In the following C code.... What is printed for the value of X and Y?

```c
int f1(int a)
{
    a = 10;
}

int f2(int *b)
{
    *b = 20;
}

int main()
{
    int x = 30;
    int y = 40;
    f1(x);
    f2(&y)
    printf("X is %d and Y is %d\n", x, y);
}
```

2) Fill in the code for a C subroutine that adds two integers together, and stores the result in the location indicated by a pointer.

```c
int addstuff(int a, int b, int *c)
{
    // One line of your code here.
}
```

3) Now, write the equivalent of addstuff from Question 2, using LC3 assembly. Here's part of the code (the main program). Make sure you test your code in the simulator after you write it.

```assembly
.ORIG x3000
    LD R6, STACKBASE
    LEA R0, Z
    ADD R6, R6, #-1
    STR R0, R6, #0
    LD R0, Y
    ADD R6, R6, #-1
    STR R0, R6, #0
    LD R0, X
    ADD R6, R6, #-1
    STR R0, R6, #0
    JSR ADDSTUFF
    ADD R6, R6, #3
    HALT
X .FILL 10
Y .FILL 20
Z .FILL 0
```
4) Write LC3 code to push the ADDRESS of a variable X onto the stack, and then call a function DOUBLE. There are a pair of LC3 instructions you would use for push:

; Main routine here -- push address of X on the stack, then call DOUBLE.
; And don't forget to do the appropriate thing to the stack after you return....
; You can assume that the register for the stack pointer is set up.

5) Write LC3 code for a function DOUBLE that will use the address provided on the stack to double the variable passed to the function.

6) You can use a stack as temporary storage for computations. If you see a number, it is pushed on the stack. If you see a math operation, it is performed immediately on the top two elements (with the result being pushed back). Suppose the following steps occur; what is on the top of the stack when you finish?
5
7
2
add
3
multiply
add

7) The following code modifies MESSAGE. What's MESSAGE look like after this runs?

LEA R0, MESSAGE
LDR R2, R0, #2
ADD R2, R2, #1
STR R2, R0, #5
HALT
MESSAGE .STRINGZ "FUNPROBLEM"