**Experiment # 06**

**OBJECTIVE:**

**To be familiar with Strings Handling Functions.**

**THEORY**

Strings are the form of data used in programming languages for storing and manipulating test, such as words, names, and sentences. In C, a string is not a formal data type as it is in some languages. Instead it is a arrays of type char. When you think about it, this makes a good deal of sense, a string is a series of characters, and that a just what an array of type char is.

**Example**

#include<iostream>

using namespace std;

#include<conio.h>

main()

{

char name[20];

cout<<" Enter a name : ";

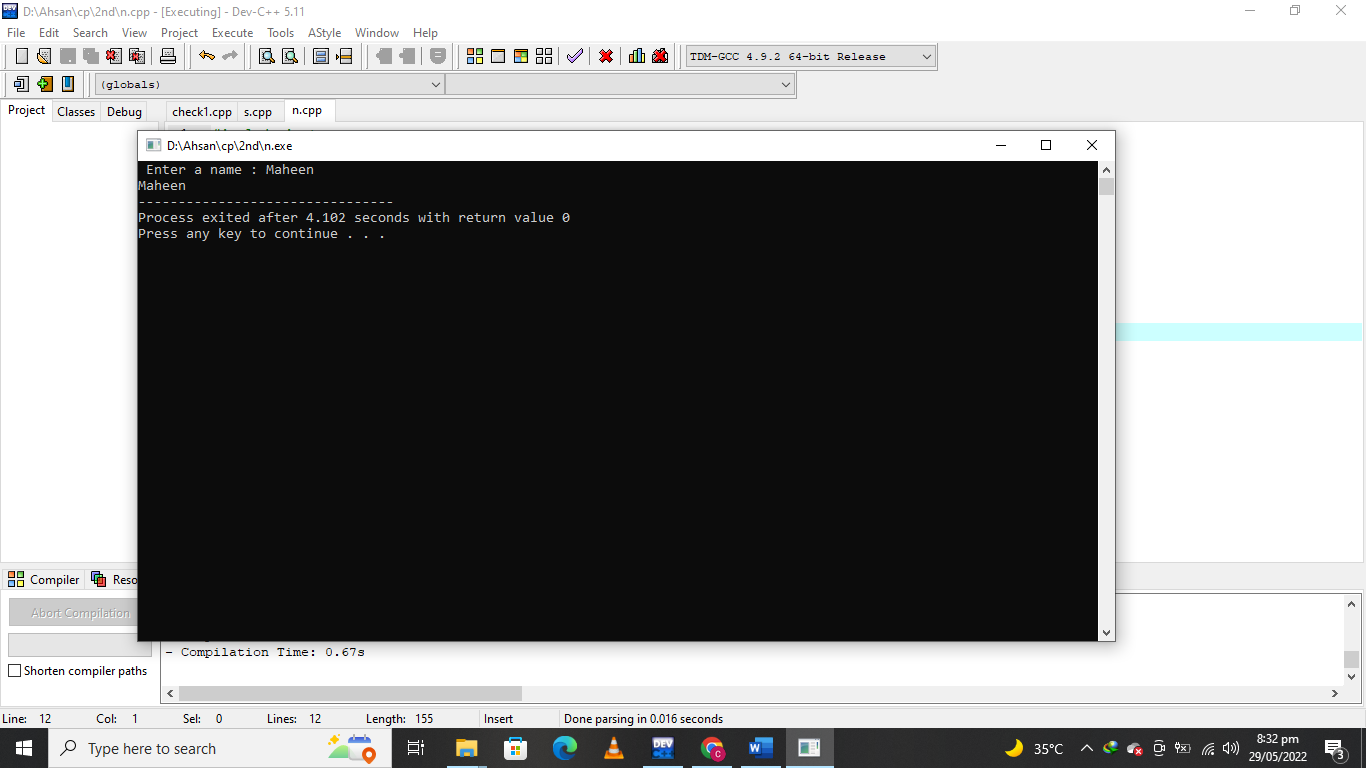
cin>>name;

cout<<name;

getch();

}

**OUTPUT:**



**Initializing Strings**

We can initialize the string as

Char name[10]=”IUB”;

**String Handling Functions**

Strlen()  
strlen() function determines the oength of the String.

**Example**

#include<iostream>

using namespace std;

#include<conio.h>

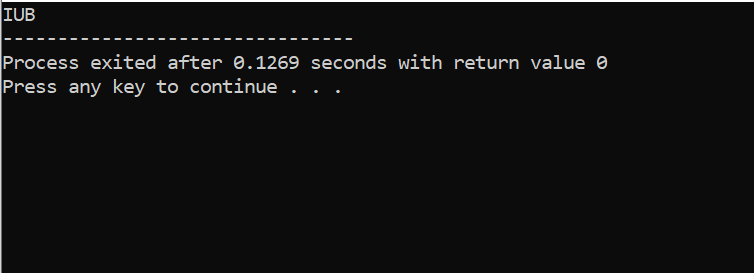
int main()

{

char name[]="IUB";

cout<<(name); }

**OUTPUT:**



**strcat() and strncat()**

the strcat() and strncat() function are used to concatenate two strings, when two strings are concatenate the contents of t he second string are copied to the end of the first string.

Strcat() function concatenate the whole string, where as strrncat() function concatenate n characters of the second string.

