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

function.

```
#include<stdio.h>

int main(){

char name[20];

 printf("Enter name: ");

 scanf("%s", name);

 printf("Your name is %s.", name);

 return 0;

}
```

OUTPUT:

```
Enter name: Saubhagya ranjan rout

Your name is saubhagya.
```

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

using puts function.

```
#include<stdio.h>

int main(){

char name[20];

 printf("Enter name: ");

 fgets(name,sizeof(name),stdin);

 printf("name: ");

 puts(name);

 return 0;

}
```

OUTPUT:

Enter name: saubhagya ranjan rout

name: Saubhagya ranjan rout

3. convert

a. Upper case to Lower case

b. Lower case to Upper case

c. Toggle case

d. Sentence case

```
//upper case to lower case
#include <stdio.h>
#include <string.h>
int main(){
 char s[100];
 int i;
 printf("Enter a string : ");
 gets(s);
 for (i = 0; s[i]!='\0'; i++) {
 if(s[i] >= 'A' && s[i] <= 'Z') {
 s[i] = s[i] + 32;
 }
 }
 printf("\nString in Lower Case = %s", s);
 return 0;
```

```
}
```

OUTPUT:

```
nter a string : APPLE

String in Lower Case = apple
```

//UPPER CASE

```
#include <stdio.h>

#include <string.h>

int main() {

char s[100];

int i;

printf("Enter a string : ");

gets(s);

for (i = 0; s[i]!='\0'; i++) {

if(s[i] >= 'a' && s[i] <= 'z') {

s[i] = s[i] - 32;

}

}

printf("\nString in Upper Case = %s", s);

return 0;

}
```

OUTPUT:

```
Enter a string : apple

String in Upper Case = Apple
```

//TOGGLE CASE

```
#include <stdio.h>

#include <string.h>

int main(){

char Str[100];

int i;

printf("Enter any string: ");

gets(Str);

for (i = 0; Str[i]!='\0'; i++){

if(Str[i] >= 'a' && Str[i] <= 'z'){

Str[i] = Str[i] - 32;

}

else if(Str[i] >= 'A' && Str[i] <= 'Z'){

Str[i] = Str[i] + 32;

}

}

printf("\n The Given String after toggle case = %s", Str);

return 0;

}
```

OUTPUT:

```
Enter any string: HeLlO

The Given String after toggle case = hElLo
```

//SENTENCE CASE

```
#include <stdio.h>

#include <ctype.h>
```

```
int main(){

char str[100];

printf("Enter a string : ");

gets(str);

str[0] = toupper(str[0]);

printf("The string is: %s.",str);

return 0;

}
```

OUTPUT:

```
Enter a string : hello programmers

The string is: Hello programmers.
```

4. perform String Concatenation (With and Without String Handling

Functions).

//CONCATE WITHOUT FUNC

```
#include <stdio.h>

int main() {

char s1[100] = "Hello ", s2[] = "World";

int length, j;

length = 0;

while (s1[length] != '\0') {

++length;

}

for (j = 0; s2[j] != '\0'; ++j, ++length) {
```

```
s1[length] = s2[j];

}

s1[length] = '\0';

printf("After concatenation: ");

puts(s1);

return 0;

}
```

OUTPUT:

After concatenation: Hello World

//WITH FUNC

```
#include <stdio.h>

#include <string.h>

int main(){

char str[100], str2[100];

printf("Enter the first string\n");

gets(str);

printf("Enter the second string\n");

gets(str2);

strcat(str,str2);

printf("String obtained on concatenation is %s\n",str);

return 0;

}
```

OUTPUT:

Enter the first string

HELLO

Enter the second string

WORLD

String obtained on concatenation is HELLOWORLD

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

//WITHOUT FUNC

```
#include<stdio.h>

#include<string.h>

int main() {

char str[100], temp;

int i, j = 0;

printf("Enter the string: ");

gets(str);

i = 0;

j = strlen(str) - 1;

while (i < j) {

temp = str[i];

str[i] = str[j];

str[j] = temp;

i++;

j--;

}

printf("\nReverse string is :%s", str);
```

```
return 0;

}
```

OUTPUT:

```
Enter the string: HELLO HI

Reverse string is :IH OLLEH
```

//WITH FUNC

```
#include <stdio.h>

#include <string.h>

int main()

{

char s[100];

printf("Enter a string to reverse ");

gets(s);

strrev(s);

printf("Reverse of the string: %s\n", s);

return 0;

}
```

OUTPUT:

```
Enter the string: HI BYE

Reverse string is :EYB IH
```

6. perform Substring Extraction (With and Without String Handling

Functions).

//WITHOUT FUNC

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

```
int main(){
char str[100], sstr[100];
int pos, l, c = 0;
printf("Input the string : ");
fgets(str, sizeof str, stdin);
printf("Input the position to start extraction :");
scanf("%d", &pos);
printf("Input the length of substring :");
scanf("%d", &l);
while (c < l)
{
sstr[c] = str[pos+c-1];
c++;
}
sstr[c] = '\0';
printf(sstr);
}
```

OUTPUT:

```
Input the string : HELLOWORLD
Input the position to start extraction :5
Input the length of substring :6
OWORLD
```

7. copy one string into another and count the no of elements copied.

