Name: Steven Chu

UID:` 905- 094-800

Integer encoding. Assume we are running code on two machines using two's complement arithmetic for signed integers. Machine 1 has 4-bit integers and Machine 2 has 6-bit integers. Fill in the empty boxes in the table below. The following definitions are used in the table:

```
int x = -5;
unsigned ux = x;
```

Expression	4-bit decimal	4-bit binary	6-bit decimal	6-bit binary
-8	-8	1000	-8	111000
$-$TMin	-8	1000	-32	100000
$x >> 1$	-3	1101	-3	111101
$(-x\hat{}(-1)) >> 2$	-2	1110	-2	111110

Integer C Puzzles

Assume that x, y, and ux are initialized as follows:

int x = rand();

int y = rand();

unsigned ux = (unsigned) x;

Are the following statements always true? If false, provide a counter example.

(Note: "statement1 => statement2" means that if we are given statement1 is true, it implies that statement2 is also true.)

a. ~x + x >= ux True.

b. x > 0 => ((x << 5) >> 6) > 0 False. Ex: x = 00011010 (x << 5) >> 6 = 11111110

c. y < 0 => ux > y True.

d. $(ux*uy) == (x*y)$

True.

e. $((x\&8) \mid y) == y) \quad \Rightarrow \quad (x<<28)>0$

False. If (8 | y) == y, then ((x&8) | y) == y) is true, but (x << 28) > 0 is not.

f. $(x \hat{} y) \hat{} x + z == y+z$

True.

g. x >> 3 == x/3 False. x = 12, x >> 3 == 1, x/3 == 4.

Operand Form Practice (see page 181 in textbook for more)

Assume the following values are stored in the indicated registers/memory addresses.

Address	Value	Register	Value
0x104	0x34	%rax	0x104
0x108	0xCC	%rcx	0x5
0x10C	0x19	%rdx	0x3
0x110	0x42	%rbx	0x4

For each instruction, write the value stored in `%rdi` after it is executed:

Note: when a movl instruction is performed on %edi, the top 32 bits of %rdi are filled with zeros.

	Value		Value
a) `movl $0x110, %edi`	0x110	f) `leaq (%rax, %rcx), %rdi`	0x109
b) `movl %rax, %edi`	0x104	g) `movl 3(%rax, %rcx), %edi`	0x19
c) `movl (0x110), %edi`	0x42	h) `leaq 256(, %rbx, 8), %rdi`	0x124
d) `movl (%rax), %edi`	0x34	i) `movl 4(%rax, %rbx, 2), %edi`	0x42
e) `movl 4(%rax), %edi`	0xCC		

Endianness

a. Suppose we declared the following 4 byte int:

```
int x =309 ;
```

and we stored this in memory location 0x100 on a little-endian system. What values would be stored in the following memory locations?

0x100	0x101	0x102	0x103
00110101	00000001	00000000	00000000

b. Suppose we declared an array of ints:

```
int arr[] = {5, 8 };
```

and we stored this in memory location 0x100 on a little endian system. What values would be stored in the following memory locations?

0x100	0x101	0x102	0x103	0x104	0x105	0x106	0x107
0x05	0x00	0x00	0x00	0x08	0x00	0x00	0x00

c. Suppose we declared a string:

```
char * s = "hello";
```

and we stored this in memory location 0x100 on a little endian system. What values would be stored in the following memory locations?

note: it's a good idea to get familiar with hex encodings of alphabetical characters, but for convenience, the hexadecimal encodings of the characters are: h (0x68), e (0x65), l (0x6c), and o (0x6f)

0x100	0x101	0x102	0x103	0x104	0x105
0x68	0x65	0x6C	0x6C	0x6F	0x00