**Example (1)**

#include<iostream>

using namespace std;

#include<conio.h>

#include<string.h>

main()

{

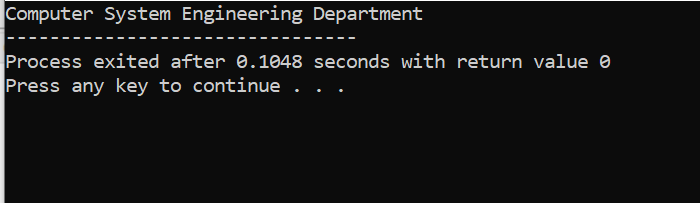
char string1[]="Computer System Engineering ";

char string2[]="Department";

strcat(string1,string2);

cout<<string1;}

**OUTPUT**



**Example (2)**

#include<iostream.h>  
#include<conio.h>  
#include<string.h>

main()  
{  
char string1[]=”Computer Systems ”;

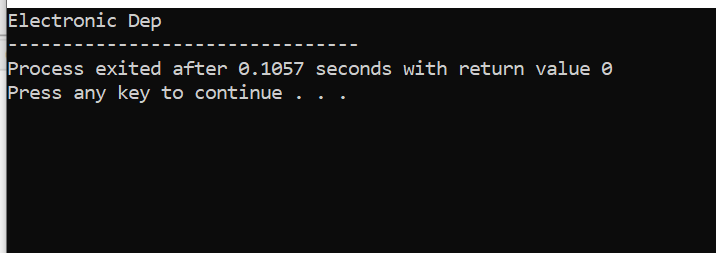
char string2[]=”Department”;

strncat(string1,string2,3);

cout<<string1;

}

**OUTPUT:**



**Strcmp() and strncmp() Functions**

Strcmp() and strncmp() Functions are two functions used to compare two strings. Comparison is performed on character by character basis. Each function return a value based on the result of the comparison. The integer value returned is:  
=0, if the strings are identical

<0, if the first string alphabetically precedes the second.

>0, if the second string alphabetically precedes the first.

Strcmp() function compare all the characters of the first string with all characters of second string. Where strncmp() compares the n no of characters from the strings.

**Example(1)**

#include<iostream>

#include<conio.h>

#include<string.h>

using namespace std;

main()

{

char string1[]="Electronics ";

char string2[]="Electronics";

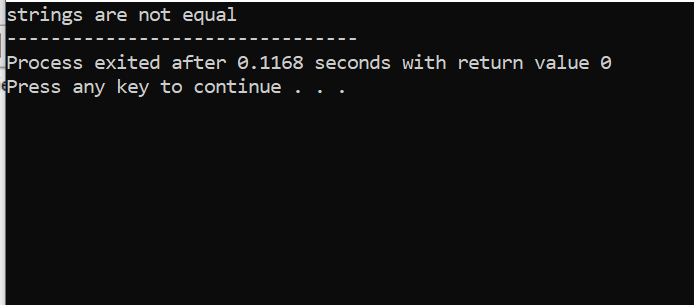
if(strcmp(string1,string2)==0)

cout<<"strings are equal";

else

cout<<"strings are not equal";}

**OUTPUT:**



**Example (2)**

#include<iostream.h>  
#include<conio.h>  
#include<string.h>  
main()  
{  
char string1[]=”Electronics ”;

char string2[]=”Electronic”;

if(strncmp(string1,string2,9)= =0)

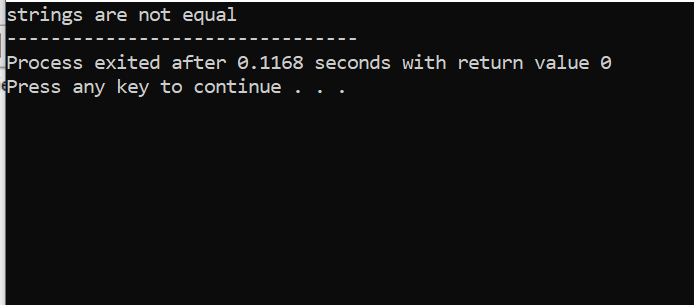
cout<<”strings are equal”;

else

cout<<”strings are not equal”;

}

**OUTPUT:**



**strrev() function**

strrev() function reverse the specified string.

**Example**

#include<iostream.h>  
#include<conio.h>  
#include<string.h>  
{  
char name[]=”NUST”;

cout<<strrev(name);

getch();

}

**OUTPUT:**



**Lab Tasks:**

Q1. Some words are spelled the same forward and backward such as level, radar, etc. are called palindromes. Develop a code in C++ that takes a string argument, returns an int value of 1 if it is palindrome and 0 otherwise. A palindrome can also be “ I am ma I”.

**QUESTION NUMBER : 01:**

Some words are spelled the same forward and backward such as level, radar, etc. are called palindromes. Develop a code in C++ that takes a string argument, returns an int value of 1 if it is palindrome and 0 otherwise. A palindrome can also be “ I am ma I”.

**PROGRAM:**

#include <iostream>

#include<conio.h>

#include <string.h>

using namespace std;

int main()

{

char string1[20];

int i, length;

int flag =0;

cout<<"Enter a string:";

cin>>string1;

length= strlen(string1);

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

{

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

{

flag=1;

break;

}

}

if(flag)

{

cout<<string1<<" is not a palindrome "<<endl;

}

else

{

cout<<string1 << " is palindrome "<<endl;

}

system("pause");

return 0;

}

**OUTPUT:**