(With and Without String Handling Functions).

```
#include<stdio.h>

//#define N 10

int main()

{

char str1[80], str2[80];

int i;

printf("Input a string: ");

scanf("%s", str2);

for(i=0; str2[i]!='\0'; i++)

str1[i]=str2[i];

str1[i]='\0';

printf("\n");

printf("Original string: %s", str1);

printf("\nNumber of characters = %d\n", i);

return 0;

}
```

OUTPUT:

```
Input a string: Shalinee

Original string: Shalinee

Number of characters = 8
```

//WITH FUNC

```
#include<stdio.h>

#include<string.h>

int main(){
```

```
char str1[100];

char str2[100];

int i;

printf("Enter the string: ");

gets(str2);

strcpy(str1,str2);

printf("\nThe copied string is: %s", str1);

for(i=0; str2[i]!='\0'; i++)

str1[i]=str2[i];

str1[i]='\0';

printf("\nNumber of characters = %d\n", i);

return 0;

}
```

OUTPUT:

```
Enter the string: Shalinee

The copied string is: Shalinee

Number of characters = 8
```

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

```
#include <stdio.h>

int main()

{

 char s[1000];

 int i,n,c=0;

 printf("Enter the string : ");
```

```
 gets(s);
 n=strlen(s);
 for(i=0;i<n/2;i++)
 {
 if(s[i]==s[n-i-1])
 c++;
}
if(c==i)
 printf("string is palindrome");
 else
 printf("string is not palindrome");
return 0;
}
```

OUTPUT:

```
Enter the string : wow
string is palindrome
```

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

```
#include <stdio.h>
int main()
{
 char s[1000],w[1000];
 int n,a[1000],i,j,k=0,l,found=0,t=0;
 printf("Enter the string : ");
```

```
 gets(s);
 printf("Enter word to be searched: ");
 gets(w);
 for(i=0;s[i];i++)
 {
 if(s[i]==' ')
 {
 a[k++]=i;
}
}
a[k++]=i;
j=0;
for(i=0;i<k;i++)
{
n=a[i]-j;
if(n==strlen(w))
{
t=0;
for(l=0;w[l];l++)
{
if(s[l+j]==w[l])
{
t++;
}
```

```
}
if(t==strlen(w))
 {
found++;
 }
}
j=a[i]+1;
}
printf("word '%s' is occurred count=%d ",w,found);


return 0;
}
```

OUTPUT:

```
Enter the string : The crazy man is crazy about her
Enter word to be searched: crazy
word 'crazy' is occurred count=2
```

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

```
#include <stdio.h>
#include <string.h>
int main ()
{
char string[100];
 printf("Enter the string : ");
scanf("%s",string);
```

```
char temp;
int i, j;
int n = strlen(string);
for (i = 0; i < n-1; i++) {
for (j = i+1; j < n; j++) {
if (string[i] > string[j]) {
temp = string[i];
string[i] = string[j];
string[j] = temp;
}
}
}
printf("The sorted string is : %s", string);
return 0;
}
```

OUTPUT:

```
Enter the string : APPLE
The sorted string is : AELPP
```

11. Print the Words Ending with Letter S

```
#include <stdio.h>
 #include <string.h>
 char str[100];
 void main()
 {
```

```
int x, t, j, len;
printf("Enter a string : ");
scanf("%[^\n]s", str);
len = strlen(str);
str[len] = ' ';
for (t = 0, x = 0; x < strlen(str); x++)
{
if ((str[x] == ' ') && (str[x - 1] == 's'))
{
for (j = t; j < x; j++)
printf("%c", str[j]);
t = x + 1;
printf("\n");
}
else
{
if (str[x] == ' ')
{
t = x + 1;
}
}
}
}
```

OUTPUT:

```
Enter a string : The class is full of students

class

is

students
```

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

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main ()

{

char str[100], word[100], twoD[10][30];

int i = 0, j = 0, k = 0, len1 = 0, len2 = 0, l = 0;

printf ("Enter the string:");

gets (str);

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

{

if (str[i] == ' ')

{

twoD[k][j] = '\0';

k ++;

j = 0;

}

else

{
```

```
twoD[k][j] = str[i];
j ++;
}
}
twoD[k][j] = '\0';
j = 0;
for (i = 0; i < k; i++)
{
int present = 0;
for (l = 1; l < k + 1; l++)
{
if (twoD[l][j] == '\0' || l == i)
{
continue;
}
if (strcmp (twoD[i], twoD[l]) == 0) {
twoD[l][j] = '\0';
present = present + 1;
}
}
}
j = 0;
for (i = 0; i < k + 1; i++)
{
```

```
if (twoD[i][j] == '\0')

continue;

else

printf ("%s ", twoD[i]);

}

printf ("\n");

return 0;

}
```

OUTPUT:

```
Enter the string:apple orange banana apple orange

apple orange banana
```