```
# DWAYNE FRASER
# PROBLEM 2

############################## PART A ##########################################
print("PART A")
n = 100
# The indexes start at 0 so the actual register for the value 100 is index 101
print([(a, b, c) for a in range(1, 1 + n) for b in range(1, 1 + n)
            for c in range(1, 1 + n) if c * c == a * a + b * b])

############################## PART B ##########################################
print("\nPART B")
string = []
print('Enter a string: ') # GETS USER INPUT
first_string = input()
string.append(first_string) # APPENDS TO LIST

answer = 'Y'

while (answer == 'Y'): # LOOP
    print('Do you want to input more? (Y or N)') # GETS USER INPUT
    answer = input()

    if (answer == 'Y'):
        print('Enter next string: ') # GETS USER INPUT
        next_string = input()
        string.append(next_string) # APPENDS TO LIST
    if (answer == 'N'):
        break

for i in string: # Traverse string elements
    if len(i) > 3:
        print(len(i), i.upper()) # PRINTS

########################### PART C ##########################################
print("\nPART C")
fullname = [] #LIST DECLARATIONS
nextfullname = []
names_list = []
new_names_list = []

print('Enter first name: ') # GETS USER INPUT
first_name = input()

print('Enter last name: ') # GETS USER INPUT
last_name = input()

print()
fullname = (first_name + " " + last_name)
names_list.append(fullname) # APPENDS TO LIST

fullname = (last_name + " " + first_name)
```

```
new_names_list.append(fullname) # APPENDS TO LIST

print(names_list[0]) # PRINTS
print(new_names_list[0])

answer = 'Y'

while (answer == 'Y'): # LOOP
    print('\nDo you want to input more? (Y or N)') # GETS USER INPUT
    answer = input()

    if (answer == 'Y'):
        print('Enter first name: ') # GETS USER INPUT
        next_first_name = input()
        print('Enter last name: ') # GETS USER INPUT
        next_last_name = input()

        nextfullname = next_first_name + " " + next_last_name
        names_list.append(nextfullname) # APPENDS TO LIST

        nextfullname = next_last_name + " " + next_first_name
        new_names_list.append(nextfullname) # APPENDS TO LIST

    if (answer == 'N'):
        break

print("\nYour old names list is:")
position = 0
while position < len(names_list): # LOOP
    print(names_list[position], end=' ') # PRINTS OLD NAME LIST
    position = position + 1

print("\nYour new names list is:")
position = 0
while position < len(new_names_list): # LOOP
    print(new_names_list[position], end=' ') # PRINTS NEW NAME LIST
    position = position + 1
print()

############################## PART D ##############################################
print("\nPART D")
def concatenate(separator, *arguments):
    """ takes as parameter a string and an arbitrary
    number of additional arguments, all strings, and that returns the
    concatenation of all given strings using the given separator.
    """
    string_sum = []

    if len(arguments) == 1: # IF LENGTH OF ARGUEMENT EQUAL TO 1
        string_sum.append(arguments[0]) # APPEND TO STRING SUM
        return string_sum
    else:
```

```
    position = 0
    while position < len(arguments): # LOOP
        x = arguments[position] + separator # ADDS SEPARATOR
        string_sum.append(x)
        position = position + 1
  return string_sum

print(concatenate(' and ',"Bonny","Clyde"))
print(concatenate(':',"one","two","three"))
print(concatenate('and',"single"))
```

```
In [1]: runfile('C:/Program Files (x86)/Work/Python/Python Dwayne Solutions/HW 2/Homework 2/p2_Fraser_Dwayne.py', wdir='C:/Program Files (x86)/Work/
Python/Python Dwayne Solutions/HW 2/Homework 2')
PART A
[(3, 4, 5), (4, 3, 5), (5, 12, 13), (6, 8, 10), (7, 24, 25), (8, 6, 10), (8, 15, 17), (9, 12, 15), (9, 40, 41), (10, 24, 26), (11, 60, 61), (12, 5,
13), (12, 9, 15), (12, 16, 20), (12, 35, 37), (13, 84, 85), (14, 48, 50), (15, 8, 17), (15, 20, 25), (15, 36, 39), (16, 12, 20), (16, 30, 34), (16, 63,
65), (18, 24, 30), (18, 80, 82), (20, 15, 25), (20, 21, 29), (20, 48, 52), (21, 20, 29), (21, 28, 35), (21, 72, 75), (24, 7, 25), (24, 10, 26), (24,
18, 30), (24, 32, 40), (24, 45, 51), (24, 70, 74), (25, 60, 65), (27, 36, 45), (28, 21, 35), (28, 45, 53), (28, 96, 100), (30, 16, 34), (30, 40, 50),
(30, 72, 78), (32, 24, 40), (32, 60, 68), (33, 44, 55), (33, 56, 65), (35, 12, 37), (35, 84, 91), (36, 15, 39), (36, 27, 45), (36, 48, 60), (36, 77,
85), (39, 52, 65), (39, 80, 89), (40, 9, 41), (40, 30, 50), (40, 42, 58), (40, 75, 85), (42, 40, 58), (42, 56, 70), (44, 33, 55), (45, 24, 51), (45,
28, 53), (45, 60, 75), (48, 14, 50), (48, 20, 52), (48, 36, 60), (48, 55, 73), (48, 64, 80), (51, 68, 85), (52, 39, 65), (54, 72, 90), (55, 48, 73),
(56, 33, 65), (56, 42, 70), (57, 76, 95), (60, 11, 61), (60, 25, 65), (60, 32, 68), (60, 45, 75), (60, 63, 87), (60, 80, 100), (63, 16, 65), (63, 60,
87), (64, 48, 80), (65, 72, 97), (68, 51, 85), (70, 24, 74), (72, 21, 75), (72, 30, 78), (72, 54, 90), (72, 65, 97), (75, 40, 85), (76, 57, 95), (77,
36, 85), (80, 18, 82), (80, 39, 89), (80, 60, 100), (84, 13, 85), (84, 35, 91), (96, 28, 100)]

PART B
Enter a string:

apple
Do you want to input more? (Y or N)

Y
Enter next string:

orange
Do you want to input more? (Y or N)

N
5 APPLE
6 ORANGE

PART C
Enter first name:

cat
Enter last name:

dog

cat dog
dog cat

Do you want to input more? (Y or N)

N

Your old names list is:
cat dog
Your new names list is:
dog cat

PART D
['Bonny and ', 'Clyde and ']
['one:', 'two:', 'three:']
['single']

In [2]:
```