

# ICS Homework Week 6

October 23, 2019

## 1 Two's-Complement Encodings

Assume we have an integer type of **8 bits**.

### 1.1

Fill in the table below.

Value	2's Complement
-1	<b>1111 1111</b>
-37	<b>1101 1011</b>
<b>-83</b>	1010 1101
<b>85</b>	0101 0101

### 1.2

Please write the **numeric range** for the integer type with unsigned encoding(UMIN,UMAX) and 2's complement encoding(TMIN,TMAX) respectively.

**Unsigned Encoding: (0, 255)**

**2's Complement Encoding: (-128, 127)**

## 2 Type Casting

### 2.1

Assume we are using **64-bit machine**. Consider the following program.

```
1 char a = -9;
2 unsigned char b = a;
3 unsigned short c = a;
4 int d = a;
5 short e = 0xa251;
6 unsigned char f = e;
```

Write the value of variables below in decimal.

Variable	Value
a	-9
b	<b>247</b>
c	<b>65527</b>
d	<b>-9</b>
e	<b>-23983</b>
f	<b>81</b>

## 2.2

Assume int value is encoded using **16 bits**. Please fill in the following table with "<", ">" or "=". *U* means the constant is unsigned.

Constant A	Constant B	A ? B
-2U	-1U	<
-1	1	<
-1	100U	>
-1	65535U	=
-32767	32768U	>

## 3 Arithmetic Operations

### 3.1

Implement 2 functions with the following prototype. The function should determine whether arguments can be added without overflow. If arguments x and y can be added without casting overflow, return 1. Otherwise return 0.

```

1 int uadd_ok(unsigned x, unsigned y) {
2     return x + y >= x;
3     /* or (return x + y >= y;)*
4 }
5
6 int tadd_ok(int x, int y) {
7     return !((x<0==y<0)&&(x<0!=(x+y)<0));
8 }

```

### 3.2

Assume x and y are both 4 bit signed integers. Fill the following table. Truncate all the results to 4 bits with 2's complement and write their value in decimal.

	$x + y$	$x - y$	$x * y$	$-y$
$x = 4, y = 7$	-5	-3	-4	-7
$x = -6, y = -8$	2	2	0	-8
$x = 5, y = -1$	4	6	-5	1
$x = -3, y = 6$	3	7	-2	-6