CS 201 – HW #2

Your Name: ____

Please print this out, write your answers ***CLEARLY***, *and turn in hardcopy.*

Perform these calculations without using a calculator!!!

QUESTION 1: Create a table showing the first 16 binary numbers, their decimal values, and their representations as a hex numeral.

Binary	Decimal	Hex
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________
____________	____________	____________

QUESTION 2: Convert the following binary numbers to hex.

0110 0011 0101 1100 1001 0001 0111 1111 : ________________

0000 0010 0100 0110 1000 1010 1100 1110 : ________________

1111 1101 1011 1001 0111 0101 0011 0001 : ________________

QUESTION 3: Convert the following hex numbers to binary.

A6C2 : ____________________________________

80 : ____________________________________

00000001 : ____________________________________

ffff : ____________________________________

87654321 : ____________________________________

9ABCDEF0 : ____________________________________

QUESTION 4: How many bits in a byte? ______

QUESTION 5: How many bytes in a

8-bit quantity? ______

16-bit quantity? ______

32-bit quantity? ______

64-bit quantity? ______

QUESTION 6: Show an arbitrary value *in binary*. (Just make up values; what counts is the number of bits.)

1 byte quantity: ____

2 byte quantity: ____

4 byte quantity: ____

QUESTION 7: Show an arbitrary value *in hex*. (Just make up values; what counts is the number of numerals.)

64-bit quantity: ______________________________

32-bit quantity: ______________________________

16-bit quantity: ______________________________

8-bit quantity: ______________________________

QUESTION 8: Create a table of powers of 2

2^0: __________

2^1: __________

2^2: __________

2^3: __________

2^4: __________

2^5: __________

2^6: __________

2^7: __________

2^8: ______________

2^9: ______________

2^{10}: ______________

2^{16}: __________________

2^{32}: ____________________________

QUESTION 9: Convert the following binary numbers to decimal:

`10` : ___________

`100` : ___________

`1000` : ___________

`10000` : ___________

`1100` : ___________

`10101010` : ___________

`100` : ___________

`01110` : ___________

QUESTION 10: Convert the following decimal numbers to binary. Show each as a 2 byte quantity!

13: ______________________________

32: ______________________________

256: ______________________________

486: ______________________________

6,831: ______________________________

89: ______________________________

143: ______________________________

65,535: ______________________________

32,768: ______________________________

QUESTION 11: How many bits in a "C" language variable of type **`char`**? ______

How many bytes? ______

QUESTION 12: How many bits in a "C" language variable of type **`short`**? ______

How many bytes? ______

QUESTION 13: How many bits in a "C" language variable of type **`int`**? ______ How many bytes? ______

QUESTION 14: How many bits in a "C" language variable of type "**`long long`**"? ______ How many bytes? ______

(HINT: "**`long long`**" is an abbreviation for "**`long long int`**". And "**`long`**" is an abbreviation for "**`long int`**". However, "**`long long int`**" and "**`long int`**" are not necessary the same size.)

QUESTION 15: How many bits in a "C" language variable of type **`float`**? ______

How many bytes? ______

QUESTION 16: How many bits in a "C" language variable of type **`double`**? ______

How many bytes? ______

QUESTION 17: Take the following bit strings and perform the bitwise logical AND operation.

```
0101 1100 1010 1111 0110 0110 0111 1011
1101 0110 0100 0011 0111 1001 1000 0011
```

Show the result in binary: ______________________________________

Show the result in hex: ______________________________________

QUESTION 18: Using the same values, perform the bitwise logical OR operation.

Show the result in binary: ______________________________________

Show the result in hex: ______________________________________

QUESTION 19: Using the same values, perform the bitwise logical XOR operation.

Show the result in binary: ______________________________________

Show the result in hex: ______________________________________

QUESTION 20: Add the following binary numbers.

```
0101 1010 0110 1011
0100 1100 0111 1010
```

Show the result in binary: ______________________________

QUESTION 21: Add the following 16-bit numbers, which are given in hex. (There are two approaches: convert the numbers to binary first or perform the addition entirely in hex. It's a good way to check your answer.)

```
3D3A
5DD9
```

Show the result in hex: ______________

QUESTION 22: Assuming two's complement representation (i.e., "signed" numbers), convert the following decimal values to 8-bit binary values and show in…

