**University of Michigan – Dearborn**

**CIS 200 – Computer Science II**

**Lab# 7**

**Professor Robert Mann**

Demetrius Johnson

[meech@umich.edu](mailto:meech@umich.edu)

October 26, 2020

**Table Content**

Contents

[Question 1 (PROGRAM 1) 3](#_Toc54714382)

[Source Code –see program 1 source code in canvas 4](#_Toc54714383)

[Screenshots 4](#_Toc54714384)

[Question 2 (PROGRAM 2) 7](#_Toc54714385)

[Source Code –see program 2 source code in canvas 8](#_Toc54714386)

[Screenshots 8](#_Toc54714387)

# Question 1 (PROGRAM 1)

// LAB 4 - CIS 200

Question 1

-------------------------

Determine smallest key value that causes stack-overflow to occur, even if you need to make array larger than 5000.

\*My smallest value for stack overflow is: userKey = 4233; some cases did worse and not every case did 4233 work without stack overflow, range is about +/- 100 for userKey value of 4223.

\*to do rest of test cases, made my array size to be 4000.

Test cases need to include (not limited to) biggest possible key value, “not found” message, and a stack overflow condition.

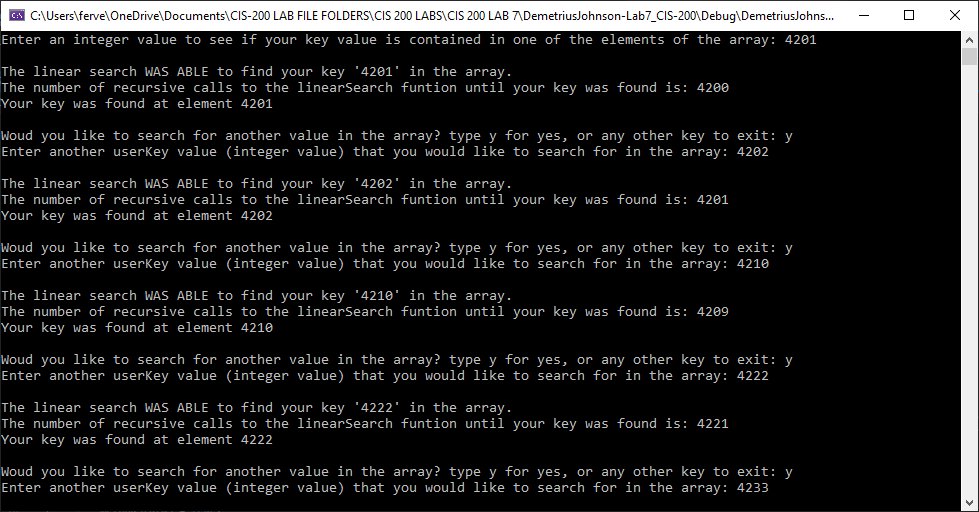
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | VALID | find stack overflow value | userKey = 4223 | Find value in the array | Exception handled; stack overflow | fail |
| 2 | VALID | see if serach can find biggest possible value in the array | userKey = 3999 | Find value in array | See screenshot | pass |
| 3 | VALID | value larger than biggest value in the array | userKey = 4000 | Output message Unable to find key | See screenshot | pass |
| 4 | VALID | search for a negative value | userKey =  -1 | Output message unable to find key | See screenshot | pass |
| 5 | VALID | test another positive integer that is in the array | userKey = 2500 | Find value in array | See screenshot | pass |

