EXPL

Contents

TEST PROGRAMS

Test Program 1 : Bubblesort (iterative)

This test program reads elements into an array and sorts them using the classic bubblesort algorithm. (iterative version)

Input : 1. Number of elements to be sorted from standard input. 2. Elements to be sorted

Output : A sorted array of elements.

This program test the iteration, conditional and arrays.

The code for the test progam can be found here

Test Program 2 : Extended Euclid algorithm (iterative)

This test program calculates the GCD (Greatest Common Divisor) of two numbers along with Bezout co-efficients. (iterative version)

Input : Two numbers

Output : GCD of the given two numbers and Benoud co-efficients.

This program test the iteration, parameter passing, passing of user-defined datatype as return value of function.

The code for the test progam can be found here

Test Program 3 : Bubblesort (recursive)

This test program reads elements into an array and sorts them using the classic bubblesort algorithm. (recursive version)

Input : 1. Number of elements to be sorted from standard input.
2. Elements to be sorted

Output : A sorted array of elements.

This program test the recursion, conditional and arrays.

The code for the test progam can be found here

Test Program 4 : Extended Euclid algorithm (recursive)

This test program calculates the GCD (Greatest Common Divisor) of two numbers along with Bezout co-efficients. (recursive version)

Input : Two numbers

Output : GCD of the given two numbers and Benoud co-efficients.

This program test the recursion, parameter passing, passing of user-defined datatype as return value of function.

The code for the test progam can be found here

Test Program 5 : Factorial (recursive)

This test program calculates and prints out the factorial of the first n numbers.

Input : Value of n read from standard input.

Output : Value of n!.

This program test the working of recursive functions.

The code for the test progam can be found here

Test Program 6 : Quicksort (recursive)

This test program reads elements into an array and sorts them using the classic quicksort algorithm. (recursive version)

Input : 1. Number of elements to be sorted from standard input.
2. Elements to be sorted

Output : A sorted array of elements.

This program test the recursion, conditional and arrays.

The code for the test progam can be found here

Test Program 7

This test program calculates the value of a function f(x) given by :
f(x) = 91 if x>=91;
f(f(x+11)) if x<91
So, the value of the function is always 91 for any given input number.

Input : A number

Output : 91.

This program test the recursion, parameter passing, calling a function inside the call of another function.

The code for the test progam can be found here

Test Program 8

This test program reads elements into a linked list and prints them.

Input : Value of length of list and list of values from standard input.

Output : The list of values stored in linked list.

This program test the working of dynamic memory allocation functions like Initialize(), Alloc() and Free().

The code for the test progam can be found here

Test Program 9 : Binary Search Tree

This test program inserts elements into a binary search tree and prints the elements in inorder, preorder and postorder traversal. The program stops taking input elements (that are to be inserted in Binary Search Tree) when the user enters 0.

Input : The elements that are to be inserted into a Binary Search Tree and a 0 at last (indicating the end of input).

Output : The inorder, preorder and postorder traversal of the tree.

This program test the iteration, recursion, conditional, arrays and passing of user-defined datatype as return value of function.

The code for the test progam can be found here
