Practice Exam 1

The exam will be open-book, so that you don't have to memorize the ASCII table or the details of the Pep/7 architecture.

1. How many memory reads are required to fetch and execute the instruction

   ```
   add     X, Y
   ```

   on a two-address architecture (where `X` and `Y` are direct-mode operands)? Recall that this instruction is roughly equivalent to `X += Y` in C++.

   How many memory *writes* are required?

   Give an equivalent sequence of instructions for the Pep/7 architecture, and tell how many memory reads and writes are required for it.

2. Fill in the missing entries.

Binary	ASCII	Decimal	Hexadecimal
1101001			
	'&'		
		126	
			4E

3. Fill in the missing entries. For this problem, all binary numbers are 8 bits.

Decimal	Sign Magnitude	2's Complement	Excess 127
-100			
	1111 1111		
		1111 1111	
			0110 1001

4. Convert the following C++ program to Pep/7 Assembly Language:

```
#include <iostream.h>

int a, b;

int main() {
  cin >> a;
  cin >> b;
  b += a;
  a = b - a;
  cout << a;
  cout << b;
}
```

5. Evaluate the following expressions, giving the results in hexadecimal:

   (a) $\texttt{3E92} + \texttt{147B}$

   (b) $\texttt{7000} - \texttt{1234}$

   (c) $\texttt{05F6} \times \texttt{8}$

6. Convert the following Pep/7 program to an equivalent program in C++:

```
newLine: .EQUATE h#000A
         BR      main
x:       .WORD   d#1
y:       .WORD   d#2
z:       .WORD   d#3
c:       .BYTE   d#4
main:    DECI    y,d
         LOADA   y,d
         ASLA
         STOREA  x,d
         ASLA
         ASLA
         ADDA    x,d
         ADDA    z,d
         STOREA  x,d
         DECO    x,d
         CHARO   newLine,i
         DECO    y,d
         LOADA   z,d
         ORA     h#0030,i
         STBYTA  c,d
         CHARO   c,d
         STOP
         .END
```

What is the output of the above program if the user enters 42?

7. Convert the decimal number 3.375 into a fixed-point binary form, with four bits for each of the integer and fractional parts (*e.g.*, `0001.0000` represents the number 1.0).

   Convert the same number into a floating-point binary form, with a three-bit (excess 3) exponent and a four-bit significand.

   Compare the ranges (smallest and largest positive numbers) representable in the above two formats. Assume the usual conventions for handling signs, infinities, NaNs, and denormalized numbers.

8. Consider the following recursive function in C++:

```
int gcd(int a, int b)
{
    if (b == 0) return a;
    int r = a % b;
    return gcd(b, r);
}
```

   Sketch the layout of items in a stack frame for this function:

   If each `int` occupies 2 bytes, and memory addresses also take 2 bytes, then how many bytes will be needed on the runtime stack to handle the call `gcd(12, 20)`?