Objectives

- Lab 3 Feedback, Discussion
- Lab 4

Animation Demonstration

String Review

- How do you concatenate strings together?
- How do you repeat a string a certain number of times?
- How can we find out the length of a string?
- How can we "get" the first character of a string?
- How can we "get" the last character of a string? (2 ways)
- How can we "get" the last 3 characters of a string?
- What are the two ways to iterate through a string?
- How can we tell if some string is part of some other string?

Checking for Bad Input

```python
if not pickedNum.isdigit():
    print("Your number must contain only numbers")
    sys.exit()

# If we get to here, know the user's input is all digits

# User enters a number that is not four digits long
if len(pickedNum) != 4:
    print("Your number must contain four numbers")
    sys.exit()
```

```python
if not pickedNum.isdigit() or len(pickedNum) != 4:
    print("Your number must be in the format: ####")
else:
    # continue on, generating winning number, etc.
```

Lab 3 Feedback

- More difficult problems
- Took off for insufficient testing
- Took off for poor naming, style, comments

Common Issue: Inefficiency

```python
if patsScore > giantsScore:
    print("The Patriots win!")
else:
    if giantsScore > patsScore:
        print("The Giants win!")
    else:
        if patsScore==giantsScore:
            print("They tied! We're going to overtime!")
```
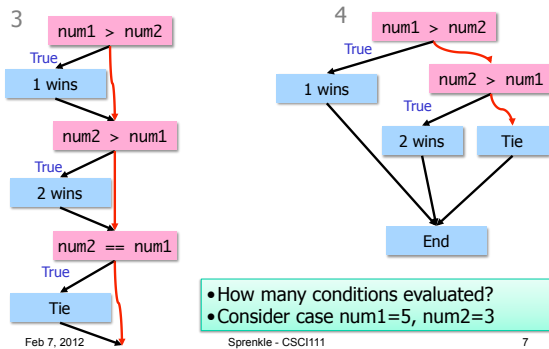
Extra if statement, not necessary
Know when hit second else that the only possibility is a tie

Slide 7

Problem 3, 4 Efficiency

3
- num1 > num2
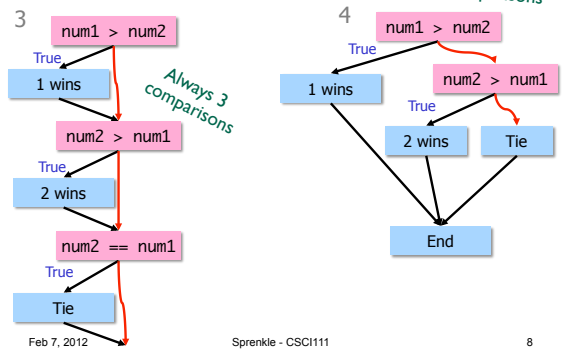 - True → 1 wins
- num2 > num1
  - True → 2 wins
- num2 == num1
  - True → Tie

4
- num1 > num2
  - True → 1 wins
- num2 > num1
  - True → 2 wins / Tie
- End

- How many conditions evaluated?
- Consider case num1=5, num2=3

Slide 8

Problem 3, 4 Efficiency

At most 2 comparisons

3
- num1 > num2
  - True → 1 wins
- num2 > num1
  - True → 2 wins
- num2 == num1
  - True → Tie

Always 3 comparisons

4
- num1 > num2
  - True → 1 wins
- num2 > num1
  - True → 2 wins / Tie
- End

Slide 9

Problem 4 Efficiency

Which tends to be more efficient?
How many conditions to evaluate?

- num1 == num2
  - True → Tie
- num2 > num1
  - True → 2 wins / 1 wins
- End

- num1 > num2
  - True → 1 wins
- num2 > num1
  - True → 2 wins / Tie
- End

Slide 10
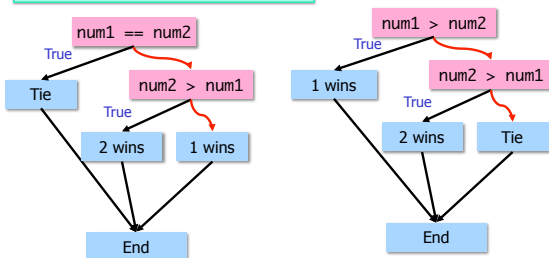
Common Issue

- AFC_TEAM=str("Patriots")
  - Redundant
  - Similar to
    - x=int(1)

- AFC_TEAM="Patriots"
  - The right hand side is a string

Slide 11

Common Issue

- Using same variable for different purposes
  - Will cause problems/confusion later

```
game = 0
for x in range(NUMSIMS+1):
   x = random.randint(MIN, MAX)
   game += 1
```

The purpose of x has changed:
   x is the counter for the loop
   x is the random number for determining who won

Separate purposes → separate variables

Another issue: don't need the *game* variable...

Slide 12

Common Issue

- Instead of

```
game = 0
for x in range(NUMSIMS+1):
   x = random.randint(MIN, MAX)
   game += 1
```

- This is better:

```
for game in range(NUMSIMS+1):
   outcome = random.randint(MIN, MAX)
```

Common Issue: Inefficient Code

```
if randNum > 0:
    AFCwins += 1
else:
    NFCwins += 1
if randNum > 0:
    print("Simulated game", (i+1), AFC_TEAM,"won")
else:
    print("Simulated game", (i+1), NFC_TEAM,"won")
```

```
if randNum > 0:
    AFCwins += 1
    print("Simulated game", (i+1), AFC_TEAM,"won")
else:
    NFCwins += 1
    print("Simulated game", (i+1), NFC_TEAM,"won")
```

Extensions

- Simulating scores

Lab 4

- 6 problems
- Due at the end of today's lab

- Questions about exam?