# Assignment 8

1. read from a terminal using scanf function and print using printf function.

```
#include<stdio.h>
int main()
{
    int n;
    char ch;
    char str[20];
    printf("Enter an integer : ");
    scanf("%d",&n);
    printf("Enter a word : ");
    scanf("%s",str);
    fflush(stdin);
    printf("Enter a character : ");
    scanf("%c",&ch);
    printf("You have entered :\n");
    printf("Integer : %d\nCharacter : %c\nWord : %s",n,ch,str);
    getch();
    return 0;
}
```

2. read a lines of text from a terminal using fgets function and print using puts function.

```
int main()
{
   char str[20];
   printf("Enter a line : ");
   fgets(str,20,stdin);
   puts(str);
   getch();
   return 0;
}
```

# 3. convert a. Upper case to Lower case b. Lower case to Upper case c. Toggle case d. Sentence case

```
#include<stdio.h>
int main()
{
   char str[50],i;
   printf("Enter a string : ");
   fgets(str,50,stdin);
   //Lower case to Upper case
   printf("Lower case to Upper case\n");
   for(i=0; str[i]!='\0'; i++)
   {
      if(str[i]>=65 && str[i]<=90)
         continue;
      if(str[i]>=97 && str[i]<=122)
      {
         str[i]=str[i]-32;
      }
   }
   puts(str);
   //Upper case to Lower case
   printf("Upper case to Lower case\n");
   for(i=0; str[i]!='\0'; i++)
   {
      if(str[i]>=97 && str[i]<=122)
         continue;
      if(str[i]>=65 && str[i]<=90)
      {
         str[i]=str[i]+32;
      }
   }
   puts(str);
   // Toggle case
   printf("Toggle case\n");
   for(i=0; str[i]!='\0'; i++)
   {
      if(str[i]>=65 && str[i]<=90)
```

```
        str[i]=str[i]+32;
    i++;
    while(str[i]!=32 && str[i]!='\0')
    {
        if(str[i]>=97 && str[i]<=122)
            str[i]=str[i]-32;
        i++;
    }

}
puts(str);
//Sentence case
printf("Sentence case\n");
for(i=0; str[i]!='\0'; i++)
{
    if(i==0)
    {
        if(str[0]>=97 && str[0]<=122)
            str[0]=str[0]-32;
    }
    else
    {
        if(str[i]=='.')
        {
            i++;
            if(str[i]>=97 && str[i]<=122)
                str[0]=str[0]-32;
        }
        else
        {
            if(str[i]>=65 && str[i]<=90)
                str[i]=str[i]+32;
        }
    }
}
puts(str);
getch();
return 0;  }
```

## 4. perform String Concatenation (With and Without String Handling Functions).

```
#include<stdio.h>
int main()
{
   char str1[100],str2[20],i=0,l;
   printf("Enter a string : ");
   gets(str1);
   printf("Enter another : ");
   gets(str2);
   puts(str1);
   puts(str2);
   // using library function
   strcat(str1,str2);
   puts(str1);
   // without using library function
   for(l=0; str1[l]!='\0'; l++);
   for(i=0; str2[i]!='\0'; i++)
   {
      str1[l]=str2[i];
      l++;
   }
   str1[l]='\0';
   puts(str1);
   getch();
   return 0; }
```

5. perform String Reversal (With and Without String Handling Functions).

```
#include<stdio.h>
#include<string.h>
int main()
{
   char str[20],temp;
   int le,i;
   printf("Enter a string : ");
   fgets(str,20,stdin);
   //using library function
   puts(strrev(str));
   //without using library function
   for(le=0; str[le]!='\0'; le++);
   for(i=0; i<le/2; i++)
   {
      temp=str[i];
      str[i]=str[le-i-1];
      str[le-i-1]=temp;
   }
   puts(str);
   getch();
   return 0;  }
```

# 6. perform Substring Extraction (With and Without String Handling Functions).

```
#include<stdio.h>
#include<string.h>
int main()
{
   char str[20],*s,sub[20];
   int i,j,index,temp;
   printf("Enter a string : ");
   fgets(str,20,stdin);
   puts(str);
   //using library function
   s=strstr(str,"world");
   puts(s);
   //without using library function
   printf("Enter substring : ");
   gets(sub);
   i=j=index=0;
   while(sub[i]!='\0' && str[j]!='\0')
   {
      while(sub[i]!=str[j] && str[j]!='\0')
         j++;
      if(sub[i]==str[j])
      {
         index=j;
         while(sub[i]!='\0')
         {
            if(sub[i]!=str[j])
               break;
            i++; j++;
         }
         if(sub[i]=='\0')
            printf("String found at index %d\n",index);
         else
            j=index+1;   }  i=0 ;   }   }
```

