NAME: _____ GRADE: _____

EE492 / EE592 Real-Time Digital Signal Processing
Quiz #1
Open Books, Notes, Calculators (no programs, no graphing)

1. Write the hexadecimal values for the following 24-bit fixed-point decimal fractions. The first one is done for you.

+0.5000000	$400 000
+0.0312500	$
-0.6250000	$

2. Write the 24-bit fixed-point, decimal fractions (at least 7 digits of precision) for the following hexadecimal values. The first one is done for you.

+0.2500000	$200 000
	$248 000
	$F24 000

3. Write the hexadecimal values for the following 56-bit fixed-point accumulator (decimal integer/fraction). The first one is done for you.

+128.0000000	$40 000 000 000 000
+ 32.0312500	$
- 4.6250000	$

4. Write the 56-bit fixed-point accumulator [decimal integer/fraction (at least 7 digits of precision)] for the following hexadecimal values. The first one is done for you.

+64.0000000	$20 000 000 000 000
	$01 300 000 000 000
	$F8 600 000 000 000
	$32 600 000 000 000

5. For each of the following hexadecimal values in register y1, determine the resulting 14-digit hexadecimal value in accumulator, a when the following instruction is executed:

```
move y1,a
```

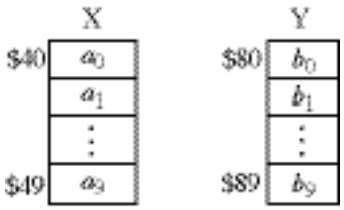
y1	a
$123 456	$
$876 543	$
$CBA 987	$

6. For each of the following hexadecimal values in accumulator b, determine the resulting 6-digit hexadecimal value in register x0, when the following instruction is executed:

```
move b,x0
```

b	x0
$00 123456 789ABC	$
$00 234567 89ABCD	$
$00 000001 500000	$
$FF 000000 800000	$

7. Assume two vectors, **a** and **b** and are stored in memory as follows:



Write a short code to do the following:

- Initialize address registers to point to a_0 and b_0 (assume linear addressing not modulo)
- Compute the inner product (use REP instruction) as

$$c = \sum_{n=0}^{9} a_n b_n$$

- The result should be rounded at the end of the calculation.

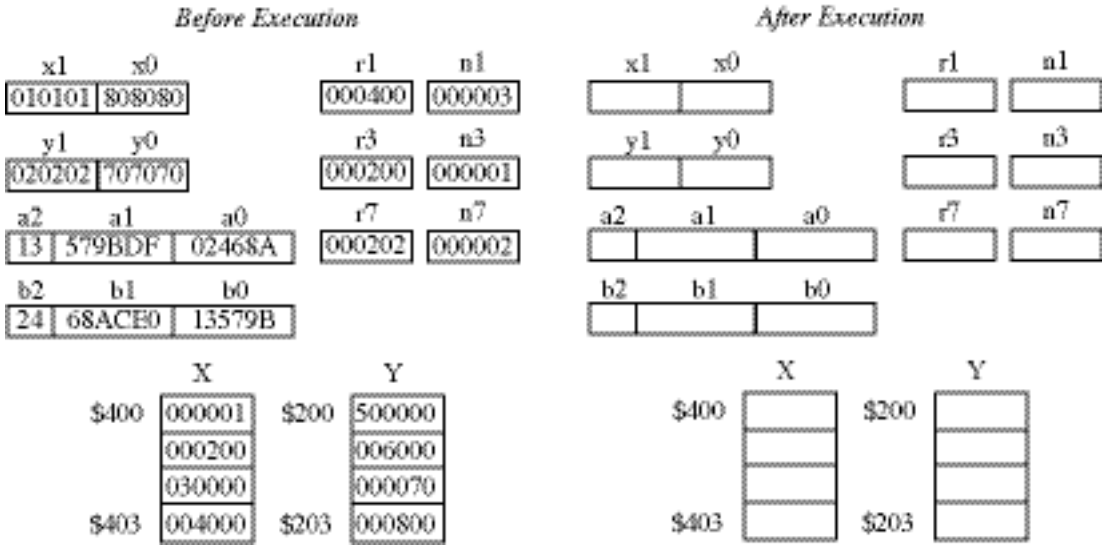
8. Write a short code to compute the following equation

