experiment\LAB10>
```

8.Write a program that will read a string and rewrite it in the alphabetical order. For example the word NEPAL should be written as AELNP.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
void rever(char *words,int first,int last,char*re){
   int start=0;
   int end= strlen(words)-1;
   while(start<=end){
      *(re+start)=words[end];
      *(re+end)=words[start];
      start++;
      end--;
   }
   return;
}
int main(){
   char words[100];
   printf("Enter the word.");
   fflush(stdin);
   gets(words);
   char *reverse = calloc((strlen(words)+1),sizeof(char));
   rever(words,0,strlen(words)-1,reverse);
   reverse[strlen(words)]='\0';
   fflush(stdout);
   puts(reverse);
   getch();
   return 0;
}
```

OUTPUT:




9.Write a program to arrange the word in ascending order

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <String.h>
void reverse(int first,char *newWord,int last){
   int begin=first+1;
```

```
    int end=last;
    int reference=first;
    char ram;
    if(first>=last){
        return;
    }
    if(begin==end){
        if(newWord[first]>newWord[last]){
            ram=newWord[first];
            newWord[first]=newWord[last];
            newWord[last]=ram;
            return;
        }
    }
    while(begin<end){
        while(newWord[reference]>newWord[begin] && begin<end){
            begin++;
        }
        while(newWord[reference]<=newWord[end] && begin<end){
            end--;
        }
        ram=newWord[begin];
        newWord[begin]=newWord[end];
        newWord[end]=ram;
    }
    if(newWord[reference]>newWord[begin]){
        ram=newWord[reference];
        newWord[reference]=newWord[begin];
        newWord[begin]=ram;
        reverse(first,newWord,begin-1);
        reverse(begin+1,newWord,last);
    }
    else if(newWord[reference]<=newWord[begin]){
        ram=newWord[begin-1];
        newWord[begin-1]=newWord[reference];
        newWord[reference]=ram;
        reverse(first,newWord,begin-2);
        reverse(begin,newWord,last);
    }
    return;
}
int main(){
    char word[100];
    printf("You are required to enter the word whose accending order you want to find\n");
    fflush(stdin);
    gets(word);
    char* wordAccending;
    wordAccending=calloc(strlen(word)+1,sizeof(char));
    strcpy(wordAccending,word);
```

```
    reverse(0,wordAccending,strlen(wordAccending)-1);
    printf("The string in asscending order is found to be \n");
    puts(wordAccending);
    getch();
    return 0;
}
```

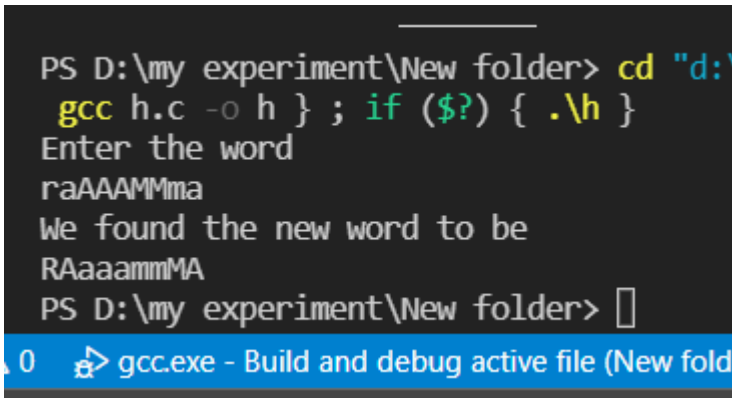
OUTPUT:

```
You are required to enter the word whose accending order you want to find
nelson mandela
The string in asscending order is found to be
 aadeellmnnnos
```

10.Write a program to read a string in main and pass it to a function. The function converts all the upper case letters to lower and vice versa.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void converter(char* word,int length)
{
    for(int i=0;i<length;i++){
        if(word[i]<='z' && word[i]>='a'){
            word[i]=word[i]-(int)('a'-'A');
        }
        else if (word[i]>='A' && word[i]<='Z'){
            word[i]=word[i]+(int)('a'-'A');
        }
    }
    return;
}
int main(){
    char word[100];
    printf("Enter the word\n");
    gets(word);
    converter(word,strlen(word));
    printf("We found the new word to be\n");
    puts(word);
}
```

OUTPUT:

