**1)Read from a terminal using scanf function and print using printf function.**

#include <stdio.h>

int main()

{

int input;

printf("enter the value of input: \n");

scanf("%d", &input);

printf("you entred: %d", input);

return 0;

}

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

#include <stdio.h>

#include<string.h>

int main()

{

char name[50];

printf("enter your name: \n");

gets(name);

printf("your name is: ");

puts(name);

return 0;

}

**3) convert :**

**a. Upper case to Lower case**

**b. Lower case to Upper case**

**c. Toggle case**

**d. Sentence case**

1. Upper case to Lower case

#include <stdio.h>

#include <string.h>

int main()

{

char name[50];

char ch;

int j=0;

printf("Enter your text : ");

gets(name);

while(name[j]){

ch=name[j];

putchar(tolower(ch));

j++;

}

return 0;

}

1. To upper

#include <stdio.h>

#include <string.h>

int main()

{

char name[50];

char ch;

int j=0;

printf("Enter your text : ");

gets(name);

while(name[j]){

ch=name[j];

putchar(toupper(ch));

j++;

}

return 0;

}

1. Toggle case

#include <stdio.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("\nThe given string after toggle case is: %s", str);

return 0;

}

1. Sentence Case

#include <stdio.h>

int main()

{

char str[100];

printf("enter any string: ");

gets(str);

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

printf("\nThe given string is: %s", str);

return 0;

}

**4) String concatenation :**

**With String handling Funtion**

#include <stdio.h>

#include <string.h>

#define MAX\_SIZE 100

int main()

{

char f\_name[MAX\_SIZE], l\_name[MAX\_SIZE];

printf("Enter first name: ");

gets(f\_name);

printf("Enter last name: ");

gets(l\_name);

strcat(f\_name, l\_name);

printf("Concatenated string = %s", f\_name);

return 0;

}

**Without string handling function:**

#include <stdio.h>

#include <string.h>

int main()

{

char f\_name[50], l\_name[50], res[50];

int i,j;

printf("Enter your first name : ");

gets(f\_name);

printf("Enter your last name : ");

gets(l\_name);

i=0;

while(f\_name[i] != '\0')

{

i++;

}

j = 0;

while(l\_name[j] != '\0')

{

f\_name[i] = l\_name[j];

i++;

j++;

}

f\_name[i] = '\0';

printf("Concatenated string = %s", f\_name);

return 0;

}

**5) String reversal:**

**With string handling function:**

#include <stdio.h>

#include <string.h>

int main()

{

char s[100];

printf("Enter a string to reverse\n");

gets(s);

strrev(s);

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

return 0;

}

**Without string handling function:**

#include <stdio.h>

int main()

{

char a[1000], b[1000];

int begin, end, count = 0;

printf("Input a string\n");

gets(a);

while(a[count] != '\0'){

count++;

}

end = count-1;

for(begin =0; begin<count; begin++){

b[begin] = a[end];

end--;

}

b[begin] = '\0';

printf("%s\n", b);

return 0;

}

6) **Perform substring extraction**.

Without string handling function

#include <stdio.h>

int main()

{

char str[100], sub\_str[100];

int pos, l, c=0;

printf("\n extract substring: ");

printf(" \n input the string: ");

fgets(str, sizeof str, stdin);

printf("input the position ");

scanf("%d", &pos);

printf("\n enter the length of substring: ");

scanf("%d", &l);

while(c<l){

sub\_str[c] = str[pos+c-1];

c++;

}

sub\_str[c] = '\0';

printf("\n the substring is: %s", sub\_str);

return 0;

}

With string handling function

#include<stdio.h>

#include <string.h>

int main() {

char string[50];

printf("enter a string: \n");

gets(string);

char \* token = strtok(string, " ");

while( token != NULL ) {

printf( " %s\n", token);

token = strtok(NULL, " ");

}

return 0;

}

**7)** **Copy one string into another and count the no of elements copied. With** String Handling Functions

#include<stdio.h>

#include <string.h>

int main() {

char string[50], string2[50];

int i,j,t;

printf("enter a string\n");

gets(string2);

for(i=0; string2[i] != '\0'; i++){

string[i] = string2[i];

string[i] = '\0';

printf("\n");

}

printf("Original string: %s", string2);

printf("\nthe number of characters copied: %d", i);

return 0;

}

Without string handling function

#include <stdio.h>

#include <string.h>

int main()

{

char source[1000], destination[1000];

printf("Input a string\n");

gets(source);

strcpy(destination, source);

int count;

count = copy\_string(source,destination);

printf("Source string: %s\n", source);

printf("Destination string: %s\n", destination);

printf("Number of characters copied: %d", count);

return 0;

}

int copy\_string(char \*source, char \*destination){

int i=0;

while(source[i] != '\0'){

destination[i] = source[i];

i++;

}

destination[i] = '\0';

return i;

}

**8) Write a program to check if a string is palindrome or not.**

#include <stdio.h>

#include <string.h>

int main()

{

char str[50];

int i=0, n;

printf("enter a string: ");

gets(str);

n = strlen(str) - 1;

while(n>0){

if(str[i++] != str[n--]){

printf("\nnot palindrome!");

return;

}

}

printf("\nthe string is palindrome");

return 0;

}

**9) Read a line of text and count all occurrences of particular word.**

#include<stdio.h>

#include <string.h>

int main() {

char string[50], word[50];

int count;

printf("enter a string: \n");

gets(string);

printf("enter a the number to search: \n");

gets(word);

count = countOccurance(string, word);

printf("the %s occured %d times", word, count);

return 0;

}

int countOccurance(char \* string, char \* word){

int i, j, found, count;

int stringlen, wordlen;

stringlen = strlen(string);

wordlen = strlen(word);

count = 0;

for(i=0; i< stringlen - wordlen; i++){

found = 1;

for(j=0; j<wordlen; j++){

if(string[i+j] != word[j]){

found = 0;

break;

}

}

if(found == 1){

count++;

}

}

return count;

}

**10) Read a string and rewrite it in the alphabetical order.**

#include<stdio.h>

#include <string.h>

int main() {

char string[50], temp;

int i,j;

printf("enter a string\n");

gets(string);

for(i=0; string[i]; i++){

for(j=i+1; j<string[j]; j++){

if(string[j] < string[i]){

temp = string[j];

string[j] = string[i];

string[i] = temp;

}

}

}

printf("\nthe string:%s ",string);

return 0;

}

**11)** **Print the Words Ending with Letter S**

#include<stdio.h>

#include <string.h>

int main() {

char string[50], temp;

int i,j,t,l;

printf("enter a string\n");

gets(string);

l = strlen(string);

string[l] = ' ';

for(i=0, t=0; i<l; i++){

if((string[i]==' ') && string[i-1]=='s'){

for(j=t; j<i; j++){

printf("%c", string[j]);

}

t = i+1;

printf("\n");

}

else{

if(string[i] == ' '){

t= i+1;

}

}

}

return 0;

}

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

#include<stdio.h>

#include <string.h>

int main() {

char string[50], word[50], twoD[20][20];

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

printf("enter a string\n");

gets(string);

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

{

if (string[i] == ' ')

{

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

k ++;

j = 0;

}

else

{

twoD[k][j] = string[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;

}

}

if (present > 0)

{

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

}

}

j = 0;

for (i = 0; i < k + 1; i++)

{

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

continue;

else

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

}

printf ("\n");

return 0;

}