Colton Willits

Dr. Graham

CSC314

Extra Credit Experimental Program for multi-case Comparison

**Desired Output in C**

The desired output is represented by the following C program code (found in multicase.c)

Text

Description automatically generated

(Desired code in C)

The desired output should be as follows

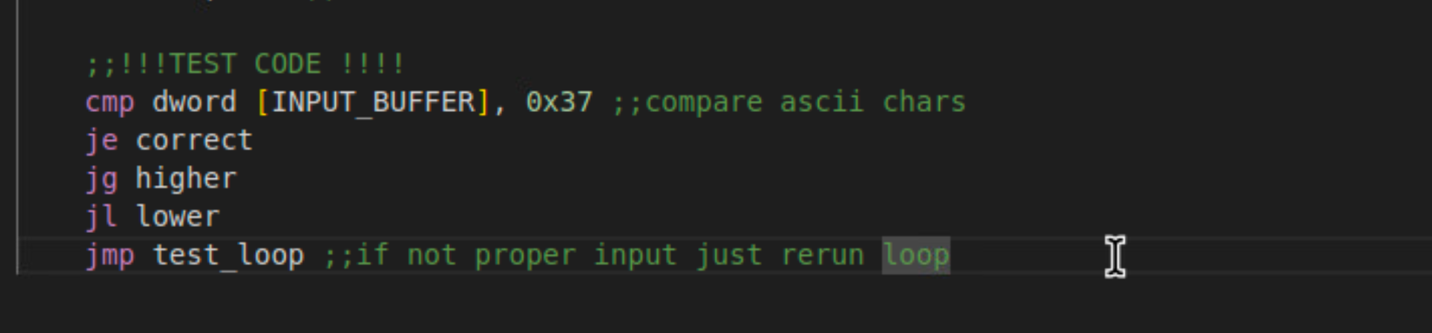
Text

Description automatically generated

(Desired output)

**First Attempt**

The assembly code that is being tested is comparing the ascii value of 7



(Test Code)

Text

Description automatically generated

(First Test Output)

The first set of test show that the output is backwards from jumping lower and higher than my original thought process. Output shows this, so I will reverse the jump greater and jump lower. Initial hypothesis is that the code will work. I possibly have bad logic in my code that reruns the loop with a null input.

**Second Attempt**

After reversing the jg and jl it appears the program works properly except for the secondary looping. It properly identifies the exit character 0, and will properly compare and jump depending on the input number. If less than 7 it will display try a number that is higher, and if greater than 7 it will display a try a number that is lower. In hindsight, this was the proper order of the messages since I didn’t put the right display messages in the correct “methods”. I originally put higher message in higher and lower message in lower. However it doesn’t affect how the test code works. Just my output that I programmed.

Text

Description automatically generated

(Reversed output test code with exit comparison)

Text

Description automatically generated

(Test output with reversed output corrected)

**Walk-though of test code in kdbg**

Neglecting that the program will loop twice on one input. We can look at the comparison in kdbg.

Testing the first value 2 we can see that it sets a flag.

Text

Description automatically generated

(Flag is set by comparison)

ASCII value of 2 is lower than the ASCII value of 7 here it jumps to the try a higher number message.

Graphical user interface

Description automatically generated with medium confidence

(Jumps from jl conditional jump to print higher message)

Following our test case of using a higher number than 7, 8,the flag is set to greater than and jumps on the jg case. Program will print try a lower number.

Text

Description automatically generated

(Flag is set by comparison)

Graphical user interface

Description automatically generated

(jg jumps to use print try a lower number)

Finally testing the correct number by comparing the ASCII value of 7 to the input of ASCII 7. We can see that the flag is set and the je goes to print the correct message and exit the program.

Text

Description automatically generated

(comparison flag is set)

Graphical user interface, text

Description automatically generated with medium confidence

(conditional jump correctly jumps to print the correct message)

**Conclusion**

Overall, the idea of using multiple jumps after a single comparison is a valid option in assembly for multiple cases. The comparison flag does not change through each conditional jump and will correctly act in an expected manner.

Text

Description automatically generated

(Desired output)

A screenshot of a computer

Description automatically generated

(Reverse Engineered Code output from Ghidra)

Looking at the multicase assembly program reversed and the multicase c program you can see that both do if-elseif-else comparisons. However, the reverse engineering software also flags this for being a bad loop as the issue of it running the loop twice per input is persistent. However, we are more focused on the comparison between our desired output and our programs output. Our code significantly matches the desired output C code in functionality. It compares if the input is 7 and then outputs a corresponding value or message. If correct it will exit the program, otherwise it will loop back.