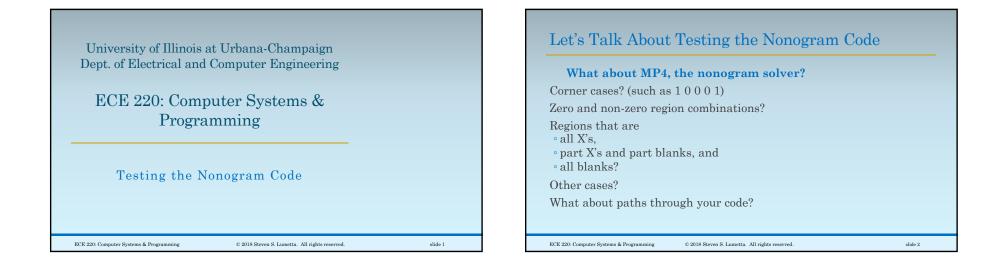
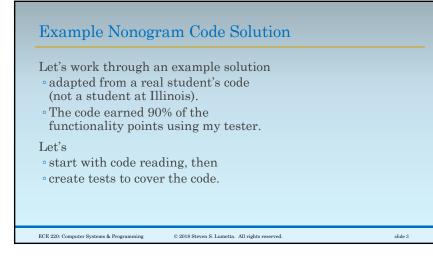
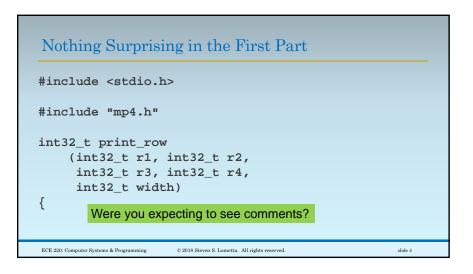
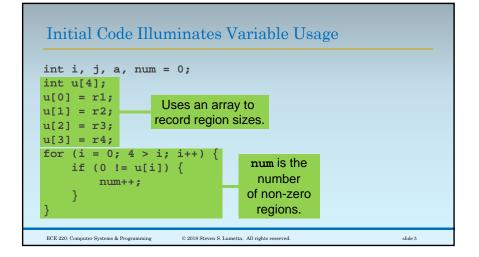
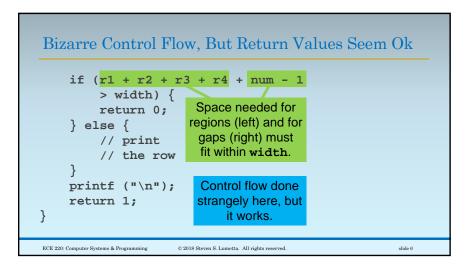
1

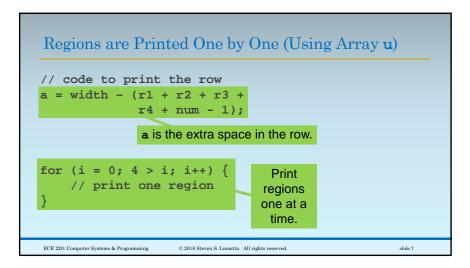




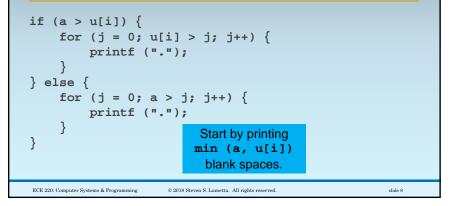


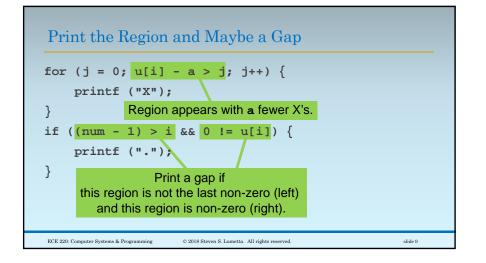


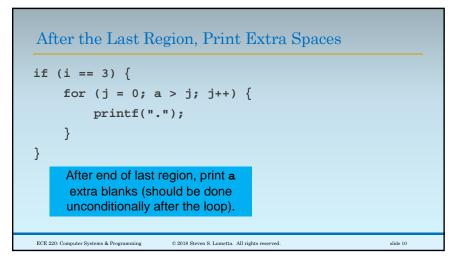


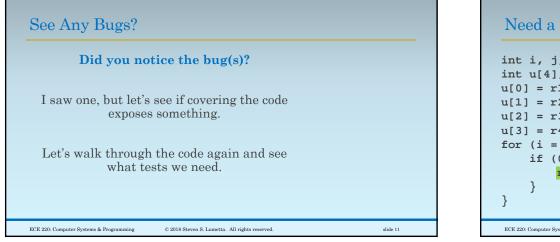


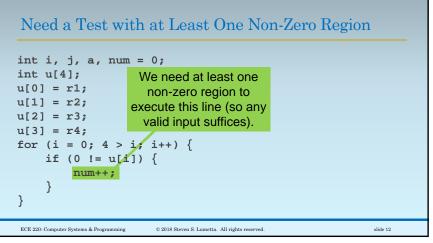
Start Each Region with Zero or More Blank Spaces

