Sairam Soundararajan

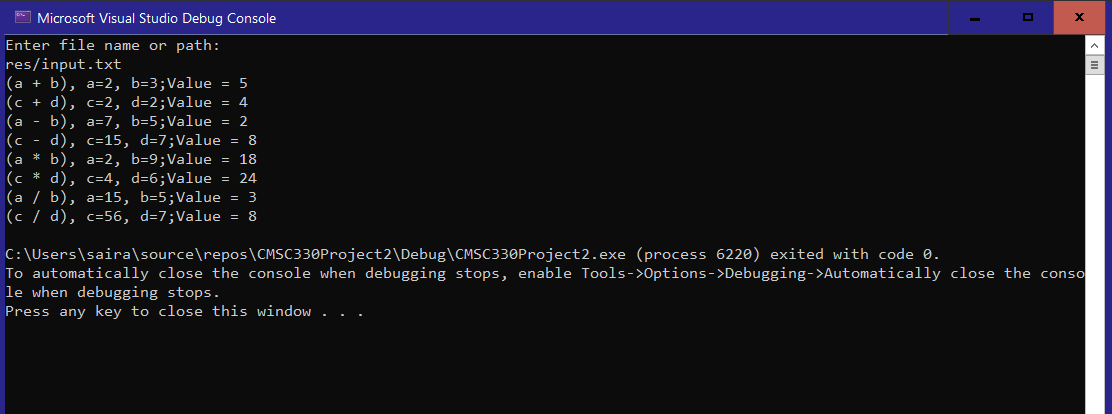
CMSC330: Advanced Programming Languages

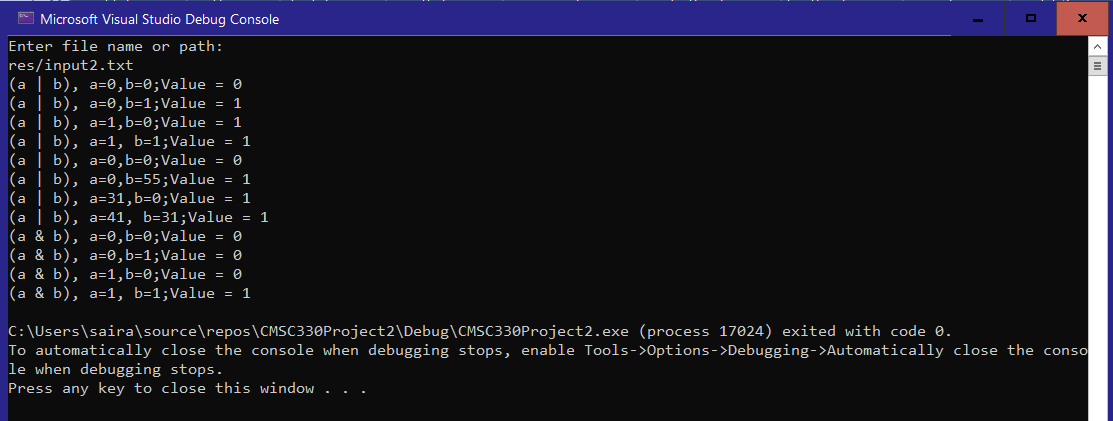
University of Maryland Global Campus

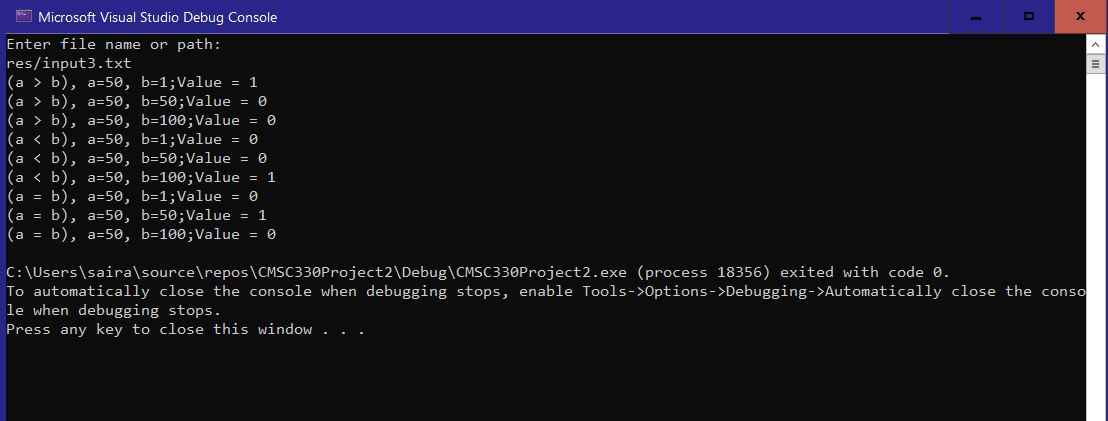
Professor Ghosh

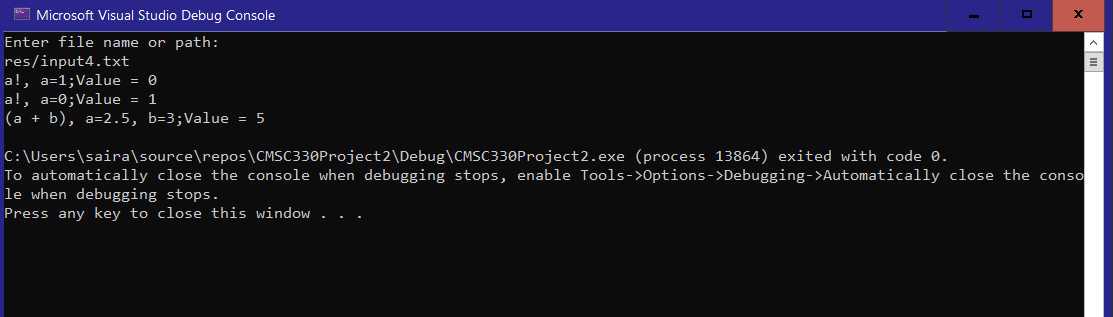
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Expected Output | Actual Output | Pass? |
| 1 | Enter file name or path:  res/input.txt | (a + b), a=2, b=3;Value = 5  (c + d), c=2, d=2;Value = 4  (a - b), a=7, b=5;Value = 2  (c - d), c=15, d=7;Value = 8  (a \* b), a=2, b=9;Value = 18  (c \* d), c=4, d=6;Value = 24  (a / b), a=15, b=5;Value = 3  (c / d), c=56, d=7;Value = 8 | (a + b), a=2, b=3;Value = 5  (c + d), c=2, d=2;Value = 4  (a - b), a=7, b=5;Value = 2  (c - d), c=15, d=7;Value = 8  (a \* b), a=2, b=9;Value = 18  (c \* d), c=4, d=6;Value = 24  (a / b), a=15, b=5;Value = 3  (c / d), c=56, d=7;Value = 8 | YES |
| 2 | Enter file name or path:  res/input2.txt | (a | b), a=0,b=0;Value = 0  (a | b), a=0,b=1;Value = 1  (a | b), a=1,b=0;Value = 1  (a | b), a=1, b=1;Value = 1  (a | b), a=0,b=0;Value = 0  (a | b), a=0,b=55;Value = 1  (a | b), a=31,b=0;Value = 1  (a | b), a=41, b=31;Value = 1  (a & b), a=0,b=0;Value = 0  (a & b), a=0,b=1;Value = 0  (a & b), a=1,b=0;Value = 0  (a & b), a=1, b=1;Value = 1 | (a | b), a=0,b=0;Value = 0  (a | b), a=0,b=1;Value = 1  (a | b), a=1,b=0;Value = 1  (a | b), a=1, b=1;Value = 1  (a | b), a=0,b=0;Value = 0  (a | b), a=0,b=55;Value = 1  (a | b), a=31,b=0;Value = 1  (a | b), a=41, b=31;Value = 1  (a & b), a=0,b=0;Value = 0  (a & b), a=0,b=1;Value = 0  (a & b), a=1,b=0;Value = 0  (a & b), a=1, b=1;Value = 1 | Yes |
