**CA3001 – Programming and Data Structures using C**

**Assignment 8 - 19.01.2021**

**Q1.** Read from a terminal using scanf function and print using printf function.

Ans – C Program & Output:

#include<stdio.h>

#include <string.h>

int main()

{

char name[20];

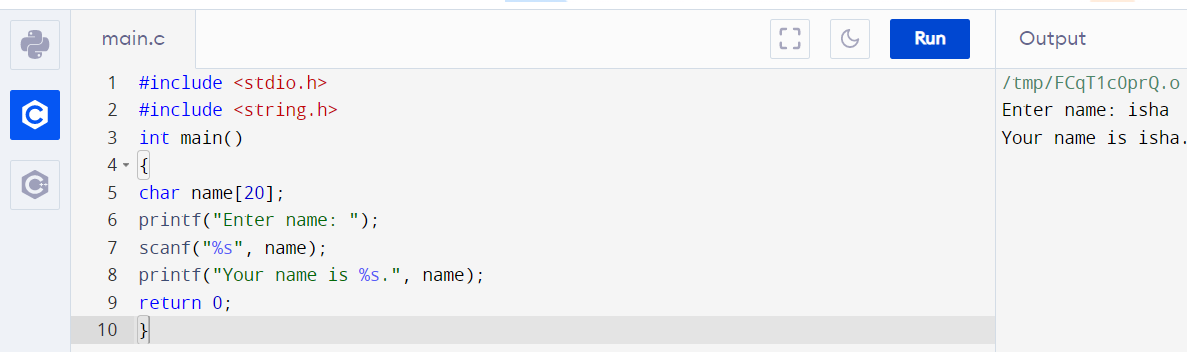
printf("Enter name: ");

scanf("%s", name);

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

return 0;

}



**Q2.** Read a lines of text from a terminal using fgets function and print using puts

function.

Ans – C Program & Output:

#include <stdio.h>

#include <string.h>

int main()

{

char name[30];

printf("Enter name: ");

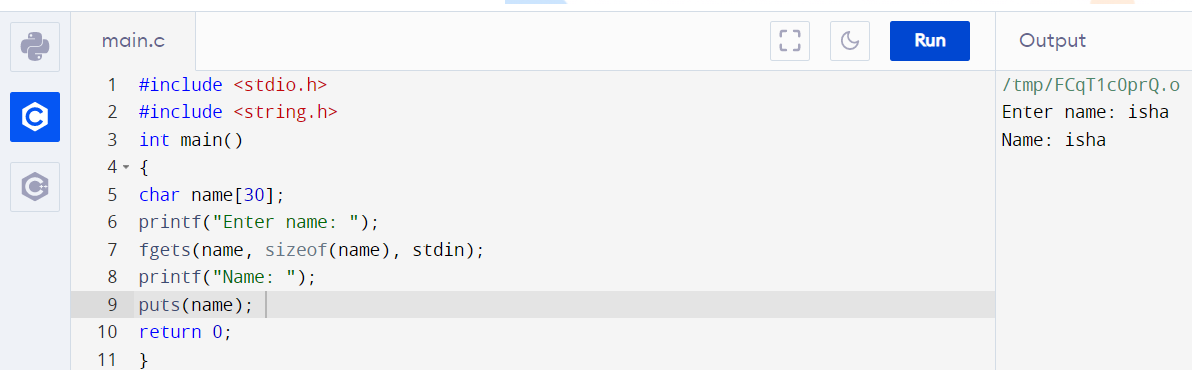
fgets(name, sizeof(name), stdin);

printf("Name: ");

puts(name);

return 0;

}



**Q3.** Convert

a. Upper case to Lower case

b. Lower case to Upper case

c. Toggle case

d. Sentence case

Ans – C Program & Output:

* a. Upper case to Lower case

#include<stdio.h>

#include <string.h>

int main(){

char str[20];

printf("Enter string: ");

gets(str);

printf("String is: %s",str);

printf("\nLower String is: %s",strlwr(str));

return 0;

}

**Q4.** Perform String Concatenation (With and Without String Handling Functions).

Ans – C Program & Output:

* With String handling Function :

#include<stdio.h>

int main()

{

char str1[25],str2[25];

int i=0,j=0;

printf("\nEnter First String:");

gets(str1);

printf("\nEnter Second String:");

gets(str2);

while(str1[i]!='\0')

i++;

while(str2[j]!='\0')

{

str1[i]=str2[j];

j++;

i++;

