

15-213

“The course that gives CMU its Zip!”

Integers

Sep 3, 2002

Topics

- **Numeric Encodings**
 - Unsigned & Two's complement
- **Programming Implications**
 - C promotion rules
- **Basic operations**
 - Addition, negation, multiplication
- **Programming Implications**
 - Consequences of overflow
 - Using shifts to perform power-of-2 multiply/divide

C Puzzles

- Taken from old exams
- Assume machine with 32 bit word size, two's complement integers
- For each of the following C expressions, either:
 - Argue that is true for all argument values
 - Give example where not true

Initialization

```
int x = foo();  
int y = bar();  
unsigned ux = x;  
unsigned uy = y;
```

• $x < 0 \Rightarrow ((x*2) < 0)$

• $ux \geq 0$

• $x \& 7 == 7 \Rightarrow (x \ll 30) < 0$

• $ux > -1$

• $x > y \Rightarrow -x < -y$

• $x * x \geq 0$

• $x > 0 \&\& y > 0 \Rightarrow x + y > 0$

• $x \geq 0 \Rightarrow -x \leq 0$

• $x \leq 0 \Rightarrow -x \geq 0$ 15-213, F'02

C Puzzle Answers

- Assume machine with 32 bit word size, two's comp. integers
- *TMin* makes a good counterexample in many cases

<code>x < 0</code>	\Rightarrow	<code>((x*2) < 0)</code>	False: <i>TMin</i>
<code>ux >= 0</code>			True: $0 = UMin$
<code>x & 7 == 7</code>	\Rightarrow	<code>(x<<30) < 0</code>	True: $x_1 = 1$
<code>ux > -1</code>			False: 0
<code>x > y</code>	\Rightarrow	<code>-x < -y</code>	False: $-1, TMin$
<code>x * x >= 0</code>			False: 30426
<code>x > 0 && y > 0</code>	\Rightarrow	<code>x + y > 0</code>	False: <i>TMax, TMax</i>
<code>x >= 0</code>	\Rightarrow	<code>-x <= 0</code>	True: $-TMax < 0$
<code>x <= 0</code>	\Rightarrow	<code>-x >= 0</code>	False: <i>TMin</i>