Iteration using While Practice

Use your previous shapes program where you asked the user for the dimensions of the shape you made earlier in the course.

Paste the code that you previously wrote for your shape in the box below   (Just the shape (ie Square), not the whole code for the picture)

```
#Omaima Gohir
#4/3/2023
#this code will return the volume of a square-based pyramid

#this function sets the parameter as base and height and defines the function as the volume of a square-based pyramid
def vol_sqr_pyramid (base, height):
#calculates the total volume using the formula - (base^2 * height)/3
    total_volume = (((base ** 2) * height) / 3)
#returns user the total volume of a square-based pyramid
#the variable, total, contains the mathematical operations for the answer
    return total_volume
```

Write a new program that will:

- Define a function that will return the volume/Area of your shape
- Fully document the function - docstrings at the start of the function
- Define a second function that will ask the user for the dimensions of the shape and how many of these shapes there are. Using loops, determine the area/volume of the shapes.
- **Level 4 programs** will also use a loop to make sure that the dimensions and number of shapes are valid entries. This means that all input numbers should be positive numbers

```
#Omaima Gohir
#5/2/2023
#this code will return the volume of a square-based pyramid
def vol_sqr_pyramid():
    """
    This function takes the users to input the height and base and checks
    if it is a positive integer(checks if its not negative or a letter)

    Preconditions:
    base and height = positive integer

    Parameter:
    The base and height and what is printed

    Returns:
    The volume of square-based Pyramid

    """
```

```
#this part ensures that the user enters a valid number for the base; the base cannot equal a negative number or
string.
    base = input("Enter base:")
#this line continues to run until the base. digit and base < 0, is not true.
    while not  base.isdigit() or float(base) < 0: #.isdigit checks if its a number
        print("Invalid Entry, PLEASE give positive number for the base!")
        base = input("Enter base:")

#this part ensures that the user enters a valid number for height; height cannot equal a negative number or string.
    height = input("Enter height:")
    while not  height.isdigit() or float(height) < 0: #.isdigit checks if its a number
        print("Invalid Entry, PLEASE give positive number for the height!")
        height = input("Enter height:")
    return float(base), float(height)

def calc_vol(base, height):
    """
    This function takes the users to input the height and base and checks
    if it is a positive integer(checks if its not negative or a letter)

    Preconditions:
    base and height = positive integer

    Parameter:
    The base and height and what is printed

    Returns:
    The volume of square-based Pyramid

    """
    return (((base ** 2) * height) / 3)


continue_or_not = "" #defining it to be an empty string
while continue_or_not != "stop": #keeps looping until the user enters stop
    base, height = vol_sqr_pyramid() #calling the first function
    results = calc_vol(base, height) #calling the second function
    print("The volume is:" , results) #printing the results
    continue_or_not = input("say stop, to exit or type any key to continue") #letting the user enter their choice
```

List the test cases that you used to determine whether your program is accurate. Explain the reasoning for each test case on the right.

Test Case	Explanation
base = a base = input("Enter base:") #this line continues to run until the base. digit and base < 0, is not true. while not base.isdigit() or float(base) < 0: #.isdigit checks if its a number print("Invalid Entry, PLEASE give positive number for the base!") base = input("Enter base:")	When this part of the code is a number that is smaller than 0 or is equal to a string, such as a letter. The code will always print "Invalid Entry, PLEASE give a positive number for the base!") UNTIL the number is greater than 0. This is because the while not function keeps repeating the function until the number is greater than 0, which is the truth of while not.

height = -1
height = input("Enter height:")
while not height.isdigit() or float(height) < 0: #.isdigit checks if its a number
print("Invalid Entry, PLEASE give positive number for the height!")
height = input("Enter height:")
return float(base), float(height)
	When this part of the code is a number that is smaller than 0 or is equal to a string, such as a letter. The code will always print "Invalid Entry, PLEASE give a positive number for the base!") UNTIL the number is greater than 0. This is because the while not function keeps repeating the function until the number is greater than 0, which is the truth of while not.

base = 3 base = input("Enter base:") #this line continues to run until the base. digit and base < 0, is not true. while not base.isdigit() or float(base) < 0: #.isdigit checks if its a number print("Invalid Entry, PLEASE give positive number for the base!") base = input("Enter base:")	When this part of the code's statement is true, meaning that a number is a positive number, because it is greater than 0, and not a string, the code will work and continue to the next lines of code, leading up to the volume of the shape.

height = 5
height = input("Enter height:")
while not height.isdigit() or float(height) < 0: #.isdigit checks if its a number
print("Invalid Entry, PLEASE give positive number for the height!")
height = input("Enter height:")
return float(base), float(height)
	When this part of the code's statement is true, meaning that a number is a positive number, because it is greater than 0, and not a string, the code will work and continue to the next lines of code, leading up to the volume of the shape.