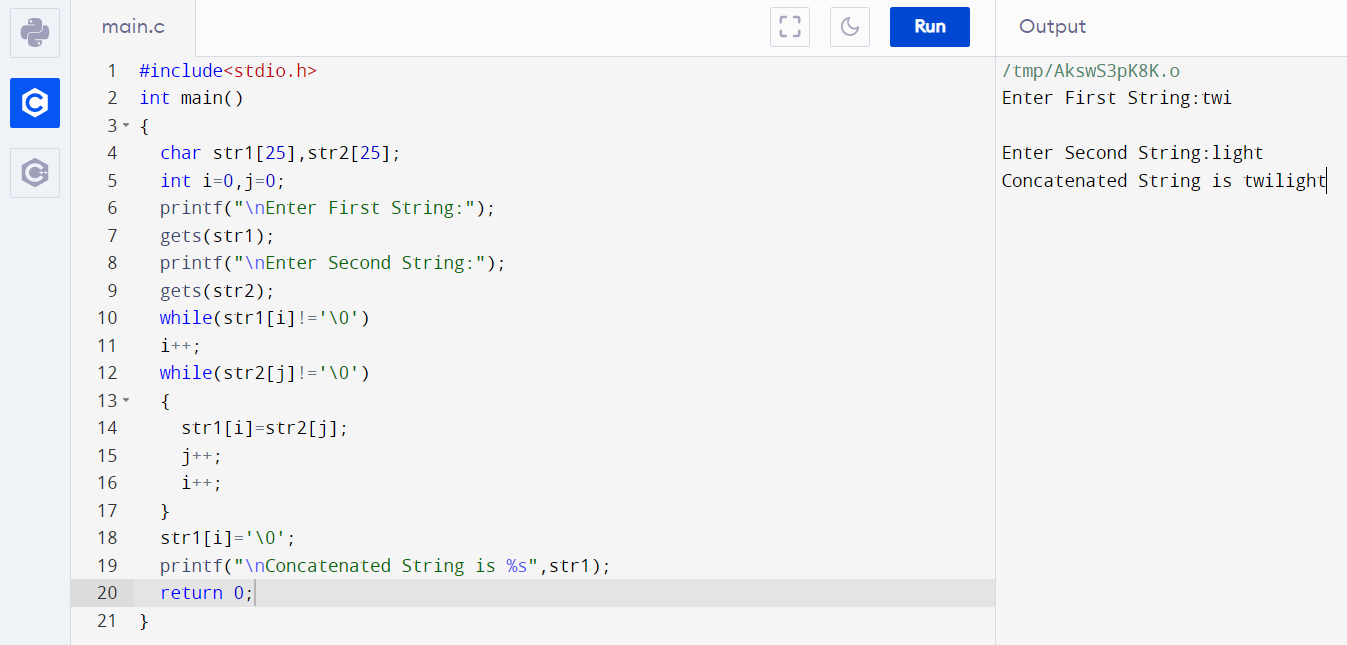
}

str1[i]='\0';

printf("\nConcatenated String is %s",str1);

return 0;

}



* Without String handling function:

#include<stdio.h>

#include<string.h>

void concat(char[], char[]);

int main() {

char s1[50], s2[30];

printf("\nEnter String 1 :");

gets(s1);

printf("\nEnter String 2 :");

gets(s2);

concat(s1, s2);

printf("\nConcated string is :%s", s1);

return (0);

}

void concat(char s1[], char s2[]) {

int i, j;

i = strlen(s1);

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

s1[i] = s2[j];

}

s1[i] = '\0';

}



**Q5.** Perform String Reversal (With and Without String Handling Functions).

Ans – C Program & Output:

* With string handling function:

#include<stdio.h>

#include <string.h>

int main(){

char str[20];

printf("Enter string: ");

gets(str);

printf("String is: %s",str);

printf("\nReverse String is: %s",strrev(str));

return 0;

}

**Q7.** Copy one string into another and count the no of elements copied. (With and Without String Handling Functions).

Ans – C Program & Output:

* With string handling function:

#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;

}



* Without using string handling functions:

#include <stdio.h>

int main()

{

char text1[100];

char text2[100];

int i;

printf("Enter any string: ");

gets(text1);

for(i=0; text1[i]!='\0'; i++)

{

text2[i] = text1[i];

}

text2[i] = '\0';

printf("First string = %s\n", text1);

printf("Second string = %s\n", text2);

printf("Total characters copied = %d\n", i);

return 0;

}



**Q8.** Read a string and prints if it is a palindrome or not.

Ans – C Program & Output:

#include <stdio.h>

#include <string.h>

int main()

{

char string1[20];

int i, length;

int flag = 0;

printf("Enter a string:");

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("%s is not a palindrome", string1);

}

else {

printf("%s is a palindrome", string1);

}

return 0;}



**Q9.** Read a line of text and count all occurrences of particular word.

Ans – C Program & Output:

