Assignments (Py01)

Problem 1 (Calculator)

Hard

Imagine you are building a calculator, which works with a speach to text system. The system can understand numbers from zero to twenty. The system can also understand the following operations: "plus", "minus", "multiply", and "divide". And a text to speech system that can convert the result back to a voice.

Write a Function that takes two word representations of numbers and an operation and returns the result of the operation as word representation.
The operation can be one of the following: "add", "subtract", "multiply", or "divide".

```
# Example usage:
# input_list = STT-system("Voice saying; eight plus nine")

input_list = ["eight", "nine", "plus"]
result = perform_operation(input_list)

# TTS-system(result) -> "seventeen"
```

If you stop here, and code this function, by yourself, you will learn a lot. (HARD)

Medium

This is only one way to solve this problem. There are many other ways to solve this problem.

Write a Function that takes a list with three arguments and returns a string as a result. The first argument describes a number as a String (e.g., "eight"), the second argument describes another number as a String (e.g., "nine"), and the third argument describes the operation as a String (e.g., "plus") between the numbers. The result of this operation should be returned as written numbers (e.g., "seventeen").

```
def perform_operation(args):
    # Defining a dictionary to map word representations to numbers

    # Extracting the arguments by unpacking the list

    # Validating the arguments by checking if they are in the dictionary

    # Converting numbers to integers by looking up the dictionary
    # Performing the operation based on the operation argument

    # Converting the result back to a string by looking up the
dictionary

    return result_string # A string representation of the result
```

If you stop here, and code this function, with the help of the instructions above, you still learn a lot. (MEDIUM)

Easy

Fill the TODOs in the code below to complete the function.

```
def perform_operation(args):
    number_map = {
        "zero": 0, "one": 1, "two": 2, "three": 3, "four": 4,
        "five": 5, "six": 6, "seven": 7, "eight": 8, "nine": 9,
        "ten": 10, "eleven": 11, "twelve": 12, "thirteen": 13,
        "fourteen": 14, "fifteen": 15, "sixteen": 16, "seventeen": 17,
        "eighteen": 18, "nineteen": 19, "twenty": 20
    }

    # Extracting the arguments
    #TODO

    # Validating the arguments
    if num1 not in number_map or num2 not in number_map:
        #TODO

    # Converting numbers to integers
    num1 = number_map[num1]
    num2 = number_map[num2]

    # Performing the operation
    if operation == "plus":
        #TODO
    elif operation == "minus":
        #TODO
    elif operation == "multiply":
        #TODO
    elif operation == "divide":
        if num2 == 0:
            #TODO
        #TODO
    else:
        raise ValueError("Invalid operation provided")
    # Converting the result back to a string
    result_string = [key for key, value in number_map.items() if value
== result]
    if len(result_string) == 0:
        raise ValueError("Invalid result")

    return result_string[0]


# Example usage:
input_list = ["eight", "nine", "plus"]
try:
    result = perform_operation(input_list)
    print(result)  # Output: "seventeen"
except (ValueError, ZeroDivisionError) as e:
    print(e)
```

You finished this problem, if you complete the code above, and test it with the example usage. (EASY)

Problem 2 (Prime Number Checker)

Hard

Imagine you are building on a cryptography system, which requires a prime number checker. The system can understand numbers from zero to one hundred The system can also understand the following operations: "is prime".
Write a Function that takes a number as an argument and returns True if the number is a prime number, and False otherwise.

```
# Example usage:
# input_list = Cryp-system("Check if 17 is prime")

input_list = [17, "is prime"]
result = is_prime(input_list)

# Cryp-system(result) -> "True"
```

If you stop here, and code this function, by yourself, you will learn a lot. (HARD)

Medium

This is only one way to solve this problem. There are many other ways to solve this problem.

Write a Function that takes a list with two arguments and returns a boolean as a result. The first argument describes a number as an Integer (e.g., 17), and the second argument describes the operation as a String (e.g., "is prime"). The result of this operation should be returned as a boolean (e.g., True).

```
def is_prime(args):
    # Extracting the arguments by unpacking the list

    # Validating the arguments by checking if the number is in the
correct range

    # Checking if the number is a prime number

    return result_boolean # A boolean representation of the result
```

'Think about what could happen if you don't validate the arguments. What could happen if you don't check if the number is less than 2? What could happen if you don't check if the number is divisible by any number from 2 to the number itself?'

If you stop here, and code this function, with the help of the instructions above, you still learn a lot. (MEDIUM)

Easy

Fill the TODOs in the code below to complete the function.

```
def is_prime(args):
    # Extracting the arguments
    #TODO

    # Validating the arguments
    if num < 0 or num > 100:
        #TODO

    # Checking if the number is a prime number
    # Check if the number is less than 2
    if #TODO:
        return False
    # Check if the number is divisible by any number from 2 to the
number itself
    # If it is, return False
    # Otherwise, return True
    # create a list with all the numbers from 2 to the number itself use
the range function range()
    search_range = range(#TODO)
    # iterate over the search_range and check if the number is divisible
by any number
```

```
    # TODO in search_range:
        # check if the number is divisible by i
        if num % i == 0:
            return False
    return True
```

'Think about what could happen if you don't validate the arguments. What could happen if you don't check if the number is less than 2? What could happen if you don't check if the number is divisible by any number from 2 to the number itself?'

You finished this problem, if you complete the code above, and test it with the example usage. (EASY)