```python
# Lab 7 solutions
#1
def my_len(iterable):
    '''returns the length of 'iterable'.'''
    result = 0
    for element in iterable:
        result += 1
    return result
#2
def my_product(my_list):
    '''returns the product of all elements in a list of numbers.'''
    if my_list == []:
        return 0
    result = 1
    for v in my_list:
        result *= v
    return result

#3
def my_count(my_list, value):
    '''returns the number of times 'value' appears as an element in my_list.'''
    result = 0
    for v in my_list:
        if v == value:
            result += 1
    return result

#4
def my_find(my_list, value):
    '''returns the index of the first element of 'iterable' that matches
    'value'. Returns -1 if value is not found.'''
    for i in range(len(my_list)):
        if my_list[i] == value:
            return i
    return -1

#Q5
def sum_cubes_r(n): #recursive function
    """ For a given n, returns the sum_cubes n**3 + (n-1)**3+ ... + 1"""
    if n==0:
        return n
    else:
        return n**3 + sum_cubes_r(n-1)

def my_find_r(my_list, value):
    '''returns the index of the first element of 'iterable' that matches
    'value'. Returns -1 if value is not found.'''
    if value not in my_list:
        return -1
    if my_list[0] == value:
        return 0
    return 1 + my_find_r(my_list[1:],value)

#1
#len function
# Test cases for Function my_len
print("Q1")
print(my_len([]))
print(my_len("Marianopolis College"))
print(my_len((1,5,9,2,1)))

#2
# Test cases for Function my_product
print("\nQ2")
print(my_product([1, 5, 9]))
print(my_product([10, 20, 30.0]))
print(my_product([])) #prints 0
#3
#count function
# Test cases for Function my_count
x = [1, 2, 3, 3]
print("\nQ3")
print(my_count(x, 1))
print(my_count(x, 3))
print(my_count(x, 4))
#4
#find function
# Test cases for Function my_find
print("\nQ4")
print(my_find([1, 5, 9, 1], 0)) #returns -1
print(my_find([1, 5, 9, 1], 9)) #returns 2
print(my_find([1, 5, 9, 1], 1)) #returns 0
#5
#sum of cubes function
# Test cases for Function sum_cubes_r
print("\nQ5")
print(sum_cubes_r(4)) #returns 100
print(sum_cubes_r(5)) #returns 225

#6
#find function recursive
# Test cases for Function my_find
print("\nQ6")
print(my_find_r([1, 5, 9, 1], 0)) #returns -1
print(my_find_r([1, 5, 9, 1], 9)) #returns 2
print(my_find_r([1, 5, 9, 1], 1)) #returns 0
```