Algorithms & Data Structures

Lab 7 (Operations on Lists)

EX1:

```python
# A slicing expression selects a range
# of elements from a sequence.
days = ['Sunday', 'Monday', 'Tuesday', 'Wednesday',
        'Thursday', 'Friday', 'Saturday']
print(days[2:5])
numbers = [1, 2, 3, 4, 5]
print(numbers[1:3])
print(numbers[:3])
print(numbers[2:])
print(numbers[:])
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(numbers[1:8:2])
print(numbers[-5:])
```

EX2:

```python
# You can search for an item in
# a list using the in operator.
# This program demonstrates the in operator
# used with a list.
def main():
    # Create a list of product numbers.
    prod_nums = ['V475', 'F987', 'Q143', 'R688']
    # Get a product number to search for.
    search = input('Enter a product number: ')
    # Determine whether the product number is in the
list.
    if search in prod_nums:
        print(search, 'was found in the list.')
    else:
        print(search, 'was not found in the list.')
# Call the main function.
main()
```

EX3:

```
# This program demonstrates how the append
# method can be used to add items to a list.
def main():
    # First, create an empty list.
    name_list = []
    # Create a variable to control the loop.
    again = 'y'
    # Add some names to the list.
    while again == 'y':
        # Get a name from the user.
        name = input('Enter a name: ')
        # Append the name to the list.
        name_list.append(name)
        # Add another one?
        print('Do you want to add another name?')
        again = input('y = yes, anything else = no: ')
        print()
    # Display the names that were entered.
    print('Here are the names you entered.')
    for name in name_list:
        print(name)
# Call the main function.
main()
```

EX4:

```
# This program demonstrates how to get the
# index of an item in a list and then replace
# that item with a new item.
def main():
    # Create a list with some items.
    food = ['Pizza', 'Burgers', 'Chips']
    # Display the list.
    print('Here are the items in the food list:')
    print(food)
    # Get the item to change.
    item = input('Which item should I change? ')
    try:
        # Get the item's index in the list.
        item_index = food.index(item)
        # Get the value to replace it with.
        new_item = input('Enter the new value: ')
        # Replace the old item with the new item.
        food[item_index] = new_item
        # Display the list.
        print('Here is the revised list:')
        print(food)
    except ValueError:
        print('That item was not found in the list.')
# Call the main function.
main()
```

EX5:

```
# This program demonstrates the insert method.
def main():
    # Create a list with some names.
    names = ['James', 'Kathryn', 'Bill']
    # Display the list.
    print('The list before the insert:')
    print(names)
    # Insert a new name at element 0.
    names.insert(0, 'Joe')
    # Display the list again.
    print('The list after the insert:')
    print(names)
# Call the main function.
main()
```

EX6:

```
# The sort method rearranges the elements
# of a list so they appear in ascending order
my_list = [9, 1, 0, 2, 8, 6, 7, 4, 5, 3]
print('Original order:', my_list)
my_list.sort()
print('Sorted order:', my_list)
my_list = ['beta', 'alpha', 'delta', 'gamma']
print('Original order:', my_list)
my_list.sort()
print('Sorted order:', my_list)
```

EX7:

```python
# This program demonstrates how to use the remove
# method to remove an item from a list.
def main():
    # Create a list with some items.
    food = ['Pizza', 'Burgers', 'Chips']
    # Display the list.
    print('Here are the items in the food list:')
    print(food)
    # Get the item to change.
    item = input('Which item should I remove? ')
    try:
        # Remove the item.
        food.remove(item)
        # Display the list.
        print('Here is the revised list:')
        print(food)
    except ValueError:
        print('That item was not found in the list.')
# Call the main function.
main()
```

EX8:

```python
# The reverse method simply reverses the
# order of the items in the list.
my_list = [1, 2, 3, 4, 5]
print('Original order:', my_list)
my_list.reverse()
print('Reversed:', my_list)
```

EX9:

```python
# the del statement removes an element
# from a specific index, regardless of
# the item that is stored at that index.
my_list = [1, 2, 3, 4, 5]
print('Before deletion:', my_list)
del my_list[2]
print('After deletion:', my_list)
```

EX10:

```python
# Python has two built-in functions named
# min and max that work with sequences.
my_list = [5, 4, 3, 2, 50, 40, 30]
print('The lowest value is', min(my_list))
print('The highest value is', max(my_list))
```