# Homework 2 Lists

1. A list of client records is stored in a dynamic **list** data structure **clients**.

The list contains the following elements:

clients = [‘Gries’, ‘Schneider’, ‘Bergin’, ‘Kozen’, ‘Pearce’]

(a) Show the state of the list after the execution of the following operations. [2]

pop(), remove(‘Bergin’), append(‘Kizza’), pop(1), append(‘Brooks’), insert(2,’Merritt’)

(b) The list is maintained in alphabetical order. Show the state of the list after the execution of the given operations. [2]

Original state of list: clients = [‘Bergin’, ‘Gries’, ‘Kozen’, ‘Pearce’, ‘Schneider’]

Operations: pop(), remove(‘Bergin’), add(‘Kizza’), x 🡨 index(Pearce), pop(x), add(‘Brooks’), add (‘Merritt’)

2. A list is maintained in alphabetical order. Write the pseudocode to show how a new item is inserted into the list at the correct location. The maximum size of the list is not known. [5]

3. Here is a list of available operations.

|  |  |
| --- | --- |
| **List operation** | **Description** |
| isEmpty() | Test for empty list |
| append(item) | Add a new item to list |
| remove(item) | Remove the first occurrence of an item from list |
| search(item) | Search for an item in list |
| length() | Return the number of items |
| index(item) | Return the position of item |
| insert(pos,item) | Add a new item |
| pop() | Remove and return the last item in the list |
| pop(pos) | Remove and return the item at position pos |

(a) Using only the operations from this list, describe how to delete an item, without using *remove(item).* [2]

(b) Using only the operations from this list, describe how you could add an item to the end of the list without using *append(item)* . [2]

(c) Using only the operations from this list, describe how to implement the queue methods *enQueue(item)* and *item 🡨 deQueue().* Do not worry about full or empty. [2]

[Total 15 Marks]