**“Array and String”**

**ASSIGNMENT**

**IV**

**Prepared by:** Srijal Dangol **Submitted To:** Ashim Sir

**Roll Number:** 17 **Subject:** C Programming

**Shift:** Morning

**BscCSIT079**

**###Theoretical Assignment###**

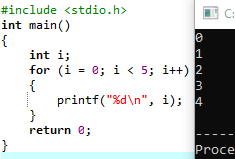
1. **Explain in detail about storage class. Write in detail about 4 different storage class.**

Storage Classes are used to describe the features of a variable/function. These features basically include the scope, visibility and life-time which help us to trace the existence of a particular variable during the runtime of a program.

C language uses 4 storage classes. They are:

* auto: This is the default storage class for all the variables declared inside a function or a block. Hence, the keyword auto is rarely used while writing programs in C language. Auto variables can be only accessed within the block/function they have been declared and not outside them (which defines their scope). Of course, these can be accessed within nested blocks within the parent block/function in which the auto variable was declared. However, they can be accessed outside their scope as well using the concept of pointers given here by pointing to the very exact memory location where the variables reside. They are assigned a garbage value by default whenever they are declared.
* extern: Extern storage class simply tells us that the variable is defined elsewhere and not within the same block where it is used. Basically, the value is assigned to it in a different block and this can be overwritten/changed in a different block as well. So an extern variable is nothing but a global variable initialized with a legal value where it is declared in order to be used elsewhere. It can be accessed within any function/block. Also, a normal global variable can be made extern as well by placing the ‘extern’ keyword before its declaration/definition in any function/block. This basically signifies that we are not initializing a new variable but instead we are using/accessing the global variable only. The main purpose of using extern variables is that they can be accessed between two different files which are part of a large program. For more information on how extern variables work, have a look at this link.
* static: This storage class is used to declare static variables which are popularly used while writing programs in C language. Static variables have the property of preserving their value even after they are out of their scope! Hence, static variables preserve the value of their last use in their scope. So we can say that they are initialized only once and exist till the termination of the program. Thus, no new memory is allocated because they are not re-declared. Their scope is local to the function to which they were defined. Global static variables can be accessed anywhere in the program. By default, they are assigned the value 0 by the compiler.
* register: This storage class declares register variables that have the same functionality as that of the auto variables. The only difference is that the compiler tries to store these variables in the register of the microprocessor if a free register is available. This makes the use of register variables to be much faster than that of the variables stored in the memory during the runtime of the program. If a free registration is not available, these are then stored in the memory only. Usually few variables which are to be accessed very frequently in a program are declared with the register keyword which improves the running time of the program. An important and interesting point to be noted here is that we cannot obtain the address of a register variable using pointers.

1. **Write one simple example of for-loop. Explain about the steps how it works.**

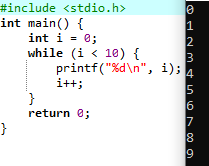


This will print out the numbers 0 to 4. Here’s the steps:

* The variable i is initialized to 0.
* The loop condition i < 5 is checked.
* If the condition is true, the code inside the loop is executed.
* The variable i is incremented by 1.
* The loop condition is checked again.
* This process repeats until the loop condition is false.

So in this example, the loop runs five times, with i taking on the values 0 to 4 in turn.

1. **Write one simple example of while-loop. Explain about the steps how it works.**



This will print out the numbers 0 to 9. Here’s how it works:

* The variable i is initialized to 0.
* The loop condition i < 10 is checked.
* If the condition is true, the code inside the loop is executed.
* The variable i is incremented by 1.
* The loop condition is checked again.
* This process repeats until the loop condition is false.

So in this example, the loop runs ten times, with i taking on the values 0 to 9 in turn.

1. **Write a paragraph and an example of each about strlen(), strcpy(), strcat(), strcmp(), strrev().**

Here are some functions that operate on strings in C:

* strlen() - This function returns the length of a string (i.e., the number of characters in it). Here’s an example:

#include <stdio.h>

#include <string.h>

int main() {

char str[] = "Hello, world!";

int len = strlen(str);

printf("The length of the string \"%s\" is %d.\n", str, len);

return 0;

}

This will output:

The length of the string "Hello, world!" is 13.