#include <string.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+

}

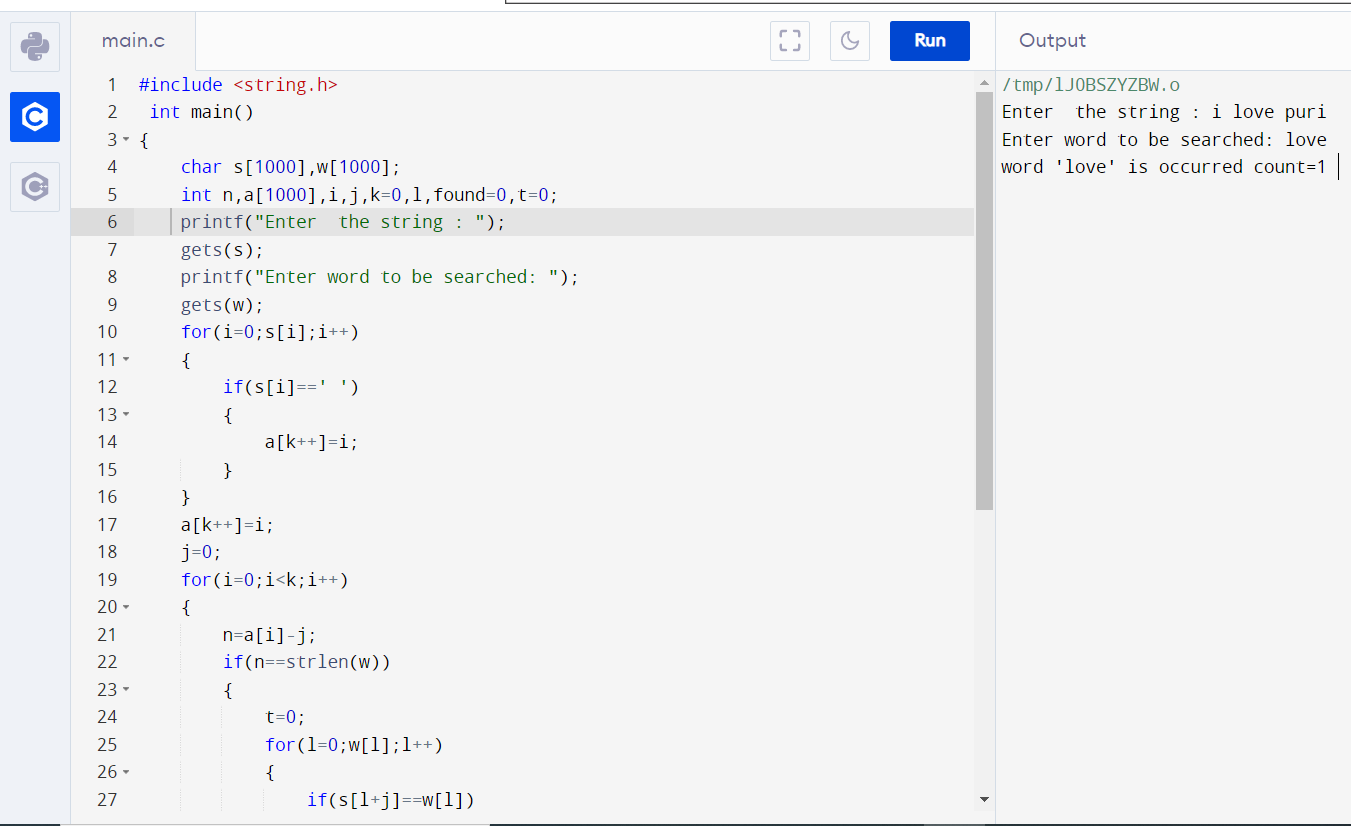
}

j=a[i]+1;

}

printf("word '%s' is occurred count=%d ",w,found);

}



**Q10.** Read a string and rewrite it in the alphabetical order.

Ans – C Program & Output:

#include <stdio.h>

#include <string.h>

void main()

{

char str[100],ch;

int i,j,l;

printf("Input the string : ");

fgets(str, sizeof str, stdin);

l=strlen(str);

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("After sorting the string appears like : \n");

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

}



**Q12.** Delete All Repeated Words in the line of text.

Ans – C Program & Output:

#include <stdio.h>

#include <string.h>

int main()

{

char str[100];

int i, j, k;

printf("\n Please Enter any String : ");

gets(str);

for(i = 0; i < strlen(str); i++)

{

for(j = i + 1; str[j] != '\0'; j++)

{

if(str[j] == str[i])

{

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

{

str[k] = str[k + 1];

}

}

}

}

printf("\n The Final String after Removing All Duplicates = %s ", str);

return 0;

}

