1. **blankSpaceReplacement**

```
/*. WAP to Take in a String and Replace Every Blank Space with special symbol*/
void main()
{
    char str[23];
    gets(str);

    char c=getchar();



    for(int i=0;i<strlen(str);i++)
    {
        if(str[i]==' ')
        str[i]=c;
    }
    printf("%s",str);

}
```

2. **exchangeLastAndFirst**

```
/*WAP to Form a New String where the First Character and the Last Character have
been Exchanged*/
#include<stdio.h>
#include<string.h>
char*getExchanged(char*str)
{
    char*str1=(char*)malloc(sizeof(str));
    for(int i=0,j=strlen(str)-1;i<strlen(str)-1;i++)
    {
        if(i==0)
        {
          char temp=str[i];
          str1[i]=str[j];
          str1[j]=temp;
        }
        else
        str1[i]=str[i];
    }
    return str1;
```

```
        }
        void main()
        {
                printf("enter the string size=");
                int size;
                scanf("%d",&size);
                fflush(stdin);
                printf("enter the string=");
                char str[size];
                gets(str);

                char*ch=getExchanged(str);
                printf("%s",ch);

        }
```

3. findNdReplace

```
/*WAP Replace all Occurrences of 'a' with $ in a String*/


#include<stdio.h>

#include<string.h>

char* findNdReplace(char*str,char f,char r)

{

        int size=strlen(str);

        for(int i=0;i<size;i++)

        {

                if(str[i]==f)

                str[i]=r;

        }

        return str;

}

void main()

{

        char str[10];

        printf("enter the string=");

        gets(str);
```

```
    char f,r;

    fflush(stdin);

    printf("enter the character to find=");

    f=getchar();

    printf("enter the character to replace=");

    fflush(stdin);

    r=getchar();


    printf("%s",findNdReplace(str,f,r));

}
```

4. **longestString**

```
/*WAP to Take in Two Strings and Display the Larger String without Using Built-in
Functions*/
#include<stdio.h>
#include<string.h>
void largestString(char*s1,char*s2)
{
        int len1=0,len2=0;
        for(int i=0;s1[i]!='\0';i++)
        {
                len1++;
        }
        for(int i=0;s2[i]!='\0';i++)
        {
                len2++;
        }
        if(len1>len2)
                printf("%s",s1);

        else
        printf("%s",s2);


}
void main()
{
        char s1[20];
        char s2[20];

        gets(s1);
        fflush(stdin);
        gets(s2);
```

```
                largestString(s1,s2);

        }
```

5. **numberOfOvels**

```
/*. WAP to Count the Number of Vowels in a String
*/
void main()
{
        char str[10];
        scanf("%s",str);
        char arr[]={'a','e','i','o','u','A','E','I','O','U'};
        int i;
        int count=0;
        for(i=0;str[i]!='\0';i++)
        {
                for(int j=0;j<sizeof(arr);j++)
                {
                        if(str[i]==arr[j])
                        count++;
                }
        }
        printf("number of ovels=%d",count);
}
```

6. **numberOfWordsIn_string**

```
/*WAP to Calculate the Number of Words Present in a String*/
#include<stdio.h>
#include<string.h>

int numberOfWords(char*str)
```

```
{

  if(strlen(str)<2)
  printf("string is holding a only character\n");

  else
  {
        int count=1;
        for(int i=0;i<strlen(str);i++)
    {
                if(str[i]==' ')
                count++;

        }
        return count;
        }
}
void main()
{
        char str[23];
        gets(str);


        printf("Number of words in a string=%d",numberOfWords(str));
}
```

7. **palindromeString**

```
/*Write a program to check the string is palindrome or not*/
#include<stdio.h>
#include<string.h>
```

```
void main()

{

            char str[20];

            gets(str);


            char cp[20];

            strcpy(cp,str);


            char*str1=strrev(str);


   int flag=1;

            for(int i=0;str[i]!='\0';i++)

            {

                        if(cp[i]!=str1[i])

                        {

                                    flag=0;

                                    break;

                        }

            }


            if(flag)

            printf("%s is palindrome",str);


            else

            printf("%s is not palindrome",str);


}
```

8. **removeElement**

```
/*  WAP to Remove the nth Index Character from a Non-Empty String

*/
```

```
#include<stdio.h>

#include<string.h>

char* removeElement(char*str,int n)

{

    int size=strlen(str);

    char*str1=(char*)malloc(sizeof(char)*size);


    for(int i=0,j=0;i<size;i++,j++)

    {

        if(i==n)

          i++;

        str1[j]=str[i];

    }

    return str1;

}

void main()

{

    printf("enter the string=");

    char str[10];

    scanf("%s",str);


    printf("enter the nth index=");

    int n;

    scanf("%d",&n);


    char* ch=removeElement(str,n);

    printf("%s",ch);

}
```

9. **removeOddIndex**
   /*WAP to Remove the Characters of Odd Index Values in a String*/
   void main()

```
{
    char str[23];
    gets(str);

    char strNew[strlen(str)/2];

    for(int i=0,j=0;i<strlen(str);i++,j++)
    {
        if(i%2!=0)
        i++;
        strNew[j]=str[i];
    }
    printf("%s",strNew);
}
```

10. searchChar

```
/*Write a program to scan string from user then scan a single character and search it
in a accepted string.
*/

#include<stdio.h>
#include<string.h>
void main()
{
    char str[10];
    printf("enter the string=");
    gets(str);
    char c;
    printf("enter the character=");
    c=getchar();

    char *ch=strchr(str,c);
    if(ch!=NULL)
    printf("character %c found at index %d\n",c,ch-str);
    else
    printf("character not found\n");

}
```