* strcpy() - This function copies one string to another. Here’s an example:

#include <stdio.h>

#include <string.h>

int main() {

char src[] = "Hello, world!";

char dest[20];

strcpy(dest, src);

printf("The source string is \"%s\".\n", src);

printf("The destination string is \"%s\".\n", dest);

return 0;

}

This will output:

The source string is "Hello, world!".

The destination string is "Hello, world!".

* strcat() - This function concatenates (i.e., joins together) two strings. Here’s an example:

#include <stdio.h>

#include <string.h>

int main() {

char str1[] = "Hello";

char str2[] = ", world!";

strcat(str1, str2);

printf("The concatenated string is \"%s\".\n", str1);

return 0;

}

This will output:

The concatenated string is "Hello, world!".

* strcmp() - This function compares two strings and returns an integer indicating whether they are equal or which one comes first alphabetically. Here’s an example:

#include <stdio.h>

#include <string.h>

int main() {

char str1[] = "apple";

char str2[] = "banana";

int result = strcmp(str1, str2);

if (result == 0) {

printf("\"%s\" and \"%s\" are equal.\n", str1, str2);

} else if (result < 0) {

printf("\"%s\" comes before \"%s\" alphabetically.\n", str1, str2);

} else {

printf("\"%s\" comes after \"%s\" alphabetically.\n", str1, str2);

}

return 0;

}

This will output:

"apple" comes before "banana" alphabetically.

* strrev() - This function reverses a string. Here’s an example:

#include <stdio.h>

#include <string.h>

int main() {

char str[] = "Hello";

strrev(str);

printf("The reversed string is \"%s\".\n", str);

return 0;

}

This will output:

The reversed string is "olleH".

1. **What is array? How array is different from ordinary variable.**

* [An array in C is a fixed-size collection of similar data items stored in contiguous memory locations](https://www.geeksforgeeks.org/c-arrays/). [It can be used to store the collection of primitive data types such as int, char, float, etc., and also derived and user-defined data types such as pointers, structures, etc.](https://www.geeksforgeeks.org/c-arrays/)

[An array is different from an ordinary variable in that it can store multiple values under a single name](https://www.geeksforgeeks.org/c-arrays/). [This makes it easier to work with large amounts of data because you can access all the elements of an array using a single name](https://www.programiz.com/c-programming/c-arrays).

1. **How can you prepare a student mark sheets software using the things you’ve learned till now? Give me the list of the things you’ve learned and how can you use it to do what kind of operation to prepare a mark sheet of a student.**

* [To create a student mark sheet software using C, I will use structures to store data such as roll number, name, marks of different subjects, and total marks](https://www.youtube.com/watch?v=l1GK7pO3KJc). [I will use loops and conditional statements to calculate the average marks of each student and their grades](https://www.youtube.com/watch?v=l1GK7pO3KJc). I will also use functions such as printf() and scanf() to display and input data respectively.

[Here’s an example of using structures in C to store data for a student mark sheet](https://www.youtube.com/watch?v=l1GK7pO3KJc):

struct student {

int roll\_number;

char name[50];

float marks[5];

float total\_marks;

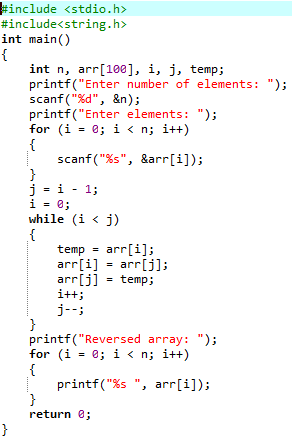
char grade;

};

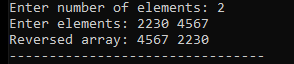
**###Practical Assignment###**

1. **WAP to read n numbers from user and store in an array and then rearrange the array in the reverse order.**

Code:

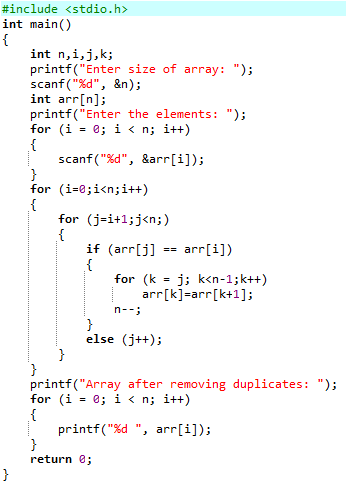


Output:

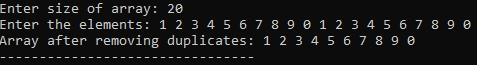


1. **WAP a program to read n numbers in an array and remove the duplicate numbers from the array.**

