

UNCC, Department of Electrical and Computer Engineering
ECGR4101/5101, Spring 2010, Homework #3, Due: 1/29/10, 5 pm (20 points) -Version 2

Consider the following listing of a program compiled from C. It includes the original C source code as comments for clarity. Show the contents of the stack and registers R0, R1, SP and FB just before the `exitd` in `$f2` executes. Also identify what each byte represents (e.g. LSB (least significant byte) of argument, MSB of variable, dynamic link, etc.). Assume that execution starts at the beginning of `f1`, at which point `FB` is `0942h` and `SP` is `093Eh`. Assume that the address of the `mov.w` instruction following `jsr $f2` is `FA012h`. Use `?` to indicate values which are unknown or in boxes which are not used. **(Version 2 changes are bolded)**

```

;## # FUNCTION f2
;## # FRAME AUTO (      arg) size  2,  offset -2
;## # REGISTER ARG (      arg) size  2,  REGISTER R1
;## # ARG Size(0) Auto Size(2)    Context Size(5)
;## # C_SRC :      int f2(int arg) {
$f2:
    enter #02H
    mov.w R1,-2[FB] ; arg arg
;## # C_SRC :      return arg+5;
    mov.w -2[FB],R0 ; arg
    add.w #0005H,R0
    exitd ; ##### Show contents before this line executes

;## # FUNCTION f1
;## # FRAME AUTO (      b) size  2,  offset -4
;## # FRAME AUTO (      a) size  2,  offset -2
;## # ARG Size(0) Auto Size(4)    Context Size(5)
;## # C_SRC :      void f1(void) {
_f1:
    enter #04H
;## # C_SRC :      int a=10, b;
    mov.w #000aH,-2[FB] ; a
;## # C_SRC :      b = f2(a);
    mov.w -2[FB],R1 ; a
    jsr $f2
    mov.w R0,-4[FB] ; b
;## # C_SRC :      }
    exitd

```

Address	Contents	Description _____
931		
932		
933		
934		
935		
936		
937		
938		
939		
93a		
93b		
93c		
93d		
93e		
93f		
940		

Register	Contents
R0	
R1	
SP	
FB	