11. Write a C program to find the length of the given string using string library functions. INPUT & OUTPUT FORMAT: Input consists of 1 string. Output print the length of the string.

#include <stdio.h>

#include <string.h>

void main()

{

char str1[50];

int i, l = 0;

scanf("%s", str1);

for (i = 0; str1[i] != '\0'; i++)

{

l++;

}

printf("the length of %s is : %d\n\n", str1, l);

}

10.

Write a C program to sort the given string. INPUT & OUTPUT FORMAT: Input consists of 1 string. Output print the ascending order of given string.

#include <stdio.h>

#include <string.h>

void main()

{

char str[100],ch;

int i,j,l;

fgets(str, sizeof str, stdin);

l=strlen(str);

/\* sorting process \*/

for(i=1;i<l;i++)

for(j=0;j<l-i;j++)

if(str[j]>str[j+1])

{

ch=str[j];

str[j] = str[j+1];

str[j+1]=ch;

}

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

}

9. **use gets and puts**

Write a C program to scan the string from the user and to print it by using gets( ) and puts( ). INPUT FORMAT: Input consists of 1 string.

#include<stdio.h>

#include <string.h>

int main(){

char name[50];

gets(name);

printf("The string is ");

puts(name);

return 0;

}

8. **String-count the vowels**

Write a C program to count the number of vowels in the given string. INPUT & OUTPUT FORMAT: Input consists of 1 string. Output print the number of Vowels.

#include <stdio.h>

int main()

{

int c = 0, count = 0;

char s[1000];

gets(s);

while (s[c] != '\0') {

if (s[c] == 'a' || s[c] == 'A' || s[c] == 'e' || s[c] == 'E' || s[c] == 'i' || s[c] == 'I' || s[c] =='o' || s[c]=='O' || s[c] == 'u' || s[c] == 'U')

count++;

c++;

}

printf("Number of vowels in the string: %d", count);

return 0;

}

7. **String palindrome**

Write a C program to find whether the given string is a palindrome or not without using string library functions. INPUT & OUTPUT FORMAT: Input consists of 1 string. If the given string is a Palindrome display “Palindrome”, else display “Not a Palindrome”.

#include <stdio.h>

#include <string.h>

int main(){

char string1[20];

int i, length;

int flag = 0;

scanf("%s", string1);

length = strlen(string1);

for(i=0;i < length ;i++){

if(string1[i] != string1[length-i-1]){

flag = 1;

break;

}

}

if (flag) {

printf(" not a palindrome", string1);

}

else {

printf(" palindrome", string1);

}

return 0;

}

6. Write a C program to change the given string to lowercase without using string library functions. INPUT & OUTPUT FORMAT: Input consists of 1 string. Output print the Lowercase of given string.

#include <stdio.h>

#include <string.h>

int main()

{

char Str1[100];

int i;

gets(Str1);

for (i = 0; Str1[i]!='\0'; i++)

{

if(Str1[i] >= 'A' && Str1[i] <= 'Z')

{

Str1[i] = Str1[i] + 32;

}

}

printf("\n String with Lower Case = %s", Str1);

return 0;

}

5. **String uppercase**

Write a C program to change the given string to uppercase without using string library functions. INPUT & OUTPUT FORMAT: Input consists of 1 string. Output print the Uppercase of given string.

#include <stdio.h>

#include <string.h>

int main() {

char s[100];

int i;

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;

}

4. Write a C program to find the concatenation of the given two strings using string library functions. INPUT FORMAT: Input consists of 2 strings.

#include <stdio.h>

#include <string.h>

int main()

{

char s1[10] ;

char s2[10];

scanf("%s",s1);

scanf("%s",s2);

strcat(s1,s2);

printf("The concatenated string is: %s", s1);

return 0;

}

3. **String compare**

Write a C program to find whether the given two strings are the same or not using string library functions. INPUT & OUTPUT FORMAT: Input consists of 2 strings. If two strings are same, display “Strings are same”, else display “Strings are not same”.

#include<stdio.h>

#include <string.h>

int main(){

char str1[20],str2[20];

gets(str1);//reads string from console

gets(str2);

if(strcmp(str1,str2)==0)

printf("Strings are equal");

else

printf("Strings are not equal");

return 0;

}

2. **String copy**

Write a C program to copy a string from one variable to other using string library functions. INPUT FORMAT: Input consists of 1 string.

#include<stdio.h>

int main(){

char ch[20];

scanf("%s",ch);

printf("The copied string is: %s",ch);

return 0;

}

1. **Print the string**

Write a C program to scan a string from the user and to print it. INPUT FORMAT: Input consists of 1 string.

#include<stdio.h>

int main()

{

char str[20];

scanf("%s", str);

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

return 0;

}