Code:

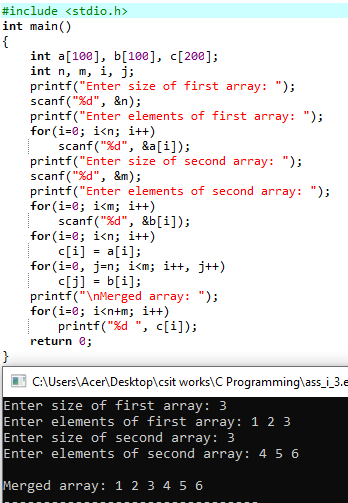


Output:



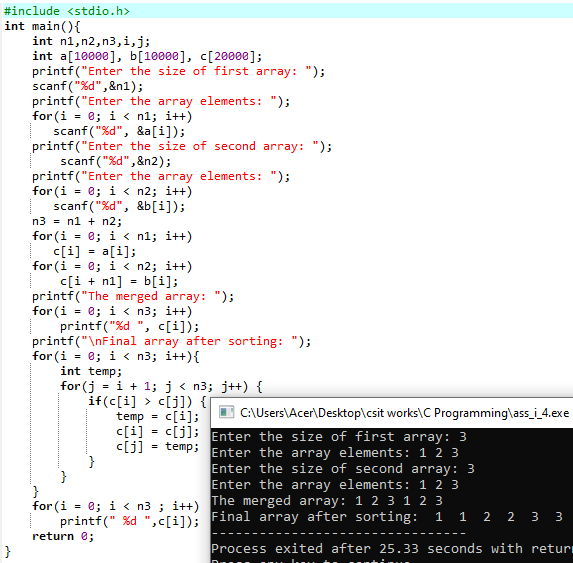
1. **WAP a program to read two arrays and merge these two arrays into third array.**

Code:



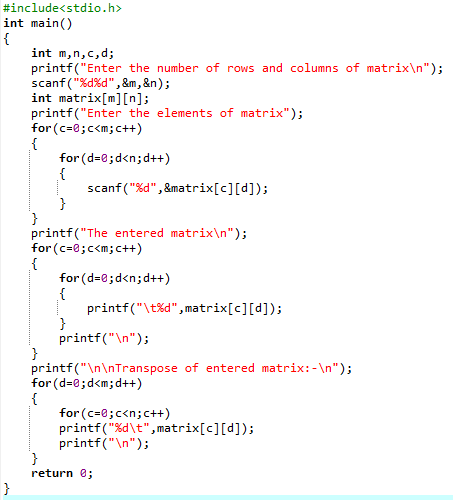
1. **WAP to merge two sorted arrays in another array in a sorted order.**

Code:

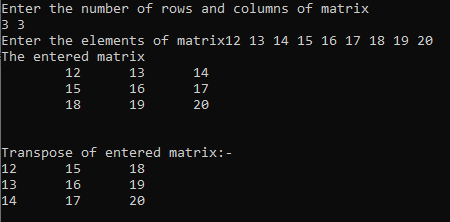


1. **WAP to find transpose of a given Matrix.**

Code:

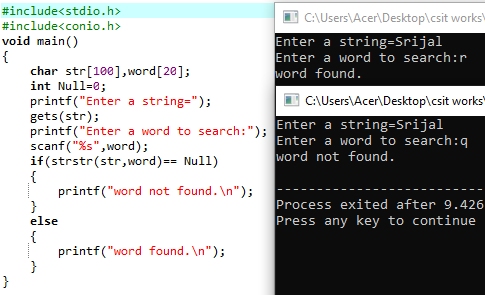


Outputs:



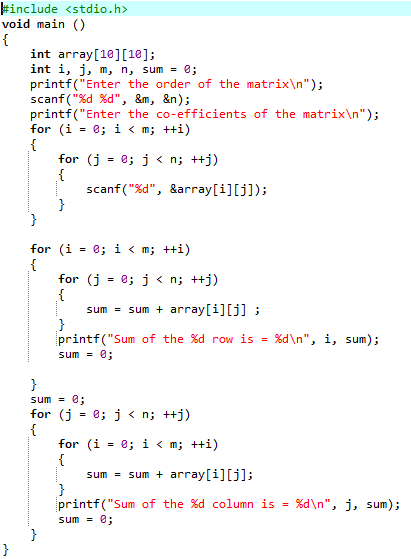
1. **Write a program to read a string and search a specified word in given string. (use function strstr() )**