## Source Code –see program 1 source code in canvas

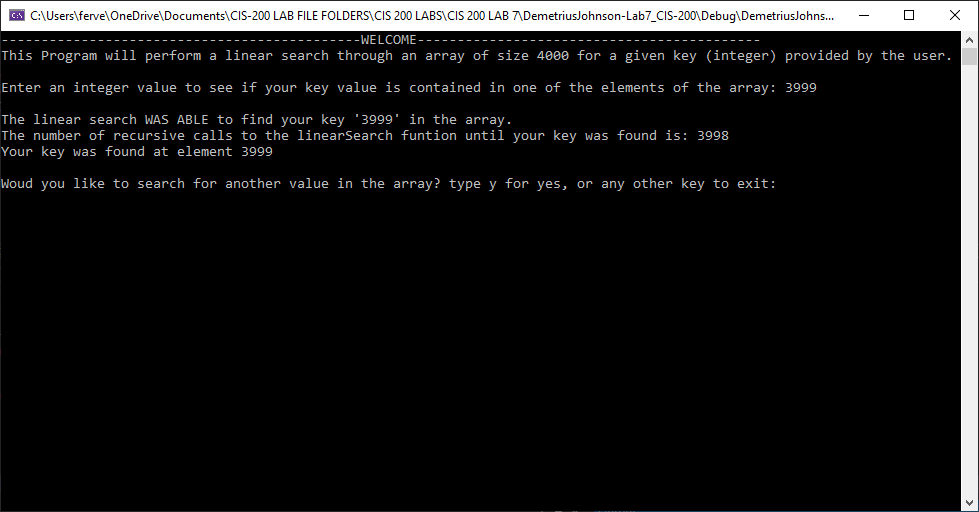
## Screenshots

TEST 1 – STACK OVERFLOW VALUE

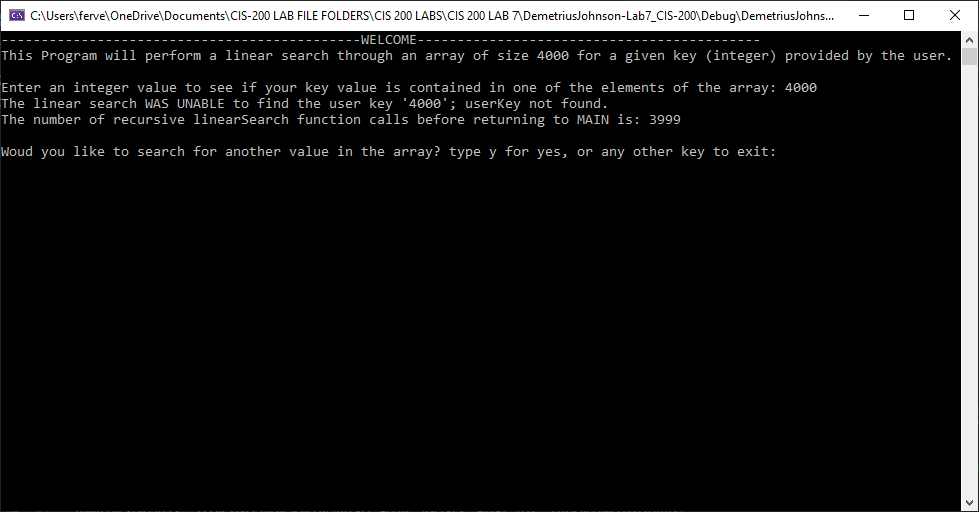
Unhandled exception at 0x000F22E9 in DemetriusJohnson-Lab7\_CIS-200.exe: 0xC00000FD: Stack overflow (parameters: 0x00000001, 0x00A02F4C).