$$(0.4)^2 + 0.5$$

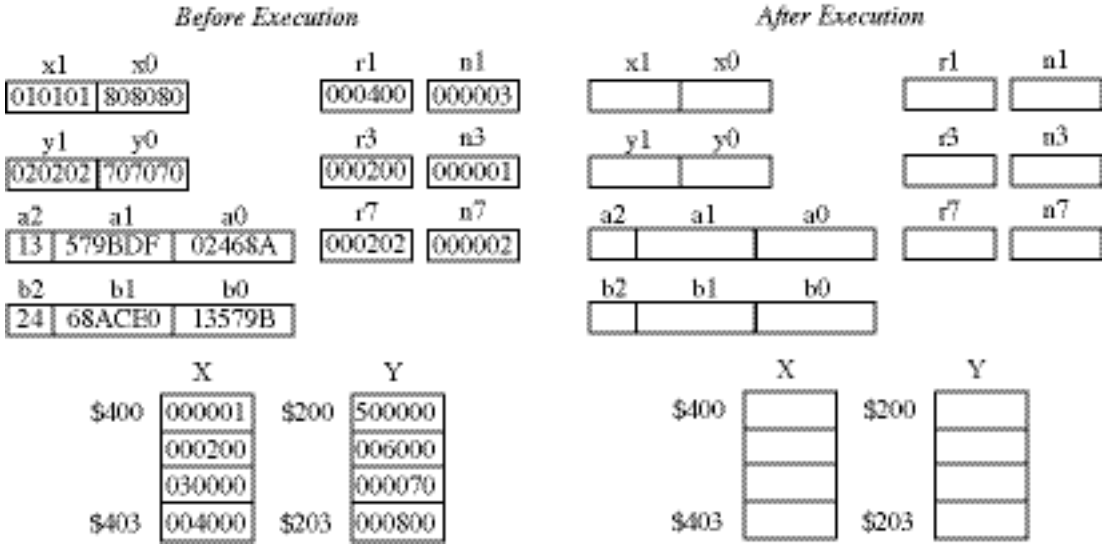
Three lines of correct code will yield full credit; four or more lines of correct code will yield half credit.

For the following questions, determine the register, accumulator, and memory states after execution of the instruction. Put 6 hexadecimal digits in each box (or 2 in the case of a2). If you leave the box blank, we assume the state does not change. If the instruction cannot be executed, write "CANNOT BE EXECUTED"