Code:

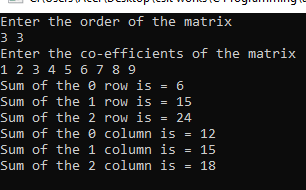


1. **WAP in C to take input M\*N from user and find a row sum and column sum.**

Code:



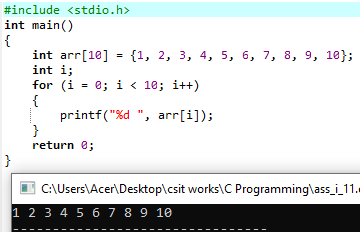
Output:



**###Arrays and Strings###**

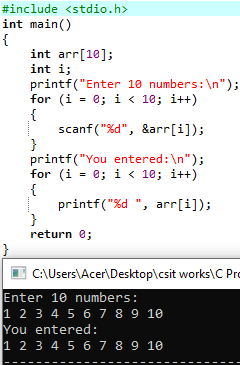
1. **WAP in C to initialize any 10 value in an array and display them.**

Code:



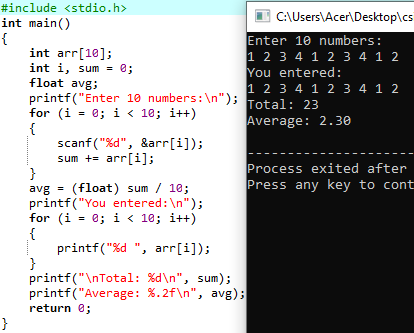
1. **WAP to input any 10 number by user and display them.**

Code:



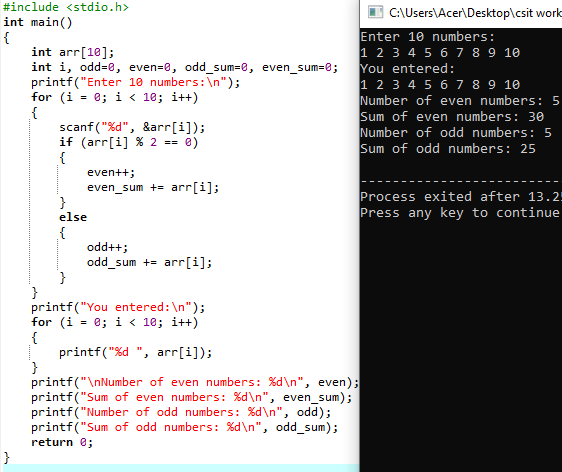
1. **WAP in C to input any 10 number in an array and print them along with the Total and average of that numbers.**

Code:

****

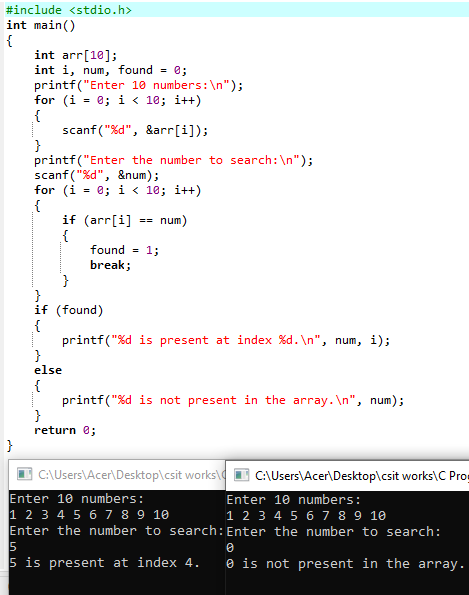
1. **WAP in C to input any 10 numbers in an array and count no. of ODD and EVEN and find out their sum and display them.**

Code:

****

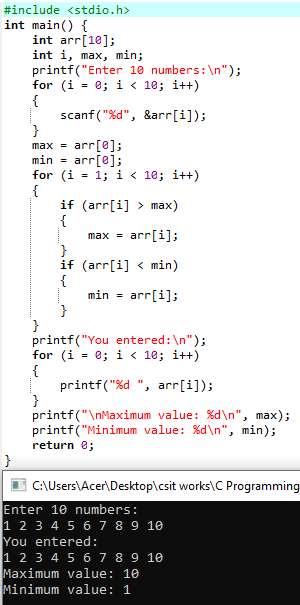
1. **WAP in C to input any 10 numbers in an array and search an element.**