Binary **Hex**

0: ______________________________ __________

+1: ______________________________ __________

+2: ______________________________ __________

+126: ______________________________ __________

+127: ______________________________ __________

-1: ______________________________ __________

-2: ______________________________ __________

-127: ______________________________ __________

-128: ______________________________ __________

QUESTION 23: Assuming two's complement representation (i.e., "signed" numbers), convert the following decimal values to 16-bit binary values and show in…

Binary **Hex**

0: ______________________________ __________

+1: ______________________________ __________

+2: ______________________________ __________

+32766: ______________________________ __________

+32767: ______________________________ __________

-1: ______________________________ __________

-2: ______________________________ __________

-32767: ______________________________ __________

-32768: ______________________________ __________

QUESTION 24: Assuming two's complement representation (i.e., "signed" numbers), convert the following 8-bit binary values to decimal:

00000000: __________

00000001: __________

00000010: __________

01111110: __________

01111111: __________

10000000: __________

10000001: __________

11111110: __________

11111111: __________

QUESTION 25: Assuming two's complement representation (i.e., "signed" numbers), convert the following 16-bit values (shown in hex) into decimal:

0000: ________________

0005: ________________

0007: ________________

7ffe: ________________

7fff: ________________

8000: ________________

8001: ________________

fffe: ________________

ffff: ________________

QUESTION 26: Describe Big Endian.

____

____

QUESTION 27: Describe Little Endian.

____

____

QUESTION 28: How many bits are used for addresses (i.e., pointers) on the IA32 architecture? __________ On the x86-64 architecture? __________

QUESTION 29: What does the following x86-64 instruction do?

```
     movq    $1234,%rax
```

____

____

QUESTION 30: What does the following x86-64 instruction do?

```
     movq     1234,%rax
```

QUESTION 31: What do the following x86-64 instructions do to register `%rax`? Be careful to specify what happens to all bits.

```
     movq     $123,%rax
```

```
     movl     $123,%eax
```

```
     movw     $123,%ax
```

```
     movb     $123,%al
```

```
     movb     $123,%ah
```

QUESTION 32: What does the following x86-64 instruction do:

```
     addq     %rbx,%rdx
```

QUESTION 33: Write an instruction to add the contents of 16-bit register `%dx` to `%cx` and place the result in `%dx`:

____

QUESTION 34: Write an instruction to add the contents of 64-bit register `%rsi` to `%rdi` and place the result in `%rdi`:

____

QUESTION 35: Assume `j` in an "`int`" and is stored in `%edx`. What instruction will perform this "C" assignment statement?

```
    j = 123;
```

____

QUESTION 36: Assume `k` in a "`long long`" and is stored in `%rbx`. What instruction will perform this "C" assignment statement?

```
    k = 123;
```

____

QUESTION 37: Consider these "C" declarations; how many bytes are used to store the variables?

```
    int i;                          ________
    long long k;                    ________
    char myCh;                      ________
    int myArry [8];                 ________
    logn long yourArry [10];        ________
    char * p;                       ________
    int * p3;                       ________
    char * myA [4];                 ________
```

QUESTION 38: Here are signed numbers, shown in hex. The size of each quantity is obvious from the number of hex numerals. Negate each quantity and show the result in hex.

00000000: ______________________________

0001: ______________________________

034B: ______________________________

AB: ______________________________

7FFFFFFF: ______________________________

80000000: ______________________________

1234: ______________________________

1AC8 F203 20B4 4957: ______________________________

FF: ______________________________

00000001: ______________________________

QUESTION 39: Here are some binary fractions. What is the number in decimal. (Please write your answer in this form: $4^3/_4$, not 19/4 or 4.75)

0.1: __________

1.01: __________

11.001: __________

0.0001: __________

111.011: __________

110.111: __________

101.101: __________

101.1010: __________

101.10100000: __________

QUESTION 40: In the x86-64 architecture, there are 16 registers of 64 bits each. What are their names? (We use ATT notation, so don't forget the %.)

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

QUESTION 41: Show the ASCII codes for the following characters:

	Decimal	**Binary**	**Hex**
‘a’:	________	____________________	_________
‘A’:	________	____________________	_________
‘j’:	________	____________________	_________
‘J’:	________	____________________	_________
‘0’:	________	____________________	_________
‘3’:	________	____________________	_________
‘)’:	________	____________________	_________
‘ ’:	________	____________________	_________
‘\0’:	________	____________________	_________
‘\n’:	________	____________________	_________
‘\r’:	________	____________________	_________