Homework Day 2 Instructions

Please complete problems 1-4 and *your choice* of one of problems 5 and 6 (though feel free to do/try both!).

Problem 1: Calculator Part 1

Your task is to write a script that accomplishes the following things:

- Asks the user to input two variables, and assign them to values x1 and x2
- Creates a variable x3 which takes the value:
    - -2, if the sum of x1 and x2 is less than -10
    - -1, if the sum of x1 and x2 is between 0 and -10
    - 0, if the sum of x1 and x2 is between 0 and 10
    - 1, otherwise
- Prints x3

Problem 2: Calculator Part 2

Your task is to write a script that accomplishes the following things:

1. Creates a first variable (with your choice of name) and assigns it the value 102.5
2. Creates a second variable which is equal to the square of the first variable.
3. Creates a third variable which is equal to 3 times the second variable plus the square root of the first variable.
4. Prints the statement: “My final answer is: “ followed by the third variable.

Challenge: can you use only one print statement to complete the last task?

Problem 3: Ibuprofen Problem

It is important for us to be able to understand what dosage of medicine should be given. To assist with this, the manufacturer has issued the following guidelines:

- Adult Dose for Fever: 200 to 400 mg orally every 4 to 6 hours as needed.
- Pediatric (6 months - 12 years) Dose for Fever:
  - For temperatures less than 102.5° F (39.2 C), 5mg/kg/dose orally every 6 to 8 hours as needed.
  - For temperatures 102.5° F (39.2 C) or higher, 10mg/kg/dose orally every 6 to 8 hours as needed.
  - Warning: **Do not exceed the maximum adult dosage.**
- Infant (less than 6 months) Dose for Fever: Do not use!

Write a script that prints the recommended amount (in mg) of ibuprofen that should be given to an individual based on inputs (you need to ask for inputs!). You can pick how many variables you will need and what to name them.

Challenge: try not to ask for things you don't need, like the weight of an adult!

Problem 4: EEG Research Problem

You work in a research lab where participants play a computer game while you record EEG signals. On each trial, you know how fast the participant answered in milliseconds (ms) and whether they got the question right or wrong.

Before you look at the EEG data, you want to separate the types of trials to use – the brain signal you're looking for only happens on correct trials. Additionally, you know that if the subject took less than 200ms to respond they didn't have time to actually do the task (impossibly fast) and that if they took more than 4000ms, they weren't paying close attention, so you also want to exclude the trials outside these bounds.

Write a script which starts with trial information (response speed in ms and whether they got the trial correct) and prints a boolean variable which tells you if a trial is good to use or not.

Problem 5: Triangles

A triangle (in 2D) is defined by a plane figure with three straight sides and three angles. There are many different types of triangles.

First, based on whether some of the sides/angles have equal length, a triangle can be categorized into the following:

1. Equilateral triangle : Three equal sides (and three equal angles).
2. Isosceles triangle : Two equal sides (and two equal angles).
3. Scalene triangle : No equal sides (and no equal angles).

Second, a triangle can also be categorized by the type of angle it has:

1. Acute triangle : All angles are less than 90 degrees.
2. Right triangle : Has a right angle (90 degrees).
3. Obtuse triangle : Has an angle more than 90 degrees.

Thus, a triangle can be described in two words, according to the above two rules. For example, a triangle can be "a scalene, obtuse triangle," or "an isosceles, right triangle."

Your job in this problem is to write a function which categorizes a triangle -- first as equilateral, isosceles, or scalene, and then as acute, right, or obtuse. Your function should use the following variables, each corresponding to the size of an angle, to make a decision: `angle_abc` , `angle_bca` , and `angle_cab` . Your function should then assign the name of the target as a string to the variable `description` , in the following form : '[A/An] [side-length-based category], [angle-based category] triangle' . For example, with `angle_abc = 46` , `angle_bca = 44` , and `angle_cab = 90` , your function should assign the value 'An isosceles, right triangle' to the variable `description` .

A few important notes:

First, there are combinations of two names that are not possible: 'an equilateral, right triangle,' and 'An equilateral, obtuse triangle'. Think about why! In fact, your script should return 'An equilateral triangle' if the triangle is equilateral, and should not return 'An equilateral, acute triangle'.