Code:



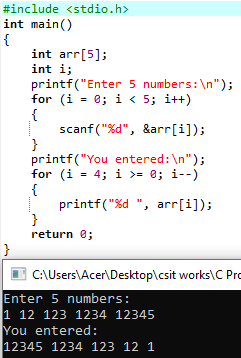
1. **WAP to input any 10 numbers in an array and find out the Maximum and Minimum Value.**

Code:



1. **WAP to input any 5 elements in an array and print them in reverse order too.**

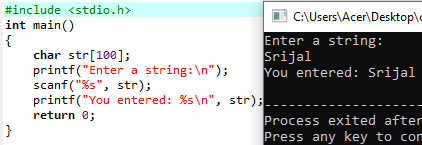
Code:



**###Character array and String###**

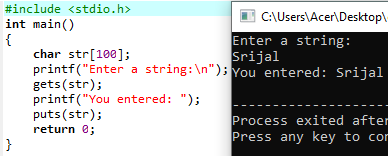
1. **WAP to input and output of String using scanf() and printf().**

Code:



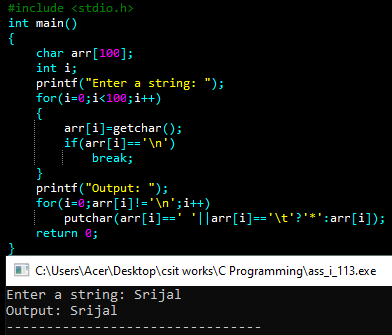
1. **WAP to input and output of String using gets() and puts().**

Code:



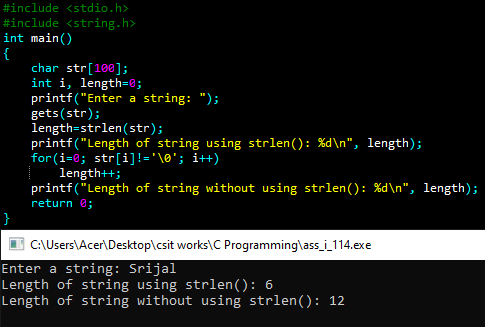
1. **WAP to input and output of Character array using getchar() and putchar().**

Code:



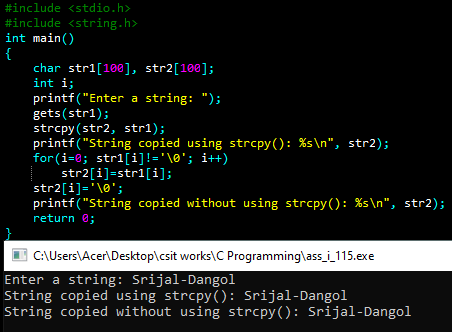
1. **WAP to input a string and find out the length of string using strlen() and without using strlen().**

Code:



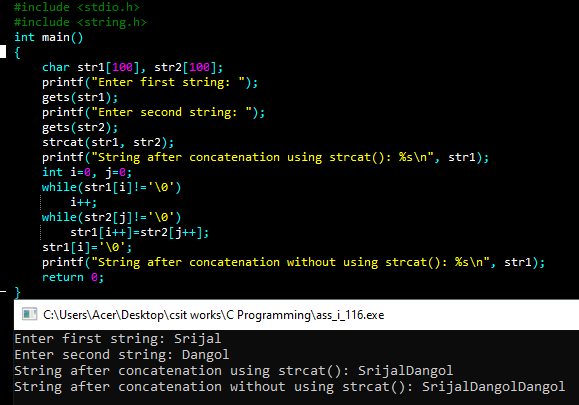
1. **WAP to copy a string from one to another using strcpy() and without using strcpy().**

Code:



