

Rounding Up

Prof. James L. Frankel
Harvard University

Version of 4:04 PM 27-Sep-2022
Copyright © 2022, 2021 James L. Frankel. All rights reserved.

Rounding Up

- To round *value* up to a multiple of *n*, perform

```
unsigned int value, n, rounded_up_value;  
rounded_up_value = ((value + (n-1))/n) * n;
```

```
if n == 8, and  
  if value == 7,  
    then result is 8  
  if value == 8,  
    then result is 8  
  if value == 9,  
    then result is 16
```

Rounding Up To a Power of Two

- To round *value* up to a multiple of *n*, where *n* is a power of two, perform

```
unsigned int value, n, rounded_up_value;  
rounded_up_value = (value + (n-1)) & ~(n-1);
```

if $n == 8$ (then, $n-1 == 7$ and $\sim(n-1) == 0xFFFFFFFF8$ on a 32-bit computer), and
if $value == 7$,
then, $value + (n-1) == 14$ (which is $0xE$) and result is 8
if $value == 8$,
then, $value + (n-1) == 15$ (which is $0xF$) and result is 8
if $value == 9$,
then, $value + (n-1) == 16$ (which is $0x10$) and result is 16

- When *n* is a power-of-two, this algorithm is more efficient (it doesn't require division and multiplication)