UNCC, Department of Electrical and Computer Engineering

ECGR3/5/6090, Fall 2003, Homework #4, Due: 9/24/03, at the beginning of class (20 points)

```
;## #     FUNCTION main
;## #     FRAME     AUTO (   array)     size 200, offset -128
;## #     FRAME     AUTO (       j)     size   2, offset 72
;## #     FRAME     AUTO (       i)     size   2, offset 74
...
     mov.w     #0000H,74[FB]  ;  i
L0:
     cmp.w     #000aH,74[FB]  ;  i
     jge  L2
     mov.w     #0000H,72[FB]  ;  j
L6:
     cmp.w     #000aH,72[FB]  ;  j
     jge  L8
     mov.w     74[FB],R0 ;  i
     mul.w     #0014H,R0
     mov.w     72[FB],R1 ;  j
     shl.w     #01H,R1
     add.w     R1,R0
     mova -128[FB],A0    ;  array
     add.w     R0,A0
     mov.w     74[FB],R0 ;  i
     mul.w     72[FB],R0 ;  j
     add.w     R0,[A0]
     add.w     #0001H,72[FB]  ;  j
     jmp  L6
L8:
     add.w     #0001H,74[FB]  ;  i
     jmp  L0
L2:
```

1. Above a snippet of MC30262 assembly code is provided. Using the variable names provided in the header of the function, “reverse-engineer” this code into C. (5 points)

2. On page 12 of Notes 4 we show a short snippet of C code which we went over in class. Assuming that the FB register contains 0x60a and the SP register contains 0x600, show the contents of the stack (I want values, not variable names like we did in class) after the program has run and just completed executing the line of code in squared, line mul.w for the first time. Assume the jsr $compute instruction in main is at address 0xF0100 and the jsr $squared instruction in compute is at address 0xF0300. Assume a drawing of the stack which is 8-bits wide. (15 points)