Question 1

Solution:
```
def selection_sort(lst):
    n = len(lst)
    for i in range(n-1):
        min_index = i
        for j in range(i+1, n):
            if lst[j] < lst[min_index]:
                min_index = j
        lst[i], lst[min_index] = lst[min_index], lst[i]
    return lst

# Example usage
input_list = [5, 416, 54, 21, 6135, 15, 741]
sorted_list = selection_sort(input_list)
print(sorted_list)
```

Question 2:

```
def get_file_types(extension_type_list, filenames):
    extension_dict = {}
    for item in extension_type_list:
        extension, file_type = item.split(',')
        extension_dict[extension] = file_type

    result_dict = {}
    for filename in filenames:
        extension = filename.split('.')[-1]
        file_type = extension_dict.get(extension, 'unknown')
        result_dict[filename] = file_type

    return result_dict
```

```
# Example usage
extension_type_string = "xls,spreadsheet;xlsx,spreadsheet;jpg,image"
filenames = ["abc.jpg", "xyz.xls", "text.csv", "123"]
result = get_file_types(extension_type_string.split(';'), filenames)
print(result)
output: {
  "abc.jpg": "image",
  "xyz.xls": "spreadsheet",
  "text.csv": "unknown",
  "123": "unknown"
}
```

Question 3:

```
def sort_list_of_dicts(lst, key):
    return sorted(lst, key=lambda x: x[key])

# Example usage
input_list = [
    {"fruit": "orange", "color": "orange"},
    {"fruit": "apple", "color": "red"},
    {"fruit": "banana", "color": "yellow"},
    {"fruit": "blueberry", "color": "blue"}
]

sorted_list_by_fruit = sort_list_of_dicts(input_list, "fruit")
print(sorted_list_by_fruit)

sorted_list_by_color = sort_list_of_dicts(input_list, "color")
```

```
print(sorted_list_by_color)
```

Output: [{"fruit": "apple", "color": "red"}, {"fruit": "banana", "color": "yellow"}, {"fruit": "blueberry", "color": "blue"}, {"fruit": "orange", "color": "orange"}]

[{"fruit": "blueberry", "color": "blue"}, {"fruit": "orange", "color": "orange"}, {"fruit": "apple", "color": "red"}, {"fruit": "banana", "color": "yellow"}]

Question 4:

```
def switch_key_value(dictionary):
    return {value: key for key, value in dictionary.items()}


# Example usage
input_dict = {
    "key1": "value1",
    "key2": "value2",
    "key3": "value3",
    "key4": "value4",
    "key5": "value5"
}


result_dict = switch_key_value(input_dict)
print(result_dict)
```

```
Output: {
  "value1": "key1",
  "value2": "key2",
  "value3": "key3",
  "value4":
```

Question 5:

```
def find_common_and_not_common(list1, list2):
    common_elements = list(set(list1) & set(list2))
    not_common_elements = list(set(list1) ^ set(list2))
    return common_elements, not_common_elements


# Example usage
mainstream = ["One Punch Man", "Attack On Titan", "One Piece", "Sword Art Online", "Bleach", "Dragon Ball Z", "One Piece"]
must_watch = ["Full Metal Alchemist", "Code Geass", "Death Note", "Stein's Gate", "The Devil is a Part Timer!", "One Piece", "Attack On Titan"]
common, not_common = find_common_and_not_common(mainstream, must_watch)
print(common)
print(not_common)
```

```
Output: ["One Piece", "Attack On Titan"]
["Dragon Ball Z", "One Punch Man", "Stein's Gate", "Sword Art Online", "Full Metal Alchemist", "Code Geass", "The Devil is a Part Timer!", "Bleach", "Death Note"]
```

Question 6:

```
def extract_sublist(lst, start_index, end_index):
    return lst[start_index:end_index:2]


# Example usage
input_list = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41]
sublist = extract_sublist(input_list, 2, 9)
print(sublist)
```

```
Output: [5, 11, 17, 23]
```

Question 7:

```
factorial = lambda n: 1 if n == 0 else n * factorial(n - 1)


# Example usage
num = 5
```

```
result = factorial(num)

print(result)

Output: 120
```

Question 8:

```
from functools import reduce


A0 = dict(zip(('a', 'b', 'c', 'd', 'e'), (1, 2, 3, 4, 5)))

A1 = range(10)

A2 = sorted([i for i in A1 if i in A0])

A3 = sorted([A0[s] for s in A0])

A4 = [i for i in A1 if i in A3]

A5 = {i: i * i for i in A1}

A6 = [[i, i * i] for i in A1]

A7 = reduce(lambda x, y: x + y, [10, 23, -45, 33])

A8 = list(map(lambda x: x * 2, [1, 2, 3, 4]))

A9 = list(filter(lambda x: len(x) > 3, ["I", "want", "to", "learn", "python"]))


print(A0)

print(list(A1))

print(A2)

print(A3)

print(A4)

print(A5)

print(A6)

print(A7)

print(A8)

print(A9)

Output: {'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[]
[1, 2, 3, 4, 5]
[1, 2, 3,
```

Question 9: from datetime import datetime, timedelta

```
def is_date_within_range(from_date, to_date, difference):
    format_str = '%y-%m-%d'
    date1 = datetime.strptime(from_date, format_str)
    date2 = datetime.strptime(to_date, format_str)
    delta = date2 - date1
    if abs(delta.days) < difference:
        return True
    else:
        return False

# Example usage
from_date = '21-05-01'
to_date = '21-05-10'
difference = 10
result = is_date_within_range(from_date, to_date, difference)
print(result)
```

Output: True

Question 10:

```
from datetime import datetime, timedelta

def get_date_n_days_before(date, n):
    format_str = '%y-%m-%d'
    given_date = datetime.strptime(date, format_str)
    new_date = given_date - timedelta(days=n)
    return new_date.strftime(format_str)

# Example usage
date = '16-12-10'
n = 11
result = get_date_n_days_before(date, n)
print(result)
Output: 16-11-29
0
```

Question 11:

```
def f(x, l=[]):
    for i in range(x):
        l.append(i*i)
    print(l)

f(2)
f(3, [3, 2, 1])
f(3)
Output: [0, 1]
[3, 2, 1, 0, 1, 4]
[0, 1, 0, 1, 4]
```