

Student Name: _____ Student ID: _____

Problem 1

Provide the assembly code that can perform the following code.

Full credit for 5 instructions or less and 10 points off for every additional instruction.
Assume that `do_something` – a function in C – is a subroutine in assembly.

Make sure no registers are altered by this code after completion.

```
for (i=10; i > 0; i--)  
{  
    do_something();  
}
```

Student Name: _____ Student ID: _____

Problem 2

Write the equivalent C code for the assembly below. Assume J and K are global variables declared with ds.b directive that are signed integers in C.

```
J      ds.b      1
K      ds.b      1
...

      ldaa  J
      cmpa #$F0
      bne  done
      cmpa #$00
      bne  done
      movb #$00, K
done:  movb #$FF, K
```

Student Name: _____ Student ID: _____

Problem 3

Below is the code for a simple subroutine that uses the stack in an unconventional manner – considered irresponsible by some. Given the initialized values in the table, fill in the rest of the table describing the registers and memory locations at different times during the execution of the function. All known values in the table must be reported in hex.

```

                                org    $3000
multi_output:                  cmpa   #$00                      ; 2 byte instruction
                                bne   output_if_nonzero          ; 2 byte instruction
output_if_zero                  movw  #$1000, 0,SP                ; 4 byte instruction
                                rts                               ; 1 byte instruction
output_if_nonzero              movw  #$1010, 0,SP                ; 4 byte instruction
                                clra                               ; 1 byte instruction
                                rts                               ; 1 byte instruction
    
```

Location	Just prior to jsr execution of calling sequence	Just after the bne execution	Prior to first RTS execution	Just after first RTS execution
A	12			
B	34			
D				
X	2000			
Y	5678			
SP	3C00			
PC	2000			
Mem 3BF5	00			
Mem 3BF6	00			
Mem 3BF7	00			
Mem 3BF8	00			
Mem 3BF9	00			
Mem 3BFA	00			
Mem 3BFB	55			
Mem 3BFC	55			
Mem 3BFD	00			
Mem 3BFE	00			
Mem 3BFF	FF			
Mem 3C00	FF			
Mem 3C01	FF			