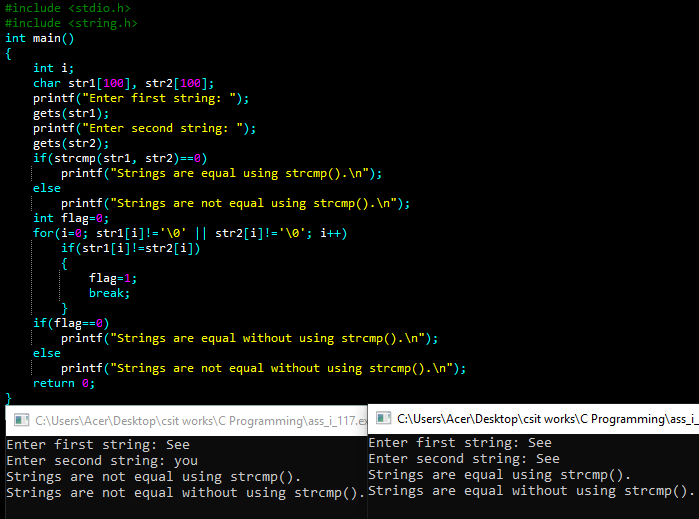
1. **WAP to concatenate two string using strcat() and without using strcat().**

Code:



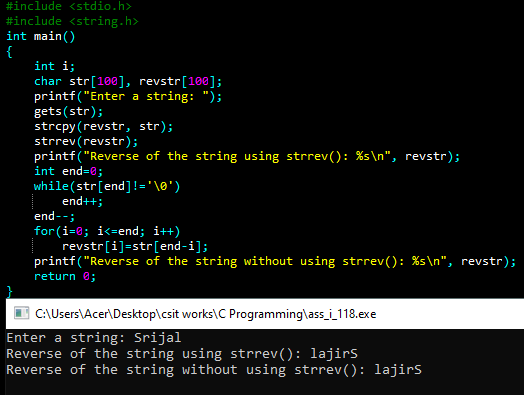
1. **WAP to compare two string using strcmp() and without using Strcmp().**

Code:



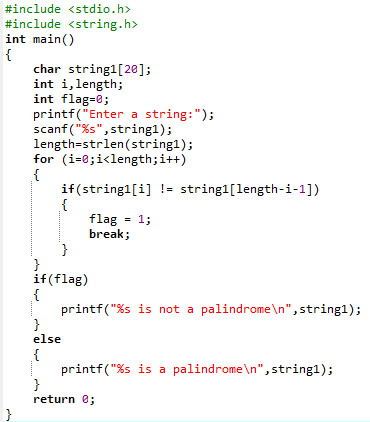
1. **WAP to reverse a string using strrev() and without using strrev().**

Code:



1. **WAP to find whether the given string is palindrome or not.**

Code:

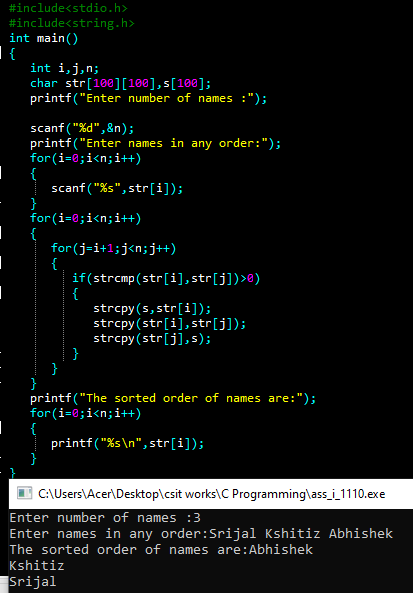


Output:



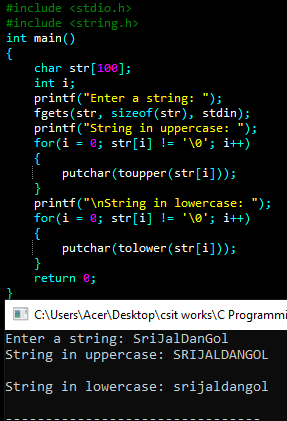
1. **WAP to input the name of any five student and sort them alphabetically in ascending order.**

Code:



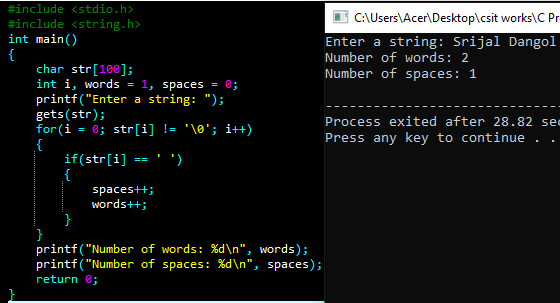
1. **WAP to input a string and convert it into upper case and vice versa.**

Code:



1. **WAP to take the string using gets(), and pass to the function to find and return number of words, whitespace in that string.**

Code:



1. **Write a program to read a string and search a specified word in given string.**

Code:

