Homework_Module_03

April 5, 2025

1 Homework/Lab 3, DSE5002

Beija Richardson 4/5/2025 Created 2/11/2025

See

Think Python https://allendowney.github.io/ThinkPython/chap05.html

2 5.14.1. Ask a virtual assistant

Ask a virtual assistant, "What are some uses of the modulus operator?"

2.1 I asked Siri and it showed " The modulus operator is commonly used in programming to check if a numbber is divisible by another number or to perform operations that require a cyclic pattern." resources were used by tecadmin.net

In this chapter, we saw two ways to write an if statement with three branches, using a chained conditional or a nested conditional. You can use a virtual assistant to convert from one to the other. For example, ask a VA, "Convert this statement to a chained conditional."

if x == y: print('x and y are equal') else: if x < y: print('x is less than y') else: print('x is greater than y')

Copy and paste the code from the VA and figure out if it works. If it doesn't, fix it

2.2 Used ChatGPT : print('x and y are equal') if x == y else print('x is less than y') if x < y else print('x is greater than y') It needs a x assigned

[7]:
```
x=1
y=3
print('x and y are equal') if x == y else print('x is less than y') if x < y␣
 ↪else print('x is greater than y')
```

```
x is less than y
```

Ask a VA, "Rewrite this statement with a single conditional."

if 0 < x: if x < 10: print('x is a positive single-digit number.')

If this doesn't work, fix ti

2.3 Used ChatGPT: if 0 < x < 10: print('x is a positive single-digit number.') It works.

[15]:
```
if 0 < x < 10: print('x is a positive single-digit number.')
```

```
x is a positive single-digit number.
```

Here's an attempt at a recursive function that counts down by two.

def countdown_by_two(n): if n == 0: print('Blastoff!') else: print(n) countdown_by_two(n-2)

[24]:
```
def countdown_by_two(n):
    if n == 0:
        print('Blastoff!')
    else:
        print(n)
        countdown_by_two(n-2)
```

3 5.14.3. Exercise

If you are given three sticks, you may or may not be able to arrange them in a triangle. For example, if one of the sticks is 12 inches long and the other two are one inch long, you will not be able to get the short sticks to meet in the middle. For any three lengths, there is a test to see if it is possible to form a triangle:

If any of the three lengths is greater than the sum of the other two, then you cannot form a triangle. Otherwise, you can. (If the sum of two lengths equals the third, they form what is called a "degenerate" triangle.)

Write a function named is_triangle that takes three integers as arguments, and that prints either "Yes" or "No", depending on whether you can or cannot form a triangle from sticks with the given lengths. Hint: Use a chained conditional.

[26]:
```
def is_triangle(a, b, c):
    print("Yes") if a + b > c and a + c > b and b + c > a else print("No")
```

[28]:
```
is_triangle(7,8,9)
```

```
Yes
```

[30]:
```
is_triangle(16,24,68)
```

```
No
```