COSI 10A Recitation #5

Agenda: Loops & Indexing

Review: Loops

Categories of loops

definite loop: Executes a known number of times.

- The `for` loops we have seen are definite loops.
- Print "hello" 10 times.
- Find all the prime numbers up to an integer *n*.
- Print each odd number between 5 and 127.

indefinite loop: the number of times its body repeats is not known in advance

- Prompt the user until they type a non-negative number.
- Print random numbers until a prime number is printed.
- Repeat until the user has typed "q" to quit.

Review: Loops

Ways to Create range

Range form	Description	Example	Numbers in range
`range(stop)`	*Range from 0 (inclusive) to stop (exclusive)*	`range(5)`	0, 1, 2, 3, 4
`range(start, stop)`	*Range from start (inclusive) to stop (exclusive)*	`range(3, 7)`	3, 4, 5, 6
`range(start,stop,step)`	*Range from start (inclusive) to stop (exclusive), increasing by step each time*	`range(4, 22, 3)`	4, 7, 10, 13, 16, 19

Review: Loops

Modify-and-assign operators

shortcuts to modify a variable's value

Shorthand	Equivalent longer version
variable += value	variable = variable + value
variable -= value	variable = variable - value
variable *= value	variable = variable * value
variable /= value	variable = variable / value
variable //= value	variable = variable // value
variable %= value	variable = variable % value

```
x += 3                  # x = x + 3
point -= 0.5            # point = point - 0.5
number *= 2             # number = number * 2
```

Review: Indexing

Index

Characters of a string are numbered with **0-based** indexes

```
name = "Ultimate"
```

index	0	1	2	3	4	5	6	7
	-8	-7	-6	-5	-4	-3	-2	-1
character	U	l	t	i	m	a	t	e

First character's index : 0

Last character's index : 1 less than the string's length

Review: Indexing

Accessing characters

You can access a character with **string[index]**:

```
name = "Merlin"
print(name[0])
```

Output: `M`

finding `String` length

```
length = len(string)


s = "Merlin"
count = len(s)     # 6
```

Review: Indexing

Looping through a string

The `for` loop through a string using range:

```
major = "CS"
for letter in range(0, len(major)):
    print(major[letter])
```

You can also use a `for` loop to print or examine each character without range.

```
major = "CS"
for letter in major:
    print(letter)
```

Output:

```
C
S
```

Problem 1

1. Write a Python program which repeatedly asks the user to enter a number n and then repeatedly applies the following rules until n == 1.

if n is divisible by 2 then divide it by 2

otherwise if n is divisible by 3 then divide it by 3

otherwise multiply it by 5 and add 1

Example Output:

```
enter n: 7
36 18 9 3 1 more? (y or n): y
enter n: 11
56 28 14 7 36 18 9 3 1 more? (y or n): y
enter n: 13
66 33 11 56 28 14 7 36 18 9 3 1 more? (y or n): n
```

Extra Credit: “invalid input” should be printed for the “more” input and the loop should stop:

```
enter n: 7
36 18 9 3 1 more? (y or n): y
enter n: 11
56 28 14 7 36 18 9 3 1 more? (y or n): japwodjopqwd
invalid input
```

https://codeshare.io/NKNKrr

1. Write a Python program which repeatedly asks the user to enter a number n and then repeatedly applies the following rules until n == 1.

if n is divisible by 2 then divide it by 2

otherwise if n is divisible by 3 then divide it by 3

otherwise multiply it by 5 and add 1

```python
def process_number(n):
    while n != 1:
        if n % 2 == 0:
            n //= 2
        elif n % 3 == 0:
            n //= 3
        else:
            n = n * 5 + 1
        print(n, end=' ')

def main():
    run = 'y'
    while run == 'y':
        n = int(input("enter n: "))
        process_number(n)
        run = input("more? (y or n): ")

main()
```

```python
def process_number(n):
    while n != 1:
        if n % 2 == 0:
            n //= 2
        elif n % 3 == 0:
            n //= 3
        else:
            n = n * 5 + 1
        print(n, end=' ')

def main():
    run = 'y'
    while run == 'y':
        n = int(input("enter n: "))
        process_number(n)
        run = input("more? (y or n): ")
    if run != 'n':
        print("invalid input")

main()
```

