

**(100 marks)** Lists are members of a general category of abstract data types (ADTs) called *containers* (i.e., objects whose purpose is to hold other objects). A *list* is a collection of items having the following defining characteristics:

- a. *Homogeneity*: All the items are of the same type.
- b. *Linearity*: Each item has a unique predecessor (except the first) and a unique successor (except the last).
- c. *Variable Length*: The number of items can vary over time.
- d. *Order*: Items may be *ordered* (i.e., as in a *sorted* list) or *unordered* (i.e., as in an *unsorted* list).
- e. *Keys*: A member or combination of members describing an item may be *unique* (i.e., there is only one item matching the description) or *duplicated* (i.e., more than one item matches the description).

## PART 1

Implement a C++ class for an array representation of a list. The methods must be based upon the pseudocode provided on the CS210 Algorithms page. Use the following declarations as a starting point for your implementation.

```
const int MAX_LENGTH = some application-specific max value;
```

```
typedef <some data type> DataType;
```

```
class List
{
public:
    methods go here
private:
    int p;
    int length;
    DataType a [MAX_LENGTH];
};
```

Note: Any other variable or constant declarations required would be local to the methods.

## PART 2

Implement a C++ class for a linked list representation of a list. The methods must be based upon the pseudocode provided on the CS210 Algorithms page. Use the following declarations as a starting point for your implementation.

```

typedef <some data type> DataType;

struct Node;

struct Node
{
    DataType value;
    Node* next;
};

class List
{
public:
    methods go here
private:
    Node* head;
    int length;
    Node* p;
    Node* prevP;
};

```

Note: Any other variable or constant declarations required would be local to the methods.

### **PART 3**

Implement a C++ program that can be used to demonstrate that the list classes implemented in Parts 1 and 2 are correct. To simplify the demonstration, set *some application-specific max value* to 10 and *some data type* to `int`.

### **PART 4**

For programming problems, the Results are worth 70%. So, for this problem, the Results are worth 70 marks out of the 100 marks available. Demonstrate that the list classes are correct. Be sure to demonstrate all possible execution sequences, including boundary conditions and error conditions as discussed in class. Your demonstration should be captured in a script file.