Exercises

1. Declare a variable my_list and assign it a list of integers. Write a function that returns the second largest number in the list.
2. Declare two variables list1 and list2, each containing a list of integers. Write a function that merges the two lists and returns a new list with all the elements sorted in ascending order.
3. Declare a variable my_list and assign it a list of strings. Write a function that takes a string as input and returns a new list containing all the strings from the original list that contain the input string.
4. Declare a variable my_list and assign it a list of integers. Write a function that takes an integer as input and returns a new list containing all the numbers from the original list that are divisible by the input integer.
5. Declare a variable my_list and assign it a list of strings. Write a function that takes a string as input and returns a new list containing all the strings from the original list that start with the input string.
6. Declare a variable my_list and assign it a list of integers. Write a function that returns a new list with the same numbers as the original list, but with all duplicates removed.
7. Declare a variable my_list and assign it a list of strings. Write a function that takes a list of strings as input and returns a new list containing only the unique strings from the original list, in the order in which they first appeared.
8. Declare a variable my_list and assign it a list of integers. Write a function that takes an integer as input and returns a new list with all the elements of the original list shifted to the left by the input integer. For example, if the input integer is 2, the first two elements of the original list should be moved to the end of the list.
9. Declare a variable my_list and assign it a list of integers. Write a function that takes an integer as input and returns a new list with all the elements of the original list shifted to the right by the input integer. For example, if the input integer is 2, the last two elements of the original list should be moved to the beginning of the list.
10. Declare a variable my_list and assign it a list of integers. Write a function that returns a new list with the same elements as the original list, but sorted in descending order.

Exercises and Solution

1. Declare a variable my_list and assign it a list of integers. Write a function that returns the second largest number in the list.

```
def second_largest(my_list):

  new_list = sorted(my_list, reverse=True)

  return new_list[1]
```

```
my_list = [3, 5, 2, 7, 1, 9, 8]
print(second_largest(my_list))
```

2. Declare two variables list1 and list2, each containing a list of integers. Write a function that merges the two lists and returns a new list with all the elements sorted in ascending order.

```
def merge_sort(list1, list2):
  new_list = list1 + list2
  return sorted(new_list)
list1 = [3, 5, 2, 7, 1]
list2 = [9, 8, 4, 6]
print(merge_sort(list1, list2))
```

3. Declare a variable my_list and assign it a list of strings. Write a function that takes a string as input and returns a new list containing all the strings from the original list that contain the input string.

```
def string_search(my_list, input_string):
  new_list = []
  for string in my_list:
    if input_string in string:
      new_list.append(string)
  return new_list
my_list = ['hello', 'world', 'goodbye', 'python', 'list']
print(string_search(my_list, 'o'))
```

4. Declare a variable my_list and assign it a list of integers. Write a function that takes an integer as input and returns a new list containing all the numbers from the original list that are divisible by the input integer.

```
def divisible_numbers(my_list, input_num):
  new_list = []
  for num in my_list:
    if num % input_num == 0:
      new_list.append(num)
  return new_list
my_list = [3, 5, 2, 7, 1, 9, 8]
print(divisible_numbers(my_list, 3))
```

5. Declare a variable my_list and assign it a list of strings. Write a function that takes a string as input and returns a new list containing all the strings from the original list that start with the input string.

```
def string_start(my_list, input_string):
  new_list = []
  for string in my_list:
    if string.startswith(input_string):
      new_list.append(string)
  return new_list
my_list = ['hello', 'world', 'goodbye', 'python', 'list']
print(string_start(my_list, 'h'))
```

6. Declare a variable my_list and assign it a list of integers. Write a function that returns a new list with the same numbers as the original list, but with all duplicates removed.

```
def remove_duplicates(my_list):
  return list(set(my_list))
my_list = [3, 5, 2, 7, 1, 9, 8, 2, 5]
print(remove_duplicates(my_list))
```

7. Declare a variable my_list and assign it a list of strings. Write a function that takes a list of strings as input and returns a new list containing only the unique strings from the original list, in the order in which they first appeared.

```
def unique_strings(my_list):
  new_list = []
  for string in my_list:
    if string not in new_list:
      new_list.append(string)
  return new_list
my_list = ['hello', 'world', 'goodbye', 'python', 'list', 'hello']
print(unique_strings(my_list))
```

8. Declare a variable my_list and assign it a list of integers. Write a function that takes an integer as input and returns a new list with all the elements of the original list shifted to the left by the input integer. For example, if the input integer is 2, the first two elements of the original list should be moved to the end of the list.

```
def shift_left(my_list, input_num):
  return my_list[input_num:] + my_list[:input_num]
```

```
my_list = [3, 5, 2, 7, 1, 9, 8]

print(shift_left(my_list, 2))
```

9. Declare a variable my_list and assign it a list of integers. Write a function that takes an integer as input and returns a new list with all the elements of the original list shifted to the right by the input integer. For example, if the input integer is 2, the last two elements of the original list should be moved to the beginning of the list.

```
def shift_right(my_list, input_num):

  return my_list[-input_num:] + my_list[:-input_num]

my_list = [3, 5, 2, 7, 1, 9, 8]

print(shift_right(my_list, 2))
```

10. Declare a variable my_list and assign it a list of integers. Write a function that returns a new list with the same elements as the original list, but sorted in descending order.

```
my_list = [3, 5, 2, 7, 1, 9, 8]

new_list = sorted(my_list, reverse=True)

print(new_list)
```