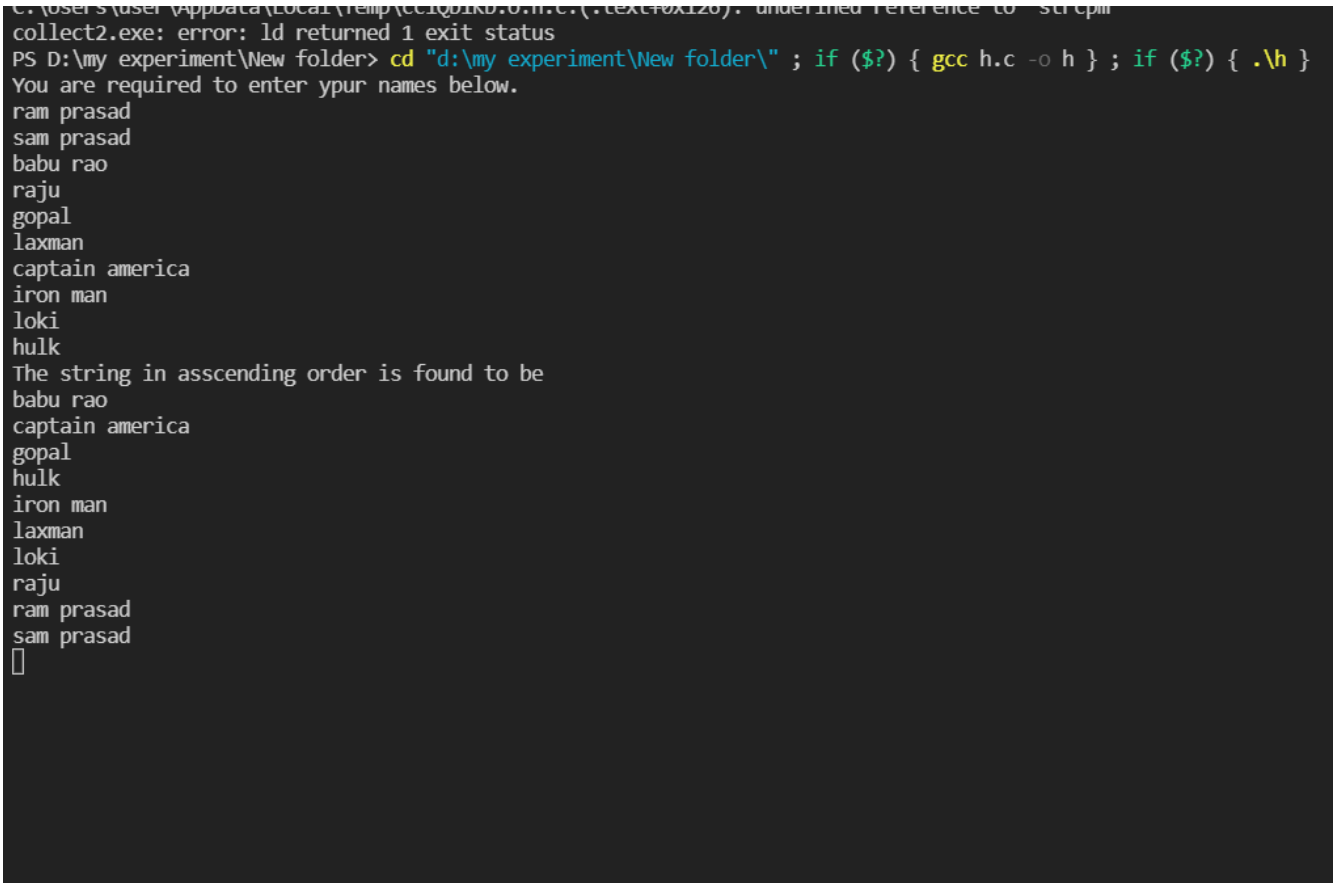



11. Write a program to read name of 10 students in main() pass the name list to a function that sort the array in ascending order.Display the sorted array from main.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <String.h>
void reverse(int first,char *Word[10],int last){
   int begin=first+1;
   int end=last;
   int reference=first;
   char* ram;
   if(first>=last){
      return;
   }
   if(begin==end){
      if(strcmp(Word[first],Word[last])>0){
         ram=Word[first];
         Word[first]=Word[last];
         Word[last]=ram;
      }
      return;
   }
   while(begin<end){
      while(strcmp(Word[reference],Word[begin])>0 && begin<=end){
         begin++;
      }
      while(strcmp(Word[reference],Word[end])<=0 && begin<end){
         end--;
      }
      ram=Word[begin];
      Word[begin]=Word[end];
      Word[end]=ram;
   }
```

```
    if(strcmp(Word[reference],Word[begin])>0){
        ram=Word[reference];
        Word[reference]=Word[begin];
        Word[begin]=ram;
        reverse(first,Word,begin-1);
        reverse(begin+1,Word,last);
    }
    else if(strcmp(Word[reference],Word[begin])<=0){
        ram=Word[begin-1];
        Word[begin-1]=Word[reference];
        Word[reference]=ram;
        reverse(first,Word,begin-2);
        reverse(begin,Word,last);
    }
    return;
}
int main(){
    char *word[10];
    printf("You are required to enter ypur names below.\n");
    for(int i=0;i<10;i++){
        word[i]=calloc(100,sizeof(char));
        fflush(stdin);
        gets(word[i]);
    }
    reverse(0,word,9);
    printf("The string in asscending order is found to be \n");
    for(int i=0;i<10;i++){
        puts(word[i]);
    }
    getch();
    return 0;
}
```

OUTPUT:




12.Write a program to do the following
To print “who is the prime minister of Nepal ?”
To accept the answer,
To print “good” of the answer is correct
To print try again if the answer is wrong
To display the answer when the anser is wrong even at the third attempt.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
int main(){
   printf("Enter the prime minister of Nepal ?\n");
   char *name;
   name=calloc(100,sizeof(char));
   gets(name);
   int count=1;
   while(strcmp(name,"Shere Bahadur Deuba") !=0 && count<=3){
      printf("Try again\n");
      gets(name);
      count++;
   }
   if(strcmp(name,"Shere Bahadur Deuba") != 0){
```

```
        printf("You have been entering the wrong name");
        printf("The corect name of the prime minister of Nepal is Shere Bahadur Deuba");
    }
    if(strcmp(name,"Shere Bahadur Deuba")==0){
        printf("Good you know who is the prime minister of Nepal");
    }
    return 0;
}
```

OUTPUT: