**CMPE-250 Assembly and Embedded Programming**

**Laboratory Exercise 3**

**Memory, Conditional Branching, and Debugging Tools**

By submitting this report, I attest that its contents are wholly my individual writing about this exercise and that they reflect the submitted code. I further acknowledge that permitted collaboration for this exercise consists only of discussions of concepts with course staff and fellow students. Other than code provided by the instructor for this exercise, all code was developed by me.

Shubhang Mehrotra

Performed 11th February 2021.

Submitted 18th February 2021.

Lab Section 1

Instructor: Muhammad Shaaban

|  |  |  |
| --- | --- | --- |
| TA: |  | Anthony Bacchetta |
|  |  | Sam Myers |
|  |  | Payton Burak |

Lecture Section 1

Lecture Instructor: Dr. Roy Melton

**Results**

The activity involved calculation of the following equations:

F = 3P + 2Q – 75

G = 2P – 4Q + 63

Result = F + G

For the following two sets of inputs.

Input Set 1: P = 9, Q = 4

Figure 1 shows the values stored in the memory after the program written was successfully debugged. The first two memory addresses show the values of F and G, respectively. They are marked in Green because they were written to.

The next to memory spaces are marked in Red as they were read from. They held the values of P and Q, respectively. The last highlighted memory address is the Result variable, which too was updated at the end of the operation.

Graphical user interface, text, application

Description automatically generated

Figure 1. Memory variables result of Input Set 1

A picture containing calendar

Description automatically generated

Figure 2. Register Content for Input Set 1

Manual Calculation for input set 1:

F = 3(9) + 2(4) – 75 = 27 + 8 – 75 = –40

F = –40

G = 2(9) –4(4) + 63 = 18 – 16 + 63 = 65

G = 65

Result = –40 + 65 = 25

Input Set 2: P = 13, Q = –14

Figure 2 shows the values stored in the memory after the program written was successfully debugged. Like the first input set, the first two memory addresses show the value of F and G. The next two addresses are the inputs P and Q, and the last address is the Result.

Graphical user interface, application

Description automatically generated

Figure 3. Memory variables result of Input Set 2

A picture containing table

Description automatically generated

Figure 4. Register Content for Input Set 2

Manual Calculation for input set 2:

F = 3(13) + 2(–14) – 75 = 39 – 28 – 75 = –64

F = –64

G = 2(13) –4(–14) + 63 = 26 + 56 + 63 = 145 <– Overflow

G = 0

Result = –64 + 0 = – 64

**Question**

*Could you reduce or eliminate overflow by changing the order of operations within the expressions (F, G, and/or Result)? Explain why or why not.*

No, overflow cannot be eliminated by changing the order of operations within the expressions. No matter what we do, we are bound to get the same result which will cause an overflow.