Problem 2

2. A certain bank offers 4.15% interest on savings accounts, compounded annually. Create a table that shows how much money a person will accumulate over the number of years that they want to save their money for. Assume that the person makes an initial investment of $1000 and deposits $100 each year after the first. Your table should indicate for each year the current balance, the interest, the new deposit, and the new balance. For example:

```
For how many years do you want to save? 4
Year        Curr Balance     Interest    New Deposit      New Balance
0           1000.0           41.5        100.0            1141.5
1           1141.5           47.37       100.0            1288.87
2           1288.87          53.49       100.0            1442.36
3           1442.36          59.86       100.0            1602.22
```

https://codeshare.io/NKNKrr

2. A certain bank offers 4.15% interest on savings accounts, compounded annually. Create a table that shows how much money a person will accumulate over the number of years that they want to save their money for. Assume that the person makes an initial investment of $1000 and deposits $100 each year after the first. Your table should indicate for each year the current balance, the interest, the new deposit, and the new balance. For example:

```
For how many years do you want to save? 4
Year        Curr Balance     Interest    New Deposit      New Balance
0           1000.0           41.5        100.0            1141.5
1           1141.5           47.37       100.0            1288.87
2           1288.87          53.49       100.0            1442.36
3           1442.36          59.86       100.0            1602.22
```

```python
def calculate_savings(years):
    balance = 1000
    interest_rate = 0.0415
    print(f"Year\tCurr Balance\tInterest\tNew Deposit\tNew Balance")
    for i in range(years):
        print(f"{i}\t{balance:.2f} \t{balance * interest_rate:.2f} \t\t100.0 \t\t{balance + balance * interest_rate + 100:.2f}")
        balance += balance * interest_rate + 100

def main():
    years = int(input("For how many years do you want to save? "))
    calculate_savings(years)

main()
```

Problem 3

Write a program that determines if a string is palindrome. A palindrome string is a string that reads the same backward as forward. For example: “ABA”, “madam”, “abba”. Prompt the user for a string and validate if the string is a palindrome. Your program should have an isPalindrome() function that returns True if the string is palindrome, False otherwise. You should continuously prompt the user until they enter an empty string. Assume the string the user entered has no punctuations and spaces.

```
(base) jameskong@JimmyK-PC:~/cosi10aTA/Recitation 5$ /home/jameskong/miniconda3/bin/python "/home/jameskong/cosi10aTA/Recitation 5/problem_3.py"
Enter a string to check if it is a palindrome (hit enter to stop): aba

aba is a palindrome.

Enter a string to check if it is a palindrome (hit enter to stop): madam

madam is a palindrome.

Enter a string to check if it is a palindrome (hit enter to stop): abBa

abBa is not a palindrome.

Enter a string to check if it is a palindrome (hit enter to stop): hOOh

hOOh is a palindrome.

Enter a string to check if it is a palindrome (hit enter to stop):
(base) jameskong@JimmyK-PC:~/cosi10aTA/Recitation 5$
```

https://codeshare.io/NKNKrr

Write a program that determines if a string is palindrome. A palindrome string is a string that reads the same backward as forward. For example: “ABA”, “madam”, “abba”. Prompt the user for a string and validate if the string is a palindrome. Your program should have an isPalindrome() function that returns True if the string is palindrome, False otherwise. You should continuously prompt the user until they enter an empty string. Assume the string the user entered has no punctuations and spaces.

```python
def isPalindrome(user_input):
    i = 0
    isPalindrome = True
    while i < len(user_input) and isPalindrome:
        if user_input[i] != user_input[len(user_input) - 1 - i]:
            isPalindrome = False
        i += 1
    return isPalindrome

def main():
    user_input = input("Enter a string to check if it is a palindrome (hit enter to stop): ")
    while user_input != "":
        if isPalindrome(user_input):
            print(f"\n{user_input} is a palindrome.\n")
        else:
            print(f"\n{user_input} is not a palindrome.\n")
        user_input = input("Enter a string to check if it is a palindrome (hit enter to stop): ")

main()
```

```python
def isPalindrome(user_input):
    for i in range(len(user_input)):
        if user_input[i] != user_input[len(user_input) - 1 - i]:
            return False
    return True
```

```python
def isPalindrome(user_input):
    palindrome = ""
    for i in range(len(user_input) - 1, -1, -1):
        palindrome += user_input[i]
    return user_input == palindrome
```

```python
def isPalindrome(user_input):
    return user_input == user_input[::-1]
```

Questions?