HOMEWORK 3

Problem 1. Write a function called poly_perimeter that takes in two parameters, len_side, and num_sides, and returns the perimeter of the polygon. The perimeter of a polygon is the length of each side times the number of sides. For example, poly_perimeter(4,5) should return 20.

Problem 2. Create a function called get_hypotenuse that takes in two parameters, a and b, which represent the lengths of the two legs of a right triangle. The function should return the length of the hypotenuse. Use the Pythagorean theorem

$$\text{hypotenuse}^2 = a^2 + b^2,$$

to calculate and return the hypotenuse.

Problem 3. Write a function called find_distance that takes in four parameters, x_1, y_1, x_2, y_2, and returns the distance between these two points (x_1, y_1) and (x_2, y_2). The distance is given by the following formula which is a consequence of the Pythagorean theorem

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Problem 4. Write a function called total_string_length that takes in two parameters, str1 and str2, and returns the total length of these two strings.

Problem 5. Write a function called address that combines 3 different string address parameters (city, state, and zip) and returns a user's address. After the city and state inputs, add a comma and a space. For example,

```
address('Seattle', 'WA', '98105')
```

should return "Seattle, WA, 98105".

Problem 6. Write a function apply_discount(price, is_member) that applies a discount based on membership status:

- 10% discount for members.
- No discount for non-members.

For example

```
    apply_discount(100, True)
```

should return 90.

Problem 7. Write a function called shipping_price that takes two arguments: weight and express and returns shipping cost based on the weight of the package and whether express shipping is selected:

- Standard Shipping:
  - $5 for weights up to 2 lbs
  - $10 for weights above 2 lbs
- Express Shipping:
  - $10 for weights up to 2 lbs
  - $20 for weights above 2 lbs

For example

```
calculate_shipping_cost(4, True)
```

should return 20.

Problem 8. Write a function called final_price that takes three arguments: price, weight, and express and returns the final total price. The final price is the sum of the price and the shipping cost which is calculated as follows

- For purchase prices $100 or more, shipping is free.
- For purchase prices below $100, the shipping fee is calculated based on the formula from the previous problem.

Your function should call the shipping_price from the previous problem.

Problem 9. Write a function countdown_to_zero(start) that counts down from the given start number to 0, printing each number. If the starting number is less than or equal to 0, the function should print a message "Enter a positive number".

Problem 10. Write a function buzz(n) that prints the numbers from 1 to n. For multiples of 3, print "Buzz" instead of the number.

Problem 11. Write a function called negative_sum(a_list) that takes a list as an input. The function will return the sum of all negative numbers in the list. For example

```
negative_sum([-1, 2, -3])
```

should return -4.

Problem 12. Write a function called even_positive_sum(a_list) that takes a list of integers as an input. The function will return the sum of all positive even numbers in the list (a positive even number is a number that is both positive and even). For example

```
negative_sum([-1, 2, -4, 3, 6])
```

should return 8.

Problem 13. Write a function called maximal_element(a_list) that takes a list as an input. The function will return the maximal element in the list. For example

```
negative_sum([1, 5, 2, 4])
```

should return 5.

Problem 14. Write a function called is_prime(n) that checks whether a number is a prime number or not. A prime number is a number whose divisors are 1 and itself. For example: 2, 5, 7, and 11 are prime numbers while 6 is not.

Problem 15. Write a function called sum_divisors(n) that calculates the sum of all of the divisors of n.

Problem 16. Write a function called sum_square(n) that calculates the sum of the squares of all numbers between 1 and n. For example, sum_square(1) should return 1 and sum_square(2) should return 5.

Problem 17. An integer n is called a perfect square if $n = m^2$ for some integer n. For example, 4 is a perfect square because $4 = 2^2$. On other other hand, 6 is not a perfect square. Write a function called is_square(n) to check whether a number n is a perfect square.

Problem 18. Write a function that prints out all perfect square numbers between 1 and 100.

Problem 19. Write a function that takes a string as an input and returns True if this string contains the lowercase letter h. Otherwise, return False.