Second, there are combinations of angles that do not constitute a triangle—the three angles of a triangle must add up to 180 degrees. When this condition is not met, your script should assign the value 'Not a triangle' to `description` .

That said, many combinations of descriptions are possible (e.g. 'a scalene, obtuse triangle' or 'a scalene, right triangle'). It might be more efficient to compute the first and the latter parts of the description in separate strings and then append them, using the "+" operator that we talked about in class.

You can test your work by assigning different values to each of the angle variables and seeing what your program prints—make sure to test thoroughly to make sure your program catches every possible combination!

Example outputs:

```
angle_abc = 45
angle_bca = 45
angle_cab = 90
####
# Your code here!
####
# Printed output:
```

```
"An isosceles, right triangle"
```

```
angle_abc = 60
angle_bca = 60
angle_cab = 60
####
# Your code here!
####
# Printed output:
"An equilateral triangle"
```

```
angle_abc = 70
angle_bca = 70
angle_cab = 70
####
# Your code here!
####
# Printed output:
"Not a triangle"
```

Problem 6: Pokemon

In an alternate universe where Pokémon are real and Brown is just a color, you are an avid Pokémon trainer. Like always, you were spending your weekend scavenging for some of the rarest Pokémon rumored to be hiding around the Ratty and Wayland dorm. When you were about to pounce upon what looked like a Pikachu, one of the rarest known Pokémon in the universe, a hyperactive Golden Retriever, suddenly jumped out of nowhere, snatched your iPhone (which doubly serves as your Pokédex), and ran away.

You can continue with your life without a phone, but you miss your Pokédex. Being a creative Brown student, you decided to build a simple python script that can tell you whether your current target is a Pikachu or some other animal. This is not a simple task because there are other animals sharing some of the defining characteristics of a Pikachu. In this problem, you will use logic operations and conditionals to write a simple script that will categorize your target based on the following 4 ***boolean*** variables:

- Does your target have red cheeks? `red_cheeks`
- Does your target walk on four legs? `four_legs`
- Is it smaller than a large moving box? `small`
- Is your target furry and golden? `furry`

Providence is teeming with wildlife. And indeed, several of them are easily confused with a Pikachu and are thus often mistargeted and caught by amateur trainers. Here are some of the defining characteristics of a Pikachu as well as 4 other regular animals which are frequently confused with a Pikachu:

Your target is `a Pikachu` if:

1. It has red cheeks.
2. It walks on four legs.
3. It is smaller than a large moving box.
4. It is furry and golden.

Your target is `a cockatiel` if:

1. It has red cheeks
2. It does not walk on four legs
3. It is smaller than a large moving box.
4. It is not furry and golden.

Your target is `a hyperactive golden retriever` if:

1. It does not have red cheeks
2. It walks on four legs
3. It is not smaller than a large moving box.
4. It is furry and golden.

Your target is `a snub-nosed monkey` if:

1. It does not have red cheeks.
2. It does not walk on four legs.
3. It is smaller than a large moving box.
4. It is furry and golden.

Your target is `a Brown student` if:

1. It does not have red cheeks.
2. It does not walk on four legs.
3. It is not smaller than a large moving box.
4. It may or may not be furry and golden.

OR

1. It has red cheeks.
2. It walks on four legs.
3. It is not smaller than a large moving box.
4. It may or may not be furry and golden.

Your script should use the four variables (`red_cheeks, two_legs, small, furry`), which define the four observable characteristics of the target. Your script should assign the name of the target as a string (case and punctuation sensitive) to the variable `target` .

You can test your work by assigning different values of the observable characteristics as variables and running your code. For example, your script should assign `a cockatiel` to the variable `target` if:

- red_cheeks = 1
- four_legs = 0
- small = 1
- furry = 0

Of course, there could be other animals as well as other Pokémon on campus that are not listed above. Whenever the 4 variables don't match with the description of any of the known target categories, your script should assign `others` to the variable `target` .

Think you have what it takes to be the very best that no one ever was? Start coding up your Pokédex!