CS 116 TUTORIAL 2

MAKING DECISIONS IN PYTHON
REMINDERS

• Assignment 2 is due on Wednesday January 24th at 10AM

• Midterm is on February 26th, at 7 PM

• Come to office hours if you need help.
BOOLEANS (REVIEW FROM LAST WEEK)

- **Values:** True, False (Capitalization!)

- **Boolean Operations:**
  and, or, not

- **Relational Operators:**
  <, >, <=, >=, ==, !=

- **Example:** 5 < 6
CONDITIONALS

• Conditions:
  – if : to start a condition
  – elif : to continue a set conditions (optional)
  – else : to execute something if all other conditions in the set are not true (optional)
• Always make sure that you have return statements inside your conditions, as desired.

• Double-check that your conditions are in the correct order
QUESTION 1

Ensure you understand the results of calling:

• choices(8)
• choices(10)
• choices(100)
• choices(111)
• choices(250)
• choices(360)

```python
def choices(n):
    answer = 0
    if n % 2 == 0:
        answer = answer + 1
    if n % 3 == 0:
        answer = answer + 1
    elif n % 5 == 0:
        answer = answer + 1
    else:
        answer = 10 * answer
    if n % 10 == 0:
        answer = answer - 1
    if n % 4 == 0:
        answer = answer // 2
    else:
        answer = 2 * answer
    return answer
```
QUESTION 2

If you are given three sticks, you may or may not be able to arrange them in 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 is_triangle that consumes three positive integers \((s1, s2, \text{ and } s3)\) representing the lengths of three sticks and returns one of the following:

“No triangle exists" if no triangle can be built with the three sticks

“Degenerate triangle exists" if a degenerate triangle exists for sticks of these lengths

“Triangle exists" if a triangle can be made from the sticks
Fermat’s Last Theorem states that given positive integers $a$, $b$, and $n$, there exists no integer $c$ for which $a^n + b^n = c^n$ unless $n \leq 2$.

Although Fermat wrote the statement of this theorem in the margin of a book in 1637, it was not proven until 1995 (and not for lack of trying – thousands of incorrect proofs of the theorem were put forward before it was finally proven).
Write a function `fermat_check` that consumes four positive integers, \( a, b, c, \) and \( n; n \geq 2 \).

- If \( n = 2 \), and \( a^2 + b^2 = c^2 \), then your function should return “Pythagorean triple.”

- If \( n = 2 \), and \( a^2 + b^2 \) is not \( c^2 \), then your function should return “Not a Pythagorean triple”.

- If \( n > 2 \), and \( a^n + b^n = c^n \), then your function should return “Fermat was wrong!”, as you have found a counterexample to Fermat’s Last Theorem.

- Otherwise, your function should return “Not a counterexample”.
A perfect number is a positive integer that is equal to the sum of its proper positive divisors (i.e. the sum of its positive divisors excluding the number itself).

Write a function `is_perfect_num` consumes a positive integer `n`. The function returns True if `n` is a perfect number, False otherwise.

For example, `is_perfect_num(6)` => True (because 1+2+3 = 6, and 1, 2, and 3 are all the proper divisors of 6).