Nahrin Sharna

Lab05

02/18/2019

**Program 01:**

**Initial Test Plan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | Invalid | User input a negative value | Key = -2 | Element not found |  |  |
| 2 | Valid | User input a value and it is found | Key = 235 | Element found after 235 iterations |  |  |
| 3 | Invalid | Use enter a key number from which the overflow starts | Key = 4445 | It shows an exception error in the program |  |  |
| 4 | Valid | The last key before the overflow starts | Key = 4444 | Element found after 4444 iteration |  |  |

**Source File:**

/\*

Author: Nahrin Sharna

Creation Date: 02/16/2019

Modification Date: 02/16/2019

\*/

#include<iostream>

using namespace std;

int index = 0;

int recursiveLinearSearch(int array[], int key, int size, bool &methodStatus)

{

if (array[index] == key)

{

methodStatus = true;

return(index);

}

if (key < 0) {

return -1;

}

else

{

index++;

return(1 + recursiveLinearSearch(array, key, size, methodStatus));

}

}

int main()

{

int i, j, key, l, m, n;

int size = 5000;

int array[5000];

int biggest;

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

array[i] = i;

}

cout << "Enter the value to be searched : " << endl;

cin >> key;

bool methodStatus = false;

n = recursiveLinearSearch(array, key, size, methodStatus);

if (n == -1)

cout << "Element not Found" << endl;

else

cout << "Element Found after " << index << " iterations" << endl << endl;

system("pause");

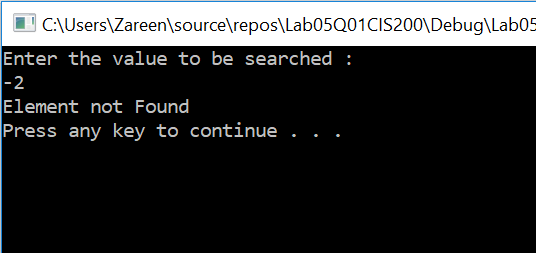
}

**Final Test:**

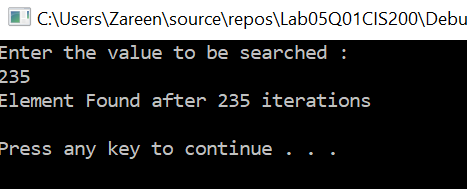
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | Invalid | User input a negative value | Key = -2 | Element not found | Element not found | Pass |
| 2 | Valid | User input a value and it is found | Key = 235 | Element found after 235 iterations | Element found after 235 iterations | Pass |
| 3 | Invalid | Use enter a key number from which the overflow starts | Key = 4445 | It shows an exception error in the program | It shows an exception error in the program | Pass |
| 4 | Valid | The last key before the overflow starts | Key = 4444 | Element found after 4444 iteration | Element found after 4444 iteration | Pass |

**Screen Shots:**

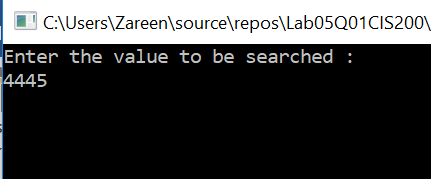
**Test Case 1:**

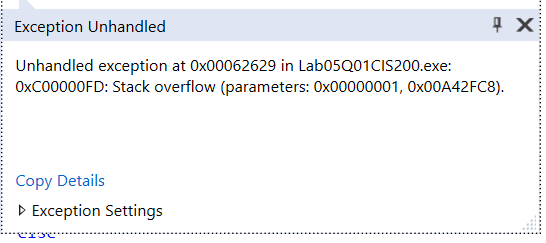


**Test Case 2:**

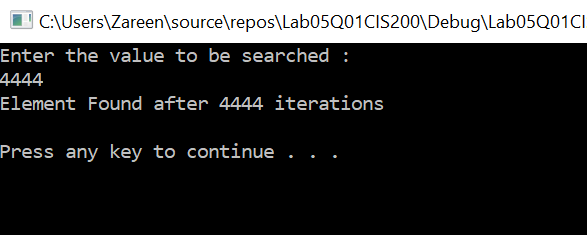


**Test Case 3:**





**Test Case 4:**



**Program 02:**

**Initial Test:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | Invalid | When user input negative value for factorial | Num = -3 | Negative integer cannot be used for factorial. |  |  |
| 2 | Valid | When user input a valid value | Num = 3 | Factorial of 3 is 6 |  |  |
| 3 | Valid | When user input value = 0 | Num = 0 | Factorial of 0 is 1 |  |  |
| 4 | Invalid | User input causes stack overflow | Num = 4790 | Program shows stack overflow |  |  |

**Source File:**

#include <iostream>

using namespace std;

//prototype

int factorial(const int);

int main() {

int n;

cout << "Enter a number: ";

cin >> n;

int y = factorial(n);

if (y == -1) {

cout << "Negative integer cannot be used for factorial." << endl;

}

else

cout << "Factorial of " << n << " is " << y << endl;

system("pause");

return 0;

}

int factorial(const int value) {

if (value < 0) {

return -1;

}

else if (value == 0) {

return 1;

}

else {

return value \* factorial(value - 1);

}

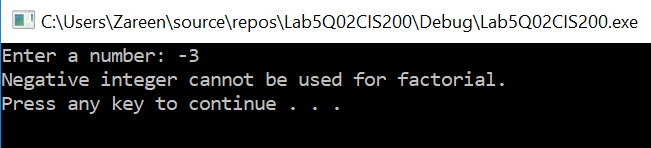
}

**Final Test:**

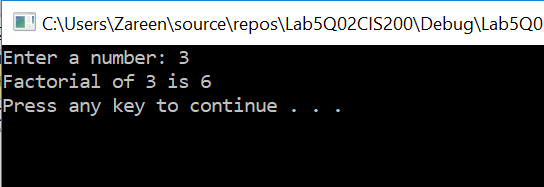
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | Invalid | When user input negative value for factorial | Num = -3 | Negative integer cannot be used for factorial. | Negative integer cannot be used for factorial. | Pass |
| 2 | Valid | When user input a valid value | Num = 3 | Factorial of 3 is 6 | Factorial of 3 is 6 | Pass |
| 3 | Valid | When user input value = 0 | Num = 0 | Factorial of 0 is 1 | Factorial of 0 is 1 | Pass |
| 4 | Invalid | User input causes stack overflow | Num = 4790 | Program shows stack overflow | Program shows stack overflow | Pass |

**Screenshots:**

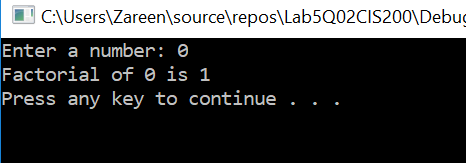
**Test Case 1:**



**Test Case 2:**



**Test Case 3:**



**Test Case 4:**

