

Midterm 1 topics (in one slide)

The C language Functions, variables, and types

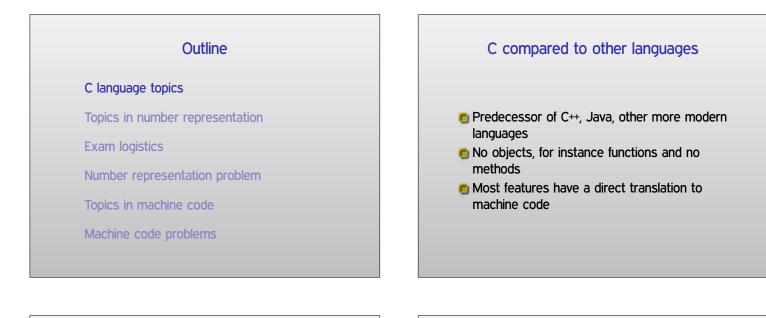
Branches and loops
Arrays, pointers, and structures

Number representation

- Bits and bitwise operators
- Unsigned and signed integers
- Floating point numbers

Machine-level code representation

- Instructions, operands, flags
- Conditions and branches



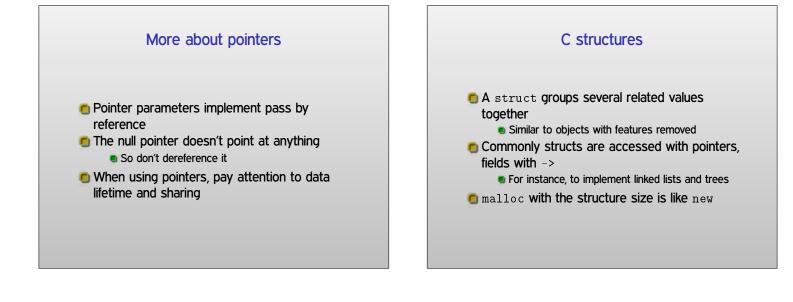
Kinds of variables and allocation C numeric types Local variables exist in one function execution, 🖲 char, short, int, and long are 8, 16, 32, or 64 bits on x86-64 and go away when it is over Even if you think you have a pointer to it! **I** Unsigned integers are > 0Global variables can be accessed from any Mixed operands upgraded to larger size and function, and last for the whole program unsigned **E For more control, allocate memory with malloc** 🖲 float and double are 32-bit and 64-bit and get a pointer floating point

C strings

- Instead of a real string type, C programs pass pointers to characters
- Usually, length of string is indicated by a \0 terminator
- Transform strings by writing loops over characters
- Programmer needs to be explicit about allocation and sharing

C pointers

- Pointers hold addresses, and the compiler knows their type
- Create a pointer to a variable with &
- Dereference a pointer with *
- Pointer arithmetic uses the element size, like an array
- In fact, a[x] is the same as *(a + x)



For instance, HA1 hashtable

Several possible designs:

- Array of pointers to list nodes
- Array of root structures pointing at list nodes
- Array of first list nodes (insert second)

Choices for string storage:

- Struct has char array, strcpy
- Struct has char pointer, strdup

Outline

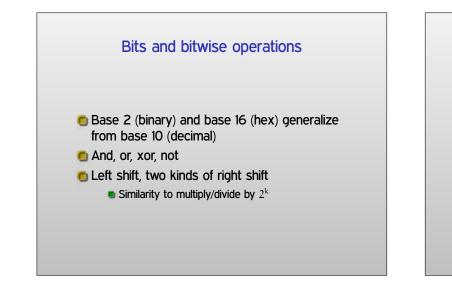
C language topics

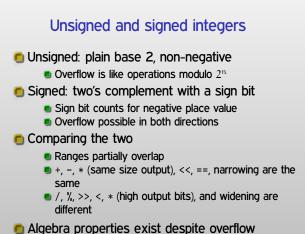
Topics in number representation

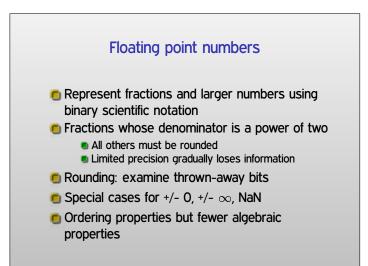
- Exam logistics
- Number representation problem

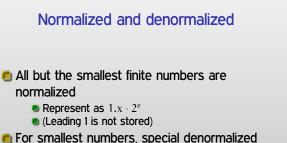
Topics in machine code

Machine code problems









- For smallest numbers, special denormalized form
 - Smallest exp encoding: same E as smallest normal
 - Leading 0 is not stored