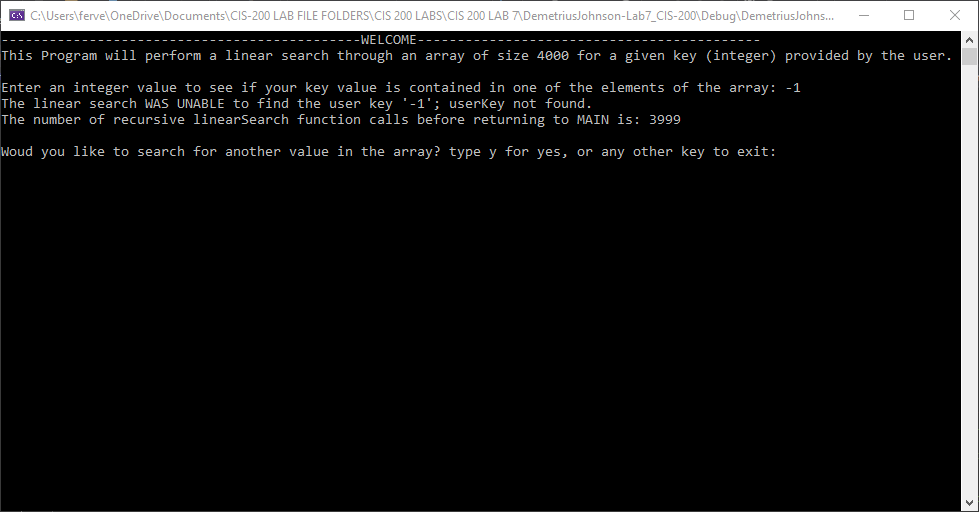
TEST 2 – SEE IF SERACH CAN FIND BIGGEST POSSIBLE VALUE IN THE ARRAY



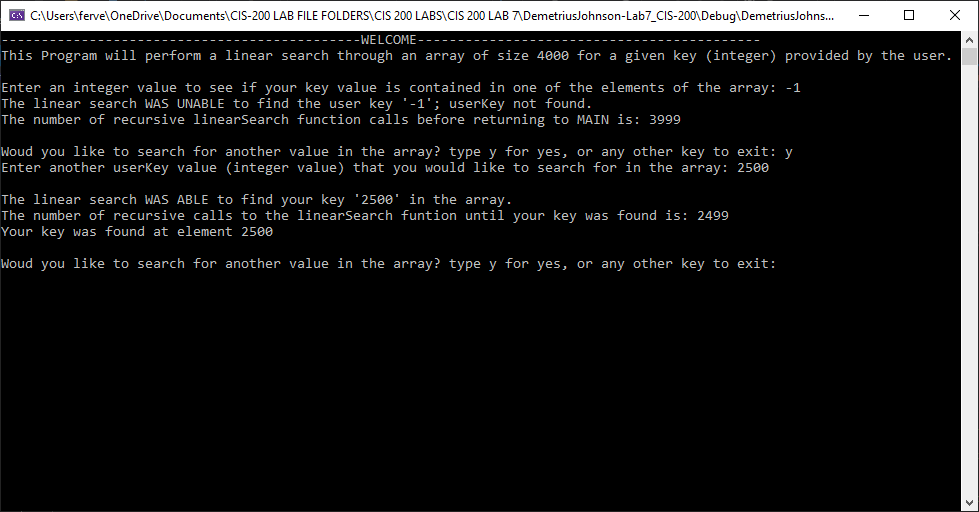
TEST 3 – UNABLE TO FIND USER VALUE 🡪 value larger than biggest value in the array



TEST 4 – SEARCH FOR A NEGATIVE VALUE



TEST 5 – TEST ANOTHER POSITIVE INTEGER THAT IS IN THE ARRAY



# Question 2 (PROGRAM 2)

// LAB 4 - CIS 200

Question 2

-------------------------

**Program 2:**

You are to create a recursive function to perform a factorial calculation.

* **int factorial(const int value)**
* return -1 if any negative number passed into the function
* Calculate the factorial of the number entered by the user

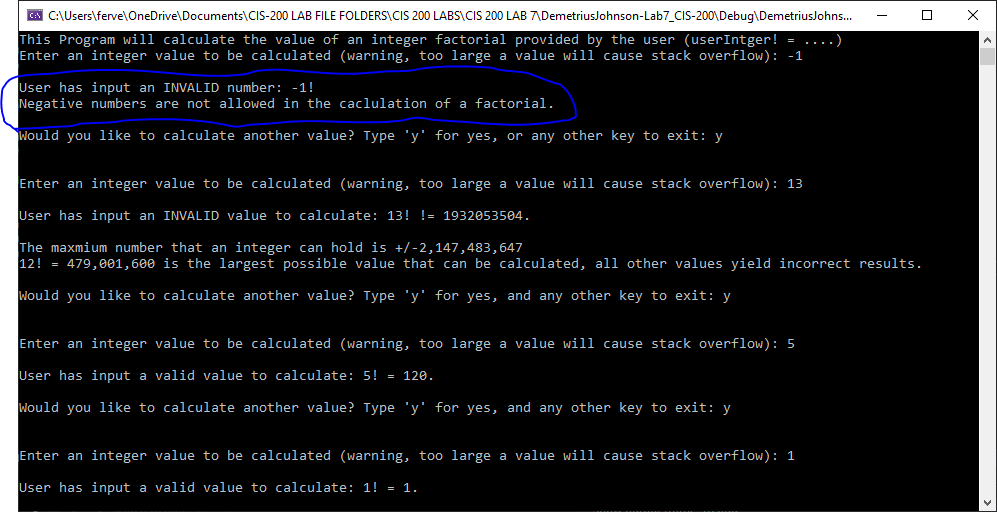
Determine value at which stack overflow occurs.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | VALID | Negative number | UserInt =  -1 | “cannot compute negative factorial” | See screenshot | pass |
| 2 | VALID | Too large of number – stack overflow | UserInt =  13 | 13! =  6227020800 | a value != to 13!. See screenshot | fail |
| 3 | VALID | Number within range | UserInt =  5 | 5! = 120 | 5! = 120  See screenshot | pass |
| 4 | VALID | Testing 1 | UserInt = 1 | 1! = 1 | 1! = 1  See SS | pass |
| 5 | VALID | Testing 0 | UserInt = 0 | 0! = 1 | 0! = 1  See SS | pass |
| 6 | VALID | Largest possible valid value **before** stack overflow | UserInt = 12 | 12! = 479001600 | 12! = 479001600  Correct value; see SS | pass |

