15-110 Recitation Week 4

Reminders

- HW2 due Monday 2/12 @ Noon
- Homework 2 Small Groups
- Recitation feedback form: https://forms.gle/dWgvmGvTSMbRi7rv5

Overview

- While loops
- For Loops
- Strings

Problems

While Loops vs. For Loops + Q/A

For loops! These allow you to set a specific range of values to iterate through ahead of time:

- General format: for i in range(x, y, z):
    - i → loop variable - contains current value of iteration
        - Can use any variable name here as long as we keep it consistent
    - range(x,y,z) → start value, end value not inclusive, step size
        - Start value is inclusive, end value is exclusive

While loops! These allow you to set a certain condition under which we keep iterating.

- General format: while (condition):
    - Condition will generally be some boolean expression. As long as this expression evaluates to True, we continue to iterate.

○   Make sure the expression evaluates to False at some point otherwise we end up in an infinite loop!

For loop example:

```
for i in range(0,10,2):
    print(i)
```

Now recall an example of a similar while loop:

```
i = 0
while i <= 10:
    print(i)
    i += 2
```

Note: <u>these two examples are not equivalent!</u> The for loop will not print 10, need to increase the end value to greater than 10. Also after the while loop, the value of i will be 12, whereas after the for loop (without changing the end value), the value of i will be 8.

While vs For Loops:

● For loops are used for a fixed number of iterations, help you avoid infinite loops
● While loops require declaring an iterator variable outside the loop and updating that variable within the loop. That is abstracted away in for loops by using range()
● While loops are more versatile, condition statement gives you more flexibility
● You can write any for loop as a while loop!

For + While Loop Code Tracing

Open the week 4 starter file and run the function `nestedFor` using the function calls provided in the file. Pay close attention to the order of the print statements as that tells you the flow of execution through the two loops!

Next, code trace through the function `mystery`. For this function pay close attention to the condition we are checking for in the while loop and the update condition that we're using. Try to come up with a general idea of what the function is doing by looking at when the loop is stopping and what do we do inside of the loop.

Quick String Questions

Given the string `s = "abcdefghi"`, answer the following short answer questions:

1) How do I access the character "i" from string s?

2) How do I create a string x which is equal to "cdef" using s?

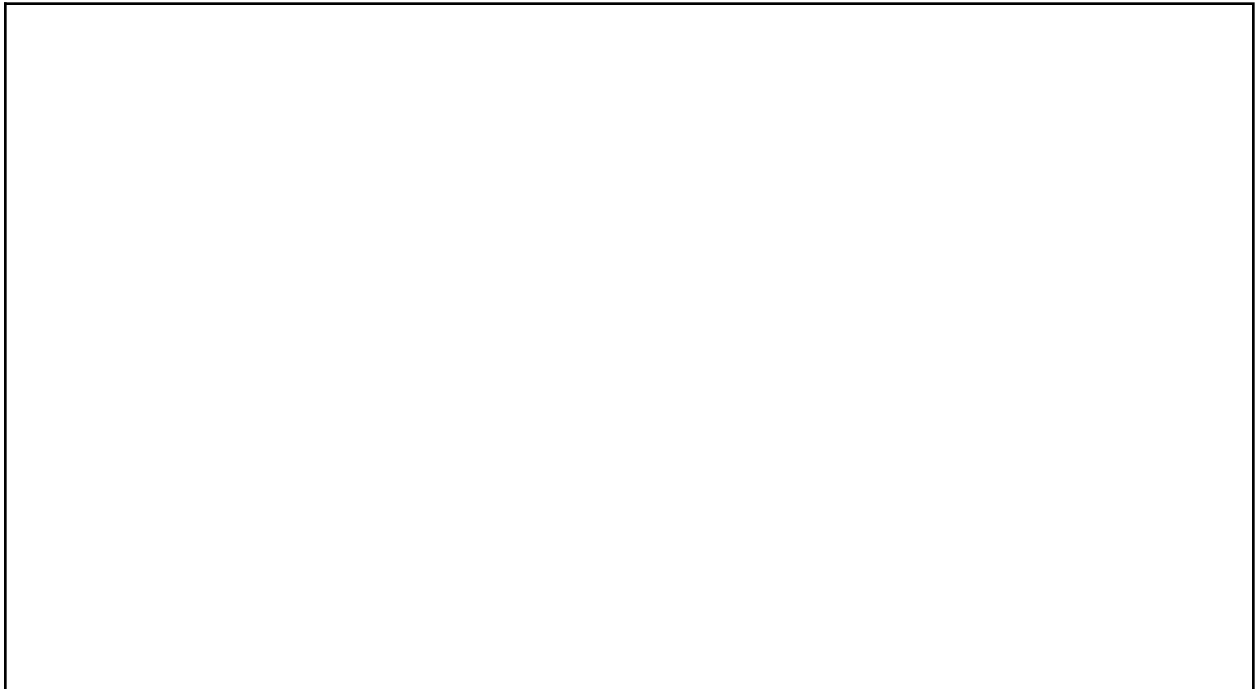3) How do I create a string x which is equal to "beh" using s?

4) How do I create a string x which is equal to the reverse of s?

Strings Functions Code Writing

Write a function `isPalindrome(s)` that takes in a string and returns `True` if it is a palindrome and `False` if it is not. Recall that a palindrome is a string that is the same front-to-back as it is back-to-front.

Ex: `isPalindrome("level")` should return `True`
Ex: `isPalindrome("lever")` should return `False`

Now, let's modify our `isPalindrome(s)` so that we can accept a wider variety of palindromes. We should ignore any whitespace, punctuation, and numbers.

Ex: `isPalindrome("a man, a plan, a canal, panama!")` should return `True`
Ex: `isPalindrome("a santa at nasa.")` should return `True`
Ex: `isPalindrome("Just some random string.")` should return `False`