NAME:_____*

**CS 255 - Computer Organization/Architecture I**

**Practice Questions for Midterm**

Note: I do not have answers to these questions. Nor do I intend to provide answers to them. You should work them out among yourselves. You should have learned all that is necessary from your project assignments already but if you still have problems with these questions, you may ask me *specific questions*. I will not answer questions like "how do you do this one ?" because I want to educate students and do not want to teach students how to prepare for tests...

**Question 1.**

Suppose registers `r0` and `r1`, and memory locations 80, 81, 82 and 83 current have the following bit pattern:

| | | | | |
|---|---|---|---|---|
| r0 | 00000001 | 00000001 | 00000001 | 00000001 |
| r1 | 11111111 | 11111111 | 11111111 | 11111111 |

Memory:

| | |
|---|---|
| 80 | 10101010 |
| 81 | 01010101 |
| 82 | 00000000 |
| 83 | 11110000 |

1. What is the bit pattern in register `r1` after the CPU executes the instruction `mov r0, r1` ?
2. Starting with the original values in the registers and in memory, what is the bit pattern in register `r1` after the CPU executes the instructions:

```
mov r0,#80
ldrsb r1,[r0]
```

3. Starting with the original values in the registers and in memory, what is the bit pattern in register `r1` after the CPU executes the instructions:

```
mov r0,#80
ldrsh r1,[r0]
```

4. Starting with the original values in the registers and in memory, what is the bit pattern in register `r1` after the CPU executes the instruction:

```
mov r1,#-75
```

5. Starting with the original values in the registers and in memory, what is the bit pattern in the memory after the CPU executes the instructions:

```
mov r1,#80
str r0,[r1]
```

6. Starting with the original values in the registers and in memory, what is the bit pattern in the memory after the CPU executes the instructions:

```
mov r1,#80
strb r0,[r1]
```

7. Starting with the original values in the registers and in memory, what is the bit pattern in the memory after the CPU executes the instructions:

```
mov r0,#80
str r1,[r0]
```

8. Starting with the original values in the registers and in memory, what is the bit pattern in the memory after the CPU executes the instruction:

```
mov r0,#-2
mov r1,#80
str r0,[r1]
```

**Question 2.**

- Give the 3 digits 10s complement representation for -95
- Give the 8 bits 2s complement representation for -95
- Give the IEEE 754 representation for -95.625
- Perform this computation in base 5: 2343 + 4244

**Question 3.**

An array of integer `MyIntArr`, an array of short `MyShortArr`, variables `byte i, short j, int k int IntVar`, and `short ShortVar` are defined as following in assembler:

```
MyIntArr:   .skip 400
MyShortArr: .skip 20
i:          .skip 1
j:          .skip 2       // Ignore alignment !
k:          .skip 4
IntVar:     .skip 4
ShortVar:   .skip 2
```

Write the ARM assembler instructions that accomplishes the equivalent of the following high level language statements:

1. `MyIntArr[k] = 1234;`
2. `MyShortArr[i+j] = 1234;`
3. `MyIntArr[k+4] = MyIntArr[i] + MyShortArr[j] + IntVar;`
4. `MyShortArr[k+4] = MyIntArr[i] + MyShortArr[j] + ShortVar;`

**Question 4.**

A class is defined as follows:

```
class MyObj
{
   int SSN;
   int Balance;
   MyObj next;
}
```

Assumed that a linked list of `MyObj` objects has been previously constructed and the assembler variable

```
head: .skip 4
```

refers to the first object in the list (i.e., contains the address of the first object in the list).

Write the ARM assembler instructions that accomplishes the equivalent of the following high level language statements:

1. `head.next.SSN = head.SSN;`
2. `head.Balance = head.next.next.Balance + 1234;`

**Question 5.**

Translate the following program fragment into ARM assembler instructions:

```
int A, B, GDC;

while ( A != B )
{
   if ( A > B )
      A = A - B;
   else
      B = B - A;
}

GDC = A;
```

**Question 6.**

The variables `x` and `y` are integers. Translate the following program fragment into ARM assembler instructions:

```
if ( x + y >= 70 && x > y )
{
   x = x + 1;
}
else if ( x < y || x + y > 50 )
{
   y = y - 1;
}
```

**Question 7.**

Write a `int parseInt(String s)` method to convert a **base 5** number string into binary.

The digits used are 0, 1, 2, 3 and 4.