Lab: Practice with functions that return values and if-elif-else

For each of the situations below, write a function that solves the problem. Each function should use exactly the arguments and return values specified. *You should not use any input or print statements*. Test each function by calling it with various arguments from the Python shell (the window with the >>> prompt). Do not write any `main()` functions.

1. Write a function called `divide_tip` that takes three arguments: the total amount of a restaurant bill, the percent you want to tip (as an integer), and the number of people at the table. The function returns the amount each person needs to pay to cover the bill if it is split evenly among all the people.

   Example: `divide_tip(40, 15, 4)` means the total bill was $40, you're leaving a 15% tip, and it's being split 4 ways. The total amount to split therefore will be $46, which means each person pays $11.50. So this function call will return 11.5 (as a float).

   Hint: the function definition line should look something like this:
   ```
   def divide_tip(bill, percent, people):
   ```

2. Write a function called `winner` that calculates the winner of an election. The function takes four arguments: the name of the first candidate and how many votes they got, followed by the second candidate's name and their votes. The function compares the votes and returns either the name of the candidate who won, or the string "tie" if there's a tie.

   Example: `winner("David", 50, "Kristin", 65)` would return the string "Kristin."
   Example: `winner("Sally", 400, "Morgan", 400)` would return the string "tie."

   Hint: the function definition line should look something like this:
   ```
   def winner(name1, votes1, name2, votes2):
   ```

3. Write a function called `postage` that calculates the amount the post office will charge you to mail a large flat envelope using first-class mail, based on its weight. The charge is based on the envelope's weight in ounces: an envelope weighing one ounce will cost 90 cents to mail, and each additional ounce is an additional 20 cents. However, you can only mail envelopes that weigh 13 ounces or less. Your function will take an integer argument called ounces and return the amount in dollars (as a float) that you should be charged. However, if the ounces argument is 0 or less, or if it is greater than 13, the function should return 0.

   For example, `postage(1)` returns 0.90, `postage(2)` returns 1.10, `postage(3)` returns 1.30, and so on, up to `postage(13)` which returns 3.30.

   Hint: the function definition line should look something like this:
   ```
   def postage(ounces):
   ```

4. Write a function `final_grade` that calculates a person's final letter grade for this class. The function takes five integer arguments: the average of the programming project grades, the zyBook grade, the two midterm grades, and the final exam grade. These are weighted 30%, 10%, 18% each, and 24%, respectively. Return the final letter grade using the grading scale 90=A, 80=B, etc.

   Example: `final_grade(85, 100, 93, 82, 87)` would return "B" because their final numeric grade comes out to about 87.88.

   Hint: the function definition line should look something like this:
   ```
   def final_grade(programs, zybook, midterm1, midterm2, final_exam):
   ```

(continues on back)

5. Write a function called `age_today` that takes three arguments: a month, day of the month, and year, all as integers. These three arguments represent someone's birthday. The function calculates how old this person will be today and returns that age.

   Ex: `age_today(11, 2, 1995)` should return 19, because this person hasn't had their birthday yet in 2015.

   Ex: `age_today(7, 29, 1995)` should return 20, because this person **has** had their birthday this year.

   Hint: the function definition line should look like this:
   ```
   def age_today(month, day, year):
   ```

6. Write a function called `is_even` that takes an integer as an argument. The function returns the string "yes" if the integer is even, and "no" if the integer is odd. Hint: use the remainder operator (the percent sign).

7. Write a function called `timezone_diff` that takes two string arguments and determines how you would adjust your watch if you traveled between the time zones. The possible arguments are "eastern", "central", "mountain", and "pacific". The answer should be returned as a positive or negative integer (or zero) expressing how many hours difference there are between the two arguments. A positive return value means you set your watch forwards, and a negative value means you set your watch backwards.

   Ex: `timezone_diff("central", "eastern")` returns 1 because when you travel from the Central time zone to the Eastern, you set your watch forward one hour.
   Ex: `timezone_diff("eastern", "central")` returns -1 because going the opposite direction, you set your watch back one hour.

   Hint: There are sixteen different combinations of time zones here, but try not to do this with a sixteen-section if-elif-else statement. Instead, think about a way of doing this using *subtraction*.

8. Write a function called `first_string` takes three string arguments and returns the one that is first alphabetically. You can compare strings alphabetically using <, <=, >, and >=.

9. Write a function that accepts three arguments: a month, day of the month, and year, and returns the day of the week on which that date falls, as a string. Use Google to find an algorithm.

10. Write a function that accepts an integer and returns the Roman numeral for that integer. So `roman(9)` returns "IX". This is challenging.