

# ICS 2008

Instructor: Aleksandar Kuzmanovic

TA: Ionut Trestian

Recitation 1

# Project 1: Datalab

- **Part I: Bit manipulations**

Name	Description	Rating	Max Ops
<code>bitNor(x, y)</code>	$\sim (x   y)$ using only <code>&amp;</code> and <code>~</code>	1	8
<code>bitXor(x, y)</code>	$\wedge$ using only <code>&amp;</code> and <code>~</code>	2	14
<code>isNotEqual(x, y)</code>	$x \neq y$ ?	2	6
<code>getBytes(x, n)</code>	Extract byte <code>n</code> from <code>x</code>	2	6
<code>copyLSB(x)</code>	Set all bits to LSB of <code>x</code>	2	5
<code>logicalShift(x, n)</code>	Logical right shift <code>x</code> by <code>n</code>	3	16
<code>bitCount(x)</code>	Count number of 1's in <code>x</code>	4	40
<code>bang(x)</code>	Compute $\neg x$ without using <code>!</code> operator	4	12
<code>leastBitPos(x)</code>	Mark least significant 1 bit	4	30

- $X = 0011$     $Y = 1010$  (NOR?) (XOR?)
- $X \text{ NOR } Y$  is 0100
- $X \text{ XOR } Y$  is 1001

# Puzzle 1

- $X \text{ NOR } Y (\sim(x|y))$  using only  $\sim$  and  $\&$
- Hints: DeMorgan's Law
  - a rule of inference, **NOT**, **AND**, and **OR** operators
  - to distribute a **negative** to a **conjunction** or **disjunction**
- Example:
  - $\sim(A | B)$ : "It is not true that the class is boring or the newspaper is interesting."
  - $\sim A \& \sim B$ : "The class is not boring and the newspaper is not interesting."

# Part II: Two's Complement Arithmetic

- a way of representing positive and negative integers
- any bit pattern that has a sign bit of 0 is a ***positive*** number, on the other hand, sign bit 1 means ***negative***
- ***Positive***: convert it straight into decimal as you would convert a normal binary number
- ***Negative***: bit pattern needs to be converted out of 2's complement before you can convert it from binary into decimal.

# Two's complement using patterns of length 3

Bit pattern	Value represented
011	3
010	2
001	1
000	0
111	-1
110	-2
101	-3
100	-4

Name	Description	Rating	Max Ops
<code>tmax(void)</code>	largest two's complement integer	1	4
<code>isNonNegative(x)</code>	$x \geq 0$ ?	3	6
<code>isGreater(x, y)</code>	$x > y$ ?	3	24
<code>divpwr2(x, n)</code>	$x / (1 \ll n)$	3	15
<code>abs(x)</code>	absolute value	4	10
<code>addOK(x, y)</code>	Does $x+y$ overflow?	3	20

# Puzzle 1

- `tmax(void)`: largest two's complement integer
- just use's 2's complement max integer definition
- `~(1 << 31)` or `~(0x1 <<31)`

# How to check your code before handin

- dlc: a modified ANSI C compiler, check your programs for compliance with the coding style rules.
- What is style rules?
  - *straightline* code (i.e., no loops or conditionals)
  - a limited number of C arithmetic and logical operators.
- Usage: `./dlc bits.c`
- Demo

# How to check your code before handin (cont.)

- `btest`: evaluate the functional correctness of your code. Read the README!
- Usage: `./btest`
- Work through the functions one at a time
- `-f` flag to instruct `btest` to test only a single function, e.g., `./btest -f isPositive`.
- Demo

# See how you're doing

- “submit.pl” script
- <http://trestian.cs.northwestern.edu/dlcontest.html>