|  |  |
| --- | --- |
| Data Structures & Algorithms Diploma in IT, CSF  Year 2 (2024/25) Semester 3 | Week 3 |
| 2-3 Hours |
| **Practical 3 – Linked List** | |

1. Analyze the specification of a List ADT below.

|  |
| --- |
| *// List.h - - Specification of List ADT (implemented using Pointers)*  #pragma once  #include<string>  #include<iostream>  using namespace std;  typedef string ItemType;  class List  {  private:  struct Node  {  ItemType item; // item  Node \*next; // pointer pointing to next item  };  Node \