# 7. copy one string into another and count the no of elements copied. (With and Without String Handling Functions).

```
#include<stdio.h>
#include<string.h>
int main()
{
   char str1[20],str2[20];
   int i,j;
   printf("Enter a string : ");
   fgets(str1,20,stdin);
   printf("Enter another string : ");
   fgets(str2,20,stdin);
   printf("First string is : %s",str1);
   printf("\nSecond string is : %s",str2);
   //using predefined function
   strcpy(str1,str2);
   printf("\nAfter Second string is copied to First string. (using predefined function)");
   printf("\nFirst string is : %s",str1);
   printf("\nSecond string is : %s",str2);
   //without using predefined function
   printf("\nEnter a string : ");
   fgets(str1,20,stdin);
   printf("Enter another string : ");
   fgets(str2,20,stdin);
   printf("First string is : %s",str1);
   printf("\nSecond string is : %s",str2);
   for(i=0; str1[i]!='\0'; i++)
   {
      str2[i]=str1[i];
   }
   str2[i]='\0';
   printf("\nAfter First string is copied to Second string. (Not using predefined function)");
   printf("\nFirst string is : %s",str1);
   printf("\nSecond string is : %s",str2);
}
```

8. read a string and prints if it is a palindrome or not.

```
#include<stdio.h>
#include<string.h>
int main()
{
   char str[20],copy[20],temp;
   int i,l;
   printf("Enter a string : ");
   gets(str);
   strcpy(copy,str);
   for(l=0; str[l]!='\0'; l++);
   for(i=0; i<l/2; i++)
   {
      temp=str[i];
      str[i]=str[l-i-1];
      str[l-i-1]=temp;
   }
   puts(str);
   puts(copy);
   if(strcmp(str,copy))
      printf("Not pallindrome");
   else
      printf("Pallindrome");
   getch();
   return 0;
}
```

## 9. read a line of text and count all occurrences of particular word.

```
#include<stdio.h>
#include<string.h>
int main()
{
   char str[20],*s,sub[20];
   int i,j,index,temp,count=0;
   printf("Enter a string : ");
   fgets(str,20,stdin);
   printf("Enter the word : ");
   gets(sub);
   i=j=index=0;
   while(sub[i]!='\0' && str[j]!='\0')
   {
      while(sub[i]!=str[j] && str[j]!='\0')
         j++;
      if(sub[i]==str[j])
      {
         while(sub[i]!='\0')
         {
            if(sub[i]!=str[j])
               break;
            i++; j++;
         }
         if(sub[i]=='\0')
         {
            count++;
         }
      }
      i=0;
   }
   printf("%d",count);
   getch();
   return 0;
}
```

10. read a string and rewrite it in the alphabetical order.

```
#include<stdio.h>
int main()
{
   char str[20],temp;
   int l,i,round;
   printf("Enter a string : ");
   gets(str);
   puts(str);
   l=strlen(str);
   // code to arranging in alphabetical order...
   for(round=1; round<l; round++)
   {
      for(i=0; i<l-round; i++)
      {
         if(str[i]>str[i+1])
         {
            temp=str[i];
            str[i]=str[i+1];
            str[i+1]=temp;
         }
      }
   }
   puts(str);
   getch();
   return 0;  }
```

## 11. Print the Words Ending with Letter S.

```
#include<stdio.h>
int main()
{
   char str[20];
   int start,i,l;
   printf("Enter a string : ");
   gets(str);

   //code to find word ending with letter s
   l=strlen(str);
   for(i=0; str[i]!='\0'; i++)
   {
      start=i;
      while(str[i]!='s' && str[i]!='\0' && str[i]!=32)
         i++;
      if(str[i]=='s' && str[i+1]==32 || str[i]=='s' && str[i+1]=='\0')
         print(str,start,i);
      else
      {
         if(str[i]=='s')
         {
            while(str[i]=='s')
               i++;
            if(str[i]==32 || str[i]=='\0')
```

```
                print(str,start,i);
            }
        }


    }
    getch();
    return 0;
}
void print(char str[],int start,int end)
{
    int i;
    for(i=start; i<=end; i++)
        printf("%c",str[i]);
    printf(" ");
}
```

## 12. Delete All Repeated Words in the line of text.

```
#include<stdio.h>
#include<string.h>
int main()
{
   int i=0,j=0,k=0,a,minIndex=0,maxIndex=0,max=0,min=0;
   char str1[100]={0},substr[100][100]={0},c;
   printf("Enter a sentence\n");
   gets(str1);
   while(str1[k]!='\0')//for splitting sentence
   {
      j=0;
      while(str1[k]!=' '&&str1[k]!='\0')
      {
         substr[i][j]=str1[k];
         k++;
         j++;
      }
      substr[i][j]='\0';
      i++;
      if(str1[k]!='\0')
      {
         k++;
      }
   }
   int len=i;
```

```
  //Removing repeated words same as removing repeated elements in arrays
  for(i=0;i<len;i++)
  {
   for(j=i+1;j<len;)
    {
     if(strcmp(substr[i],substr[j])==0)
      {
        for(k=j;k<len;k++)
         {
           strcpy(substr[k],substr[k+1]);
         }
           len--;
      }
     else
      {
       j++;
      }
    }
  }
for(i=0;i<len;i++)
{
 printf("%s ",substr[i]);
}
printf("\n");
getch();
return 0;  }
```