Outline

C language topics

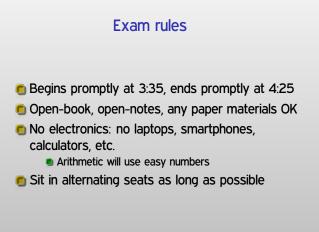
Topics in number representation

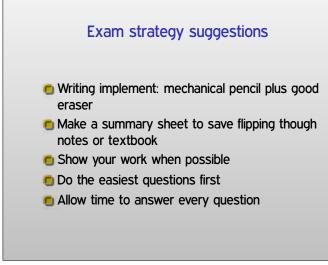
Exam logistics

Number representation problem

Topics in machine code

Machine code problems





Outline

C language topics

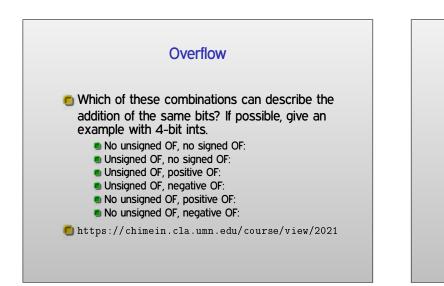
Topics in number representation

Exam logistics

Number representation problem

Topics in machine code

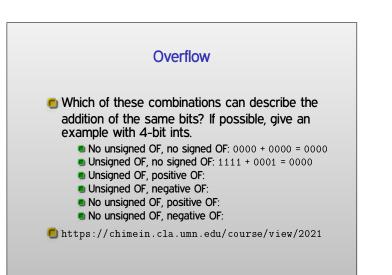
Machine code problems

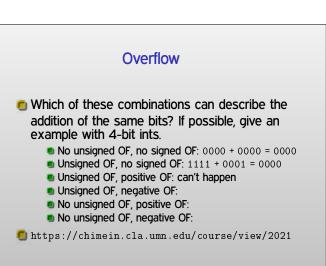


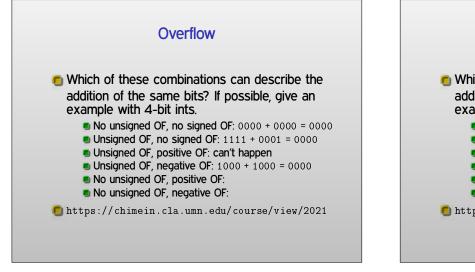
Overflow Which of these combinations can describe the addition of the same bits? If possible, give an example with 4-bit ints. No unsigned OF, no signed OF: 0000 + 0000 = 0000 Unsigned OF, no signed OF: Unsigned OF, positive OF: Unsigned OF, negative OF:

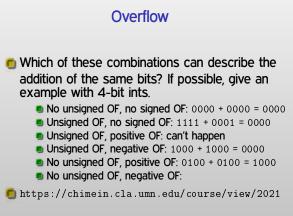
- No unsigned OF, positive OF:
- No unsigned OF, negative OF:

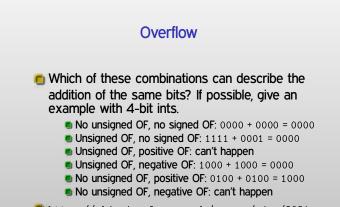
https://chimein.cla.umn.edu/course/view/2021











https://chimein.cla.umn.edu/course/view/2021

Outline

C language topics

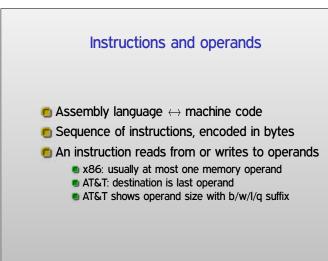
Topics in number representation

Exam logistics

Number representation problem

Topics in machine code

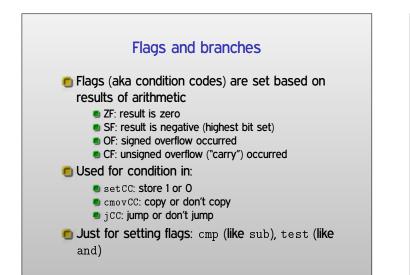
Machine code problems



Addressing modes

General form: disp(base,index,scale)

- Displacement is any constant, scale is 1, 2, 4 or 8
- Base and index are registers
- Formula: mem[disp + base + index scale]
- 🖲 All but base are optional
 - Missing displacement or index: 0
 - Missing scale: 1
 - Drop trailing (but not leading) commas
- Do same computation, just put address in register: lea



Outline

C language topics

Topics in number representation

Exam logistics

- Number representation problem
- Topics in machine code
- Machine code problems

	Working with ordering	
x < y y < x ! (x < y)	ese conditions are the same? x > y x <= y x >= y y > x y <= x y >= x !(x > y) !(x <= y) !(x >= y) !(y > x) !(y <= x) !(y >= x)	

Working with ordering

Which of these conditions are the same?

COI. 1	COI. 2	COI. 3	COI. 4			
А:х < у	B : x > y	C:x <= y	D:x >= y			
y < x	y > x	y <= x	y >= x			
!(x < y)	!(x > y)	!(x <= y)	!(x >= y)			
!(y < x)	!(y > x)	!(y <= x)	!(y >= x)			
https://chimein.cla.umn.edu/course/view/2021						

Working with ordering

Which of these	e conditions ar	e the same?	
Col. 1	Col. 2	Col. 3	Col. 4
A :x < y	B :x > y	C :x <= y	D:x >= y
B :y < x	A :y > x	D:y <= x	C:y >= x
D :!(x < y)	C :!(x > y)	B :!(x <= y)	A :!(x >= y)
C :!(y < x)	D :!(y > x)	A :!(y <= x)	B :!(y >= x)