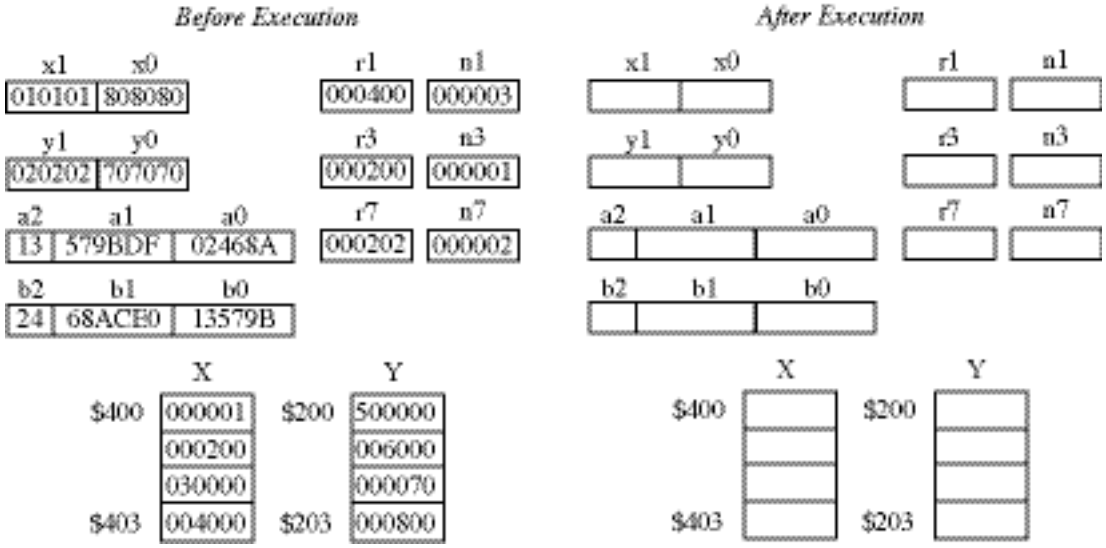
9. `move x:(r1+n1),y1`



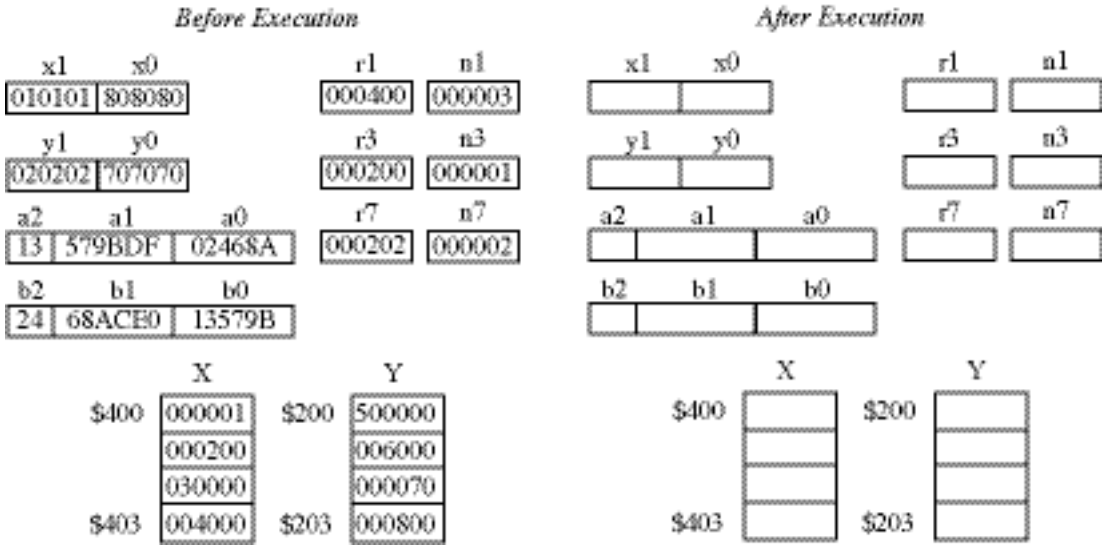
10. `move x0,y:(r3+n3)`



11. `move a,y:(r3)+`



12. `move b1,y:-(r7)`



This page EE592 only

13. For the following moduli (plural of modulus) and starting addresses, determine the first valid address, *after* the starting address, where an array may be modulo-addressed. The first one is done for you.

Modulus (Decimal)	Starting Address	First Valid Address for Modulo Addressing
8	$F	$10
40	$1A	
255	$10F	
256	$10A	
257	$10F	

For the following questions, determine the address that the assembler assigns for array storage. Note the use of DSM (Define Storage Modulo) and DS (Define Storage). The first one is done as an example (values are decimal).

14.

```
      org x:$a
hisvec      dsm         4
```

ANSWER: The array hisvec is stored at address x:$00000C

15.

```
      org x:$123
hervec      dsm         42
```

16.

```
      org x:$246
myvec       ds          64
```

17.

```
      org x:$100
yourvec     dsm         128
```

18.

```
      org x:$3
vec1        dsm         24
vec2        ds          16
vec3        dsm         10
```