CSc 110, Autumn 2017
Lecture 16: Fencepost Loops and Review
Adapted from slides by Marty Stepp and Stuart Reges
A deceptive problem...

- Write a method `print_letters` that prints each letter from a word separated by commas.

For example, the call:

```
print_letters("Atmosphere")
```

should print:

```
A, t, m, o, s, p, h, e, r, e
```
Flawed solutions

• def print_letters(word):
    for i in range(0, len(word)):
        print(word[i] + ',', end='')
    print()  # end line

• Output: A, t, m, o, s, p, h, e, r, e,

• def print_letters(word):
    for i in range(0, len(word)):
        print('', '' + word[i], end='')
    print()  # end line

• Output: , A, t, m, o, s, p, h, e, r, e
Fence post analogy

• We print $n$ letters but need only $n - 1$ commas.
• Similar to building a fence with wires separated by posts:
  • If we use a flawed algorithm that repeatedly places a post + wire, the last post will have an extra dangling wire.

\[
\text{for length of fence:}
\begin{align*}
\text{place a post.} \\
\text{place some wire.}
\end{align*}
\]
Fencepost loop

• Add a statement outside the loop to place the initial "post."
  • Also called a fencepost loop or a "loop-and-a-half" solution.

place a post.
for length of fence – 1:
  place some wire.
  place a post.
Fencepost function solution

• def print_letters(word):
  
  print(word[0])
  for i in range(1, len(word)):
      print(", ", word[i], end=’’)
  print() # end line

• Alternate solution: Either first or last "post" can be taken out:

  def print_letters(word):
      for i in range(0, len(word) - 1):
          print(word[i] + ", ", end=’’)
      last = len(word) - 1

      print(word[last]) # end line
Fencepost question

• Write a function `print_primes` that prints all prime numbers up to a max.
  
  • Example: `print_primes(50)` prints
    2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47
  
  • If the maximum is less than 2, print no output.

• To help you, write a function `count_factors` which returns the number of factors of a given integer.
  
  • `count_factors(20)` returns 6 due to factors 1, 2, 4, 5, 10, 20.
Fencepost answer

# Prints all prime numbers up to the given max.
def print_primes(max):
    if (max >= 2):
        print("2", end='')
        for i in range(3, max + 1):
            if (count_factors(i) == 2):
                print("", " + str(i))
        print()

# Returns how many factors the given number has.
def count_factors(number):
    count = 0
    for i in range(1, number + 1):
        if (number % i == 0):
            count += 1  # i is a factor of number
    return count
Review question

• Write a function `random_triangle` that prints a triangle of the passed in string that is random height between 1 and 10. It should return the total number of stars printed.

  • Example: `random_triangle("*")` might print
    *
    **
    ***
    ****
    In this case it would return 10
Review question

• Write a function `average_temp` that prompts the user for temperatures and prints the average. The average should be rounded to one number after the decimal point.

  • Example: `average_temp()` might print
    
    How many temperatures? 4
temperature? 92
temperature? 90
temperature? 85
temperature? 95
Average temperature: 90.5