1. Write a function to calculate length of the string

Program -
```
#include<stdio.h>

int length(char[]);
int main()
{
    char str[50];

    printf("Enter a string: ");
    fgets(str,50,stdin);
    printf("Length of the string = %d",length(str));
    return 0;
}

int length(char ch[])
{
    int len;
    for(len = 0 ; ch[len] ; len++);
    return len-1;
}
```

Output -
Enter a string: Hello World
Length of the string = 11

---

2. Write a function to reverse a string.

Program -
```
#include<stdio.h>

void strRev(char[],int);
int main()
{
    char str[50];
    int l;

    printf("Enter a string: ");
    fgets(str,50,stdin);

    for(l = 0 ; str[l] ; l++);
    strRev(str , l);
    return 0;
}
```

```c
void strRev(char ch[] , int len)
{
    char s;
    int i;

    for(i = 0 ; i < len/2 ; i++)
    {
        s = ch[i];
        ch[i] = ch[len - 1 - i];
        ch[len - 1 - i] = s;
    }

    printf("Reverse is : %s",ch);
}
```

Output -
Enter a string: Ineuron
Reverse is :
noruenI

---

3. Write a function to compare two strings.

Program -
```c
#include<stdio.h>
#include<string.h>

void compare(char[],char[]);

int main()
{
    char a[50],b[50];

    printf("Enter first string: ");
    fgets(a,50,stdin);
    printf("Enter second string: ");
    fgets(b,50,stdin);

    compare(a,b);

    return 0;
}
void compare(char ch[] , char str[])
{
    int c = strcmp(ch,str);
    if(c == 0)
        printf("Strings are equal");
    else
        printf("Strings are not equal");
}
```

Output -
Enter first string: mysirg
Enter second string: mysirg
Strings are equal

---

4. Write a function to transform string into uppercase

Program -
```c
#include<stdio.h>

void uppercase(char[]);

int main()
{
    char a[50];

    printf("Enter a string: ");
    fgets(a,50,stdin);

    uppercase(a);

    printf("String in uppercase: %s",a);
    return 0;
}

void uppercase(char ch[])
{
    int i;

    for(i = 0 ; ch[i] ; i++)
    {
        if(ch[i] >= 'a' && ch[i] <= 'z')
            ch[i] = ch[i] - 32;
    }
}
```

Output -
Enter a string: peter
String in uppercase: PETER

---

5. Write a function to transform a string into lowercase

Program -
```c
#include<stdio.h>

void lowercase(char[]);

int main()
```

```
{
    char a[50];

    printf("Enter a string: ");
    fgets(a,50,stdin);

    lowercase(a);

    printf("String in lowercase: %s",a);
    return 0;
}

void lowercase(char ch[])
{
    int i;

    for(i = 0 ; ch[i] ; i++)
    {
        if(ch[i] >= 'A' && ch[i] <= 'Z')
            ch[i] = ch[i] + 32;
    }
}
```

Output -
Enter a string: AVENGERS
String in lowercase: avengers

---

6. Write a function to check whether a given string is an alphanumeric string or not. (Alphanumeric string must contain at least one alphabet and one digit)

Program -
```
#include<stdio.h>

int checkString(char[]);

int main()
{
    char ch[50];

    printf("Enter a string: ");
    fgets(ch,50,stdin);

    if(checkString(ch))
        printf("Alphanumeric string");
    else
        printf("Not an alphanumeric string");

    return 0;
}
```

```c
int checkString(char str[])
{
    int digits = 0 , alpha = 0 , i;

    for(i = 0 ; str[i] ; i++)
    {
        if(str[i] >= '0' && str[i] <= '9')
            digits++;
        if(str[i] >= 'a' && str[i] <= 'z' || str[i] >= 'A' && str[i]
<= 'Z')
            alpha++;
        if(digits >= 1 && alpha >= 1)
            return 1;
    }
    return 0;
}
```