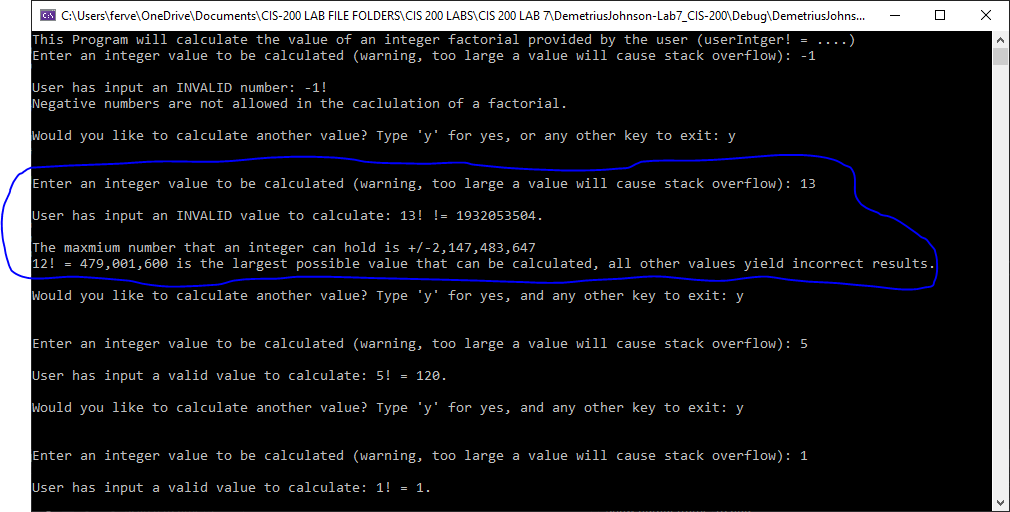
## Source Code –see program 2 source code in canvas

