Functions, Loops, Scope, Tuples, and Sets

```
# 1: Using the return Statement
def cube(num):
    return num ** 3

print("Task 1 Output:", cube(3))  # Output: 27



# 2: Using break in a Loop
while True:
    num = int(input("Task 2 - Enter a number: "))
    if num < 0:
        print("Negative number entered. Exiting.")
        break
    print("Cube:", num ** 3)

# 3: Using continue in a Loop
print("Task 3 Output:")
for i in range(1, 21):
    if i % 2 != 0:
        continue
    print(i)



# 4: Using break and continue in a Loop
total = 0
while True:
    num = int(input("Task 4 - Enter a number (0 to stop): "))
    if num < 0:
        continue
    if num == 0:
        break
    total += num

print("Task 4 Output - Sum of positive numbers:", total)



# 5: Local Scope in Functions
def square():
    local_var = 5
    result = local_var ** 2
    print("Task 5 Output - Square is:", result)

square()
# print(local_var)  # Uncommenting this will cause an error: NameError



# 6: Modifying a Global Variable Inside a Function
counter = 0

def increment():
    global counter
    counter += 1
    print("Task 6 Output - Counter:", counter)

increment()
increment()
increment()
```

```
# 7: Demonstrating Local and Global Scope
x = 10


def modify_global():
    global x
    x = 20
    print("Task 7 Output - Inside function, global x:", x)


def local_scope():
    y = 30
    print("Task 7 Output - Inside function, local y:", y)


modify_global()
print("Task 7 Output - Outside function, global x:", x)


local_scope()
# print(y)  # Uncommenting this will cause an error: NameError



# 8: Working with Tuples in Python
fruits = ("apple", "banana", "cherry")
print("Task 8 Output - Second element:", fruits[1])


# fruits[1] = "orange"  # This will raise a TypeError


fruits_list = list(fruits)
fruits_list[1] = "orange"
fruits = tuple(fruits_list)
print("Task 8 Output - Modified tuple:", fruits)
```

```
# 9: Working with Sets in Python
set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}

print("Task 9 Output - Union:", set1 | set2)
print("Task 9 Output - Intersection:", set1 & set2)
print("Task 9 Output - Difference:", set1 - set2)

set1.add(6)
set2.remove(4)

print("Task 9 Output - Updated set1:", set1)
print("Task 9 Output - Updated set2:", set2)

print("Task 9 Output - Is 3 in set1?", 3 in set1)



# 10: Tuple and Set Operations
my_tuple = (10, 20, 30, 40, 50)
print("Task 10 Output - Third element:", my_tuple[2])

my_set = set(my_tuple)
my_set.add(60)
print("Task 10 Output - Updated set:", my_set)



# 11: List to Set Conversion & Basic Set Operations
num_list = [1, 2, 2, 3, 4, 4, 5]
unique_set = set(num_list)
print("Task 11 Output - Unique elements:", unique_set)
```

```
another_set = {3, 4, 5, 6, 7}

print("Task 11 Output - Intersection:", unique_set & another_set)
print("Task 11 Output - Union:", unique_set | another_set)



#12: Working with Tuples, Sets, and Lists
num_list = [10, 20, 30, 40, 50]
num_list.append(60)

num_tuple = tuple(num_list)
print("Task 12 Output - Tuple:", num_tuple)

num_set = set(num_list)
print("Task 12 Output - Unique values (set):", num_set)
```