Output -
Enter a string: apj123
Alphanumeric string

---

7. Write a function to check whether a given string is palindrome or not.

Program -
```c
#include<stdio.h>
#include<string.h>

int check(char[]);

int main()
{
  char ch[20];
  printf("Enter a string: ");
  scanf("%s",ch);
  if(check(ch))
    printf("String is palindrome");
  else
    printf("String is not palindrome");
  return 0;
}

int check(char str[])
{
  int len , i;
  char temp[20] , s;
  len = strlen(str);

  for(i = 0 ; i <= len ; i++)
    temp[i] = str[i];
```

```c
    for(i = 0 ; i < len/2 ; i++)
    {
        s = temp[i];
        temp[i] = temp[len-1-i];
        temp[len-1-i] = s;
    }

    if(strcmp(str,temp) == 0)
        return 1;
    else
        return 0;
}
```

Output -
Enter a string: RADAR
String is palindrome

---

8. Write a function to count words in a given string

Program -
```c
#include<stdio.h>
#include<string.h>

void remove_extra_spaces(char*, int);
int countWords(char*);

int main()
{
    char str[100];
    int len;

    printf("Enter a string: ");
    fgets(str,100,stdin);

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

    remove_extra_spaces(str, len);

    printf("\nNumber of words: %d",countWords(str));

    return 0;
}

void remove_extra_spaces(char *ptr, int size)
{
    char ch[size+1];
    char *p = ch;
    int i = 0, j = 0;
```

```c
        while(*(ptr+i))
        {
            while(*(ptr+i) == ' ')
                i++;

            while(*(ptr+i) != ' ' && *(ptr+i) != '\0')
            {
                ch[j] = *(ptr+i);
                i++; j++;
            }
            if(ch[j-1] == ' ' && *(ptr+i) == '\0')
              j--;

            ch[j] = *(ptr+i);
                j++;
        }

    strcpy(ptr,ch);
}

int countWords(char *p)
{
  int i, count = 0;
  for(i = 0 ; *(p+i) ; i++)
  {
      if(*(p+i) == ' ')
         count++;
  }
  return count+1;
}
```

Output -
Enter a string: A quick brown fox jumps over the lazy dog.
Number of words: 9

---

9. Write a function to reverse a string word wise. (For example if the given string is "Mysirg Education Services" then the resulting string should be "Services Education Mysirg")

Program -
```c
#include<stdio.h>
#include<string.h>

void swap(char[], int, int);

int main()
{
    char str[50];
```

```c
    int i = 0, start = 0, end = 0, flag = 0;

    printf("Enter a string: ");
    fgets(str,50,stdin);

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

    while(str[i] != '\0')
    {
        while(str[i] != ' ')
        {
            if(str[i] == '\0')
            {
                flag = 1;
                break;
            }
            end++;
            i++;
        }

        if(flag == 1)
        {
            swap(str,start,end-1);
            break;
        }
        swap(str,start,end-1);
        start = end+1;
        i++;
        end++;
    }

    swap(str,0,end-1);
    printf("Reverse of the string wordwise:-\n%s",str);

    return 0;
}

void swap(char ch[], int start, int end)
{
    int i = start, j = end;
    char s;

    while(i <= j)
    {
        s = ch[i];
        ch[i] = ch[j];
        ch[j] = s;
        i++;
        j--;
    }
}
```

Output -
Enter a string: I love C programming
Reverse of the string wordwise:-
programming C love I

---

10. Write a function to find the repeated character in a given string.

Program -
```c
#include<stdio.h>

void display(char[]);
int main()
{
    char str[50];

    printf("Enter a string\n");
    fgets(str,50,stdin);


    printf("Repeated characters are\n");
    display(str);

    return 0;
}

void display(char ch[])
{
    char s;
    int i , j;

    for(i = 0 ; ch[i] ; i++)
    {
        s = ' ';
        for(j = 0 ; ch[j] ; j++)
        {
            if(i != j && ch[i] == ch[j] &&  ch[j] != ' ')
            {
                s = ch[j];
                ch[j] = ' ';
            }
        }
        if(s != ' ')
            printf("%c\n",s);
    }
}
```

Output -
Enter a string
mama

Repeated characters are
m
a