CSCI 127: Additional Programming Problems

Created By: Owen Kunhardt

The following problems are not for a grade or extra credit. They are for practice. There are no Gradescope submissions for these problems, you must test them yourself. They will require more problem solving and time than the assignments in class. These problems may require concepts you may not have yet learned, but will be covered later in the course. You can find hints after all the problems. Try the problem on your own before looking at the hints. You can write your solution in Python or C++. Should you have any questions or issues, please contact me at tutoring AT owenkunhardt DOT com.

1 Backwards String

Write a program that asks the user for a word and spells it backwards. A sample run of your program should look like:

```
Enter word to be spelled backwards: Hello, World!
!dlroW ,olleH
```

2 Palindrome

Write a program that asks the user for a word and determines if that word is a palindrome. A palindrome is a word that reads the same backwards as it does forward. Your program should not be case sensitive. Sample runs of your program should look like:

```
Enter a word: Kayak
Kayak is a palindrome.

Enter a word: Python
Python is not a palindrome.
```

3 Word Triangle

Write a program that asks the user for a word and prints a triangle where the first row is the first character of the word, the second is the first and second characters, ..., and the last row is all the characters of the word. A sample run of your program should look like:

```
Enter a word: Hello there

H
He
Hel
Hell
Hello
Hello
Hello t
Hello th
Hello the
Hello ther
Hello there
```

4 Half Pyramid

Write a program that asks the user for a height and then prints a half-pyramid based on that height. Your program should not accept a height greater than 20 or a negative number. Sample runs of your program should look like:

```
Enter height: 7

      *
     **
    ***
   ****
  *****
 ******
*******

Enter height: -1

Height must be not be negative or greater than 20.
Enter height: 21

Height must be not be negative or greater than 20.
Enter height: 3

  *
 **
***
```

5 Change Owed

Write a program that asks the user the amount of changed owed and prints out the minimal number of total coins, quarters, dimes, nickels, and pennies that it requires to give that amount of change. Your program should not accept a negative number. Sample runs of your program should look like:

```
Enter amount of change: 0.66

Coins:  5
Quarters:  2
Dimes:  1
Nickels:  1
Pennies:  1

Enter amount of change: -5.01

Change must not be negative.
Enter amount of change: 5.01

Coins:  21
Quarters:  20
Dimes:  0
Nickels:  0
Pennies:  1
```

Hints

General

In Python, to print without a newline you must put end="" in the print statement e.g. print("Hello, World!", end="").

Backwards String

Create a new empty string, then add the last character of the entered string, then the second to last character, ..., then add the first character.

Palindrome

Try to make use of the backwards string program you just wrote. Since a palindrome is defined as a word that is the same forwards as it is backwards, the forwards string should be equal to the backwards string.

Word Triangle

What is the relationship between row and number of characters of the entered string that are printed? Once you establish a relationship, try converting it to code.

Half Pyramid

What type of loop would you want to use if you are unsure how many times the loop may need to run? Try using that for checking if the user input meets the conditions.

What is the relationship between the current row, the height entered by the user, and how many spaces and asterisks are in that row? Consider how you would print the half pyramid without any spaces. How would you add the appropriate amount of spaces to each row? Once you establish a relationship, try converting it to code.

Change Owed

What type of loop would you want to use if you are unsure how many times the loop may need to run? Try using that for checking if the user input meets the conditions.

Try converting the change and coins to integers, it will make it easier to calculate and avoid any possible imprecision. Don't just type cast, how many cents equal one dollar?

You want to use the largest coin possible. Can you use a quarter? If yes, add 1 to the counter, subtract 25 cents from the change, and check again. If no and change is not 0, go to the next largest coin, a dime. Can you use a dime? If yes, add 1 to the counter, subtract 10 cents, and check again. If no and change is not 0, go to the next largest coin, a nickel. Can you use a nickel? If yes, add 1 to the counter, subtract 5 cents, and check again. If no and change is not 0, go to the next largest coin, a penny. Can you use a penny? If yes, add 1 to the counter, subtract 1 cent, and check again. If no, the remaining change must be 0. Think about why the remaining change must be 0 if you cannot use another penny. When the change is 0, you have the solution.