## Screenshots

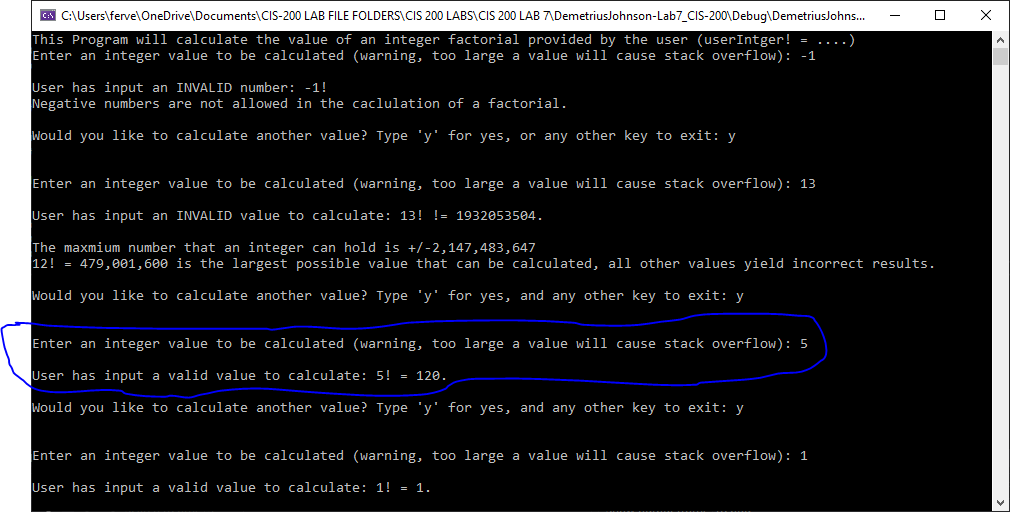
TEST 1 – TEST WITH NEGATIVE NUMBER



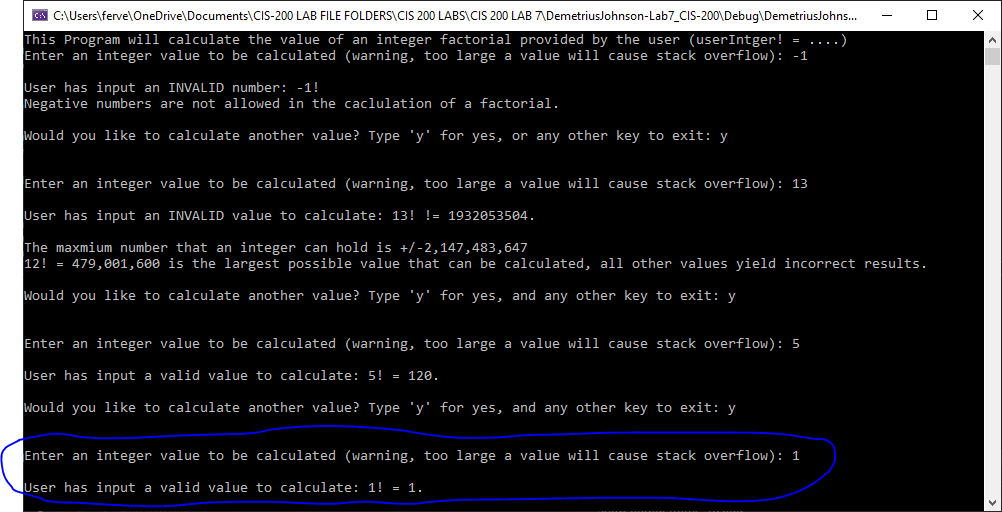
TEST 2 – TEST WITH TOO LARGE A NUMBER – STACK OVERFLOW



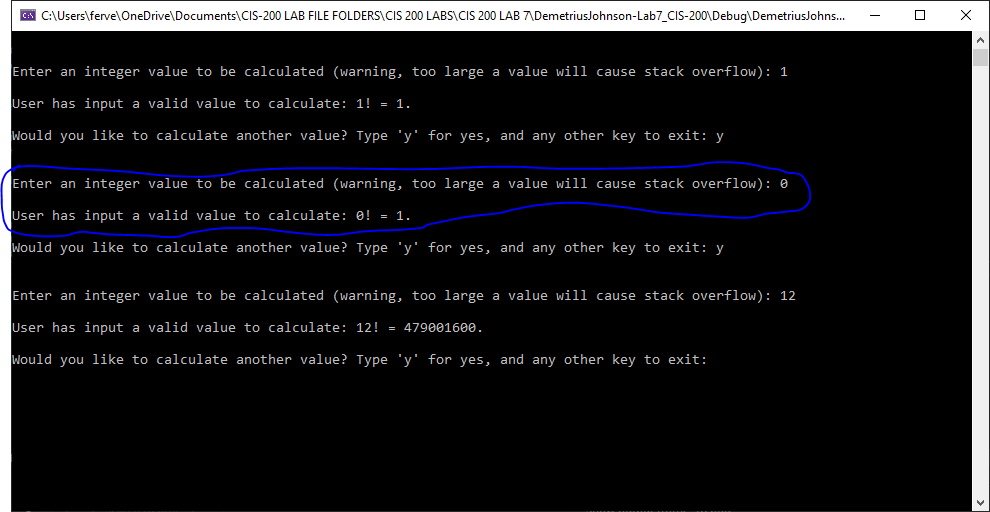
TEST 3 – TEST WITH A VALID NUMBER WITHIN STACK RANGE



TEST 4 – TEST WITH 1 TO SEE IF FUNCTION CAN RESOLVE 1!



TEST 5 – TEST WITH 0 TO SEE IF FUNCTION CAN RESOLVE 0!



TEST 6 – TEST WITH LARGEST POSSIBLE VALID VALUE **BEFORE** STACK OVERFLOW