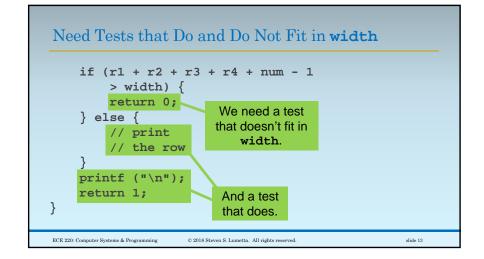


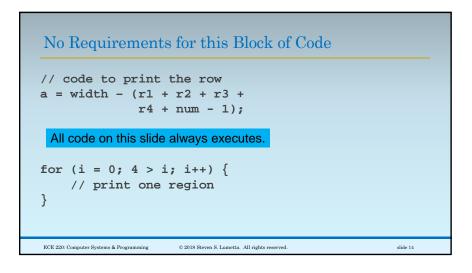


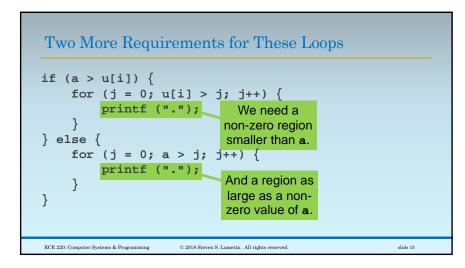


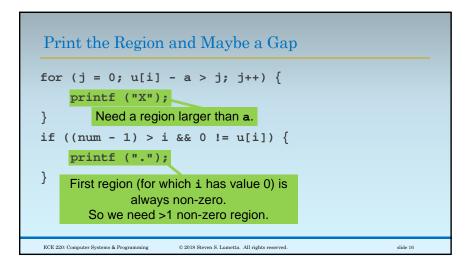


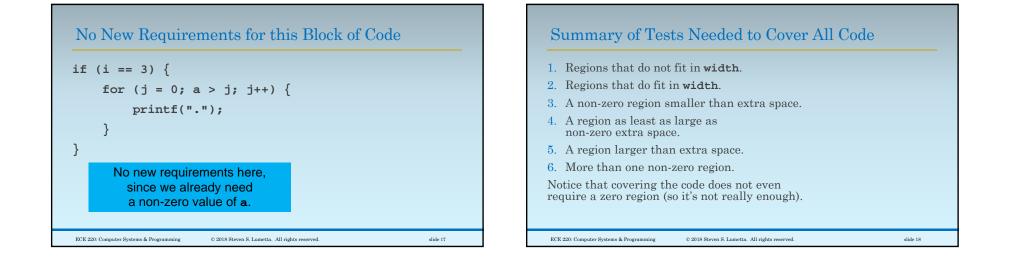












slide 19

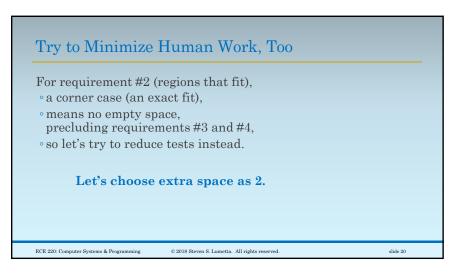


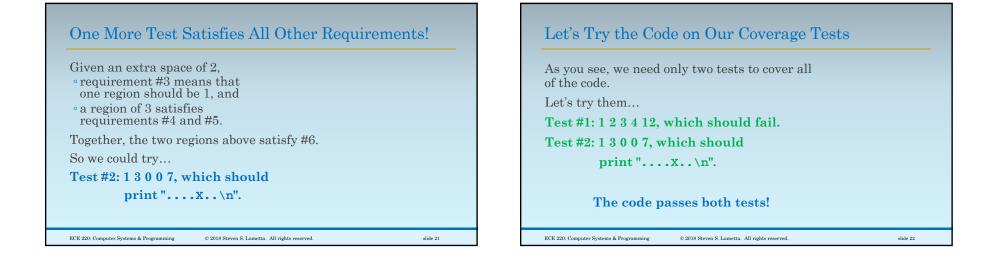
Try to **use corner cases**. For example, • for #1 (regions that do not fit in **width**), • let's make the regions 1 too large. • Say 1, 2, 3, and 4, which needs width 13, • so we'll set **width** to 12.

Test #1: 1 2 3 4 12, which should fail.

© 2018 Steven S. Lumetta. All rights reserved.

ECE 220: Computer Systems & Programming







When we see

- an **if** statement with an **else**,
- we cover both then and else blocks.

Did we cover else blocks that do nothing?

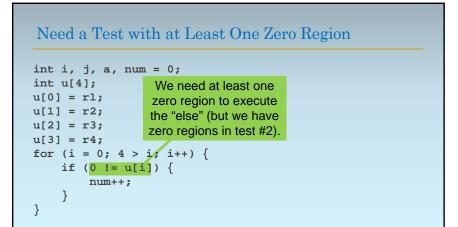
Let's take a look.

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved.

All rights reserved.

slide 23

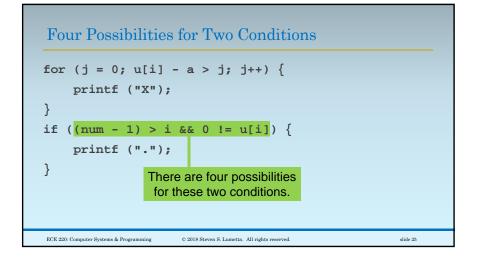
ECE 220: Computer Systems & Programming



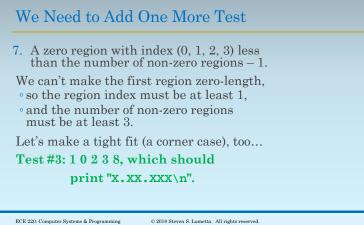
© 2018 Steven S. Lumetta. All rights reserved.

6

slide 24



Test #2: 1 3 0 0 7			
num – 1 > i	0 != u[i]	Test #2?	
false	false	regions 3 & 4	
false	true	region 2	
true	false	not covered	
true	true	region 1	



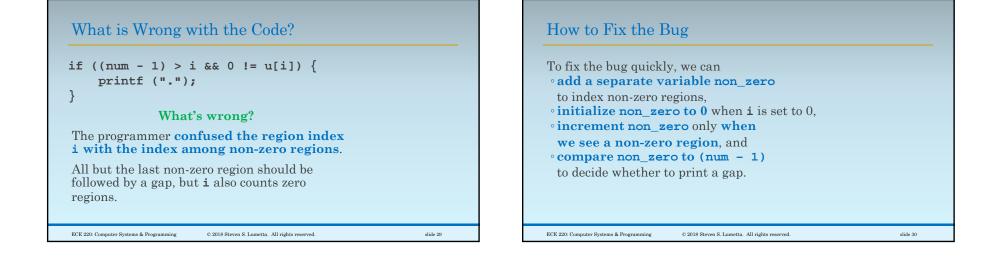
Let's Try the Code on Our Coverage Tests Let's try the last test... Test #1: 1 2 3 4 12, which should fail.

Test #2: 1 3 0 0 7, which should print "....X..\n". Test #3: 1 0 2 3 8, which should print "X.XX.XXX\n".

The code fails the third test!

slide 27

ECE 220: Computer Systems & Programming



Fixing the Bug

Here's how it might look
(except for declaration and initialization).

if (0 != u[i] &&
 (num - 1) > non_zero++) {
 printf (".");
}
With this change, the code passes
 all 6,391 of my tests as well.
EXE 202 Computer Systems & 2021 Sector All right reserved.

Fixing the Bug

