

Signed vs Unsigned

Bits	Signed	Unsigned
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	-8	8
1001	-7	9
1010	-6	10
1011	-5	11
1100	-4	12
1101	-3	13
1110	-2	14
1111	-1	15

Diagram illustrating the mapping between signed and unsigned integer representations. The "Signed" column shows values from 0 to -1, and the "Unsigned" column shows values from 0 to 15. A double-headed arrow with "=" indicates that the bit patterns 0-7 map directly. A second double-headed arrow with "+/- 16" indicates that bit patterns 8-15 map to values 8-15 in the unsigned column, which are 16 units higher than their signed counterparts.

Project 1 Preview

- **sign(x)**: Given an int x , return 1 if x is positive, 0 if x is zero, and -1 if x is negative.
- No loops or conditionals!
- Allowed operators: ! ~ & ^ | + << >>
- **Hint**: First consider how to get -1 if x is negative and 0 otherwise. Then extend for 1 if positive.

Unsigned Bugs...

```
float sum_elements(float a[], unsigned length) {  
  
    int i;  
    float result = 0;  
  
    for (i = 0; i <= length - 1; i++) {  
        result += a[i];  
    }  
  
    return result;  
}
```