| 3 | Enter file name or path:  res/input3.txt | (a > b), a=50, b=1;Value = 1  (a > b), a=50, b=50;Value = 0  (a > b), a=50, b=100;Value = 0  (a < b), a=50, b=1;Value = 0  (a < b), a=50, b=50;Value = 0  (a < b), a=50, b=100;Value = 1  (a = b), a=50, b=1;Value = 0  (a = b), a=50, b=50;Value = 1  (a = b), a=50, b=100;Value = 0 | (a > b), a=50, b=1;Value = 1  (a > b), a=50, b=50;Value = 0  (a > b), a=50, b=100;Value = 0  (a < b), a=50, b=1;Value = 0  (a < b), a=50, b=50;Value = 0  (a < b), a=50, b=100;Value = 1  (a = b), a=50, b=1;Value = 0  (a = b), a=50, b=50;Value = 1  (a = b), a=50, b=100;Value = 0 | Yes |
| 4 | Enter file name or path:  res/input4.txt | a!, a=1;Value = 0  a!, a=0;Value = 1  (a + b), a=2.5, b=3;Value = 5 | a!, a=1;Value = 0  a!, a=0;Value = 1  (a + b), a=2.5, b=3;Value = 5 | Yes |

Snapshots of Test Run:









Reflection:

For this project, I used C++ to make an expression evaluator. This program computes operations of simple infix expressions through arithmetic, relational, logical, and conditional operations. My job was to allow this program to support these operations. With the plus class already designed for addition, I continued by providing the classes that compute subtraction, multiplication, and division. Along with that, I made other classes that compute relational, logical, and conditional operations. Although I attempted to test complex expressions, I realized that function was not what the project specializes in. Making the file input/output portion of the code was difficult at first, but through some online assistance, I was able to modify the program to enable the file input/output function. For every expression, the result of each operation always results in an int, after converting from a double to an int.