

# CSE 160 Section 3 - SOLUTIONS

## Question 1

SOLUTION:

```
List 1: [98, 2, 3, 4, 99]
List 2: [98, 2, 3, 4, 99]
List 3: [1, 2, 3, 4, 5]
```

## Question 2

ONE POSSIBLE SOLUTION:

```
list_6.reverse()
list_1.extend(list_6)
list_2.insert(1, list_7[0])
list_1.extend(list_2)
list_1.extend(list_4)
list_1.extend(list_5)

for item in list_3:
    list_1.insert(-2, item)
```

## Question 3

ONE POSSIBLE SOLUTION:

```
list_1.extend(list_2)
list_1.extend(list_3)
list_1.extend(list_4)
list_1.extend(list_5)
list_1.extend(list_6)
list_1.extend(list_7)

list_1.sort()
print list_1
```

## Question 4

### POSSIBLE SOLUTIONS:

1.

```
def find_index(my_list, my_value):
    for index in range(len(my_list)):
        if my_list[index] == my_value:
            return index
    return -1
```

2.

```
def friend_list_index(friends_list, name):
    for index in range(len(friends_list)):
        if friends_list[index][0] == name:
            return index
    return -1
```

3.

```
def really_my_friends(friends_list):
    numFriends = 0
    my_list = friends_list[0]
    others = friends_list[1:]
    for friends in others:
        if "me" in friends:
            numFriends = numFriends + 1
    return numFriends
```

4.

```
def really_friends(friends_list, name):
    numFriends = 0
    index = friend_list_index(friends_list, name)
    name_list = friends_list[index]
    others = friends_list[0:len(friends_list)]
    others.remove(name_list)
    for friends in others:
        if name in friends:
            numFriends = numFriends + 1
    return numFriends
```

5.

```
def all_mutual_friends(friends_list):
    for person_list in friends_list:
        if really_friends(friends_list, person_list[0]) ==
(len(person_list) - 1):
            return True
    return False
```