**CMPE-250 Laboratory Exercise Nine**

**Multiprecision Arithmetic with**

**Mixed C and Assembly Language Programming**

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; however, other than code provided by the instructor for this exercise, all code was developed by me.

John Judge

Performed 4/7/16

Submitted 4/14/16

Lab Section 02

Instructor: Dr.Shaaban

TAs: Peter Muller

Stephen Moore

Connor Goss

Lecture Section 01

Professor: Allesandro Sarra

1. The results of the program are shown in Figure (1.0). The program ran as expected and the results matched the desired outputs. During the testing, the program was verified to be operating correctly by several test cases. To begin the verification the program was asked to compute the sum of zero and an integer. The program responded correctly to the add by zero test. Next the program was tested for overflow by asking it to compute a sum more than the program could display. The program responded correctly by displaying “OVERFLOW” as the sum. Then the use lowercase hex values was tested and the program correctly preformed the addition and displayed the sum with capital letters. Then the program was tested with several invalid entries and for each it correctly responded with the “Invalid number” prompt. One final addition operation with valid entries took place and after this it was verified the program operated correctly.

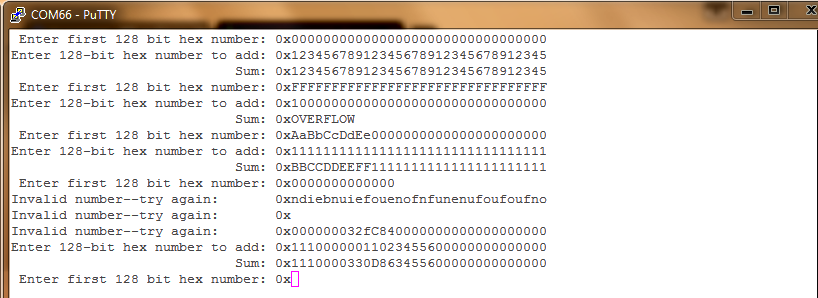


Figure (1.0): The results of the program

1. Following the test of the program the memory map of the program was examined. This memory map vastly differed from those of previous exercises due to the mixed C implementation. The Image Symbol table contained far more addresses than previous exercises, most of which were created by the compiler. From examining the Image Symbol Table the starting address of several functions in ROM were recorded. AddIntMultiU was found to be located at address 0x00000411. GetStringSB was located at 0x00000543. The function PutStringSB was located at 0x00000573. Init\_UART0\_IRQ was located at address 0x0000064f. Since the UART0\_ISR is an interrupt service routine it was located at the first address in memory. Since the subroutines Enqueue and Dequeue were not exported from the assembly file they did not appear in the memory map. In order to find the locations of the C files the memory map needed to be examined. GetHexIntMulti was located at 0x000002de. PutHexIntMulti was located at address 0x00000758 in ROM. The main C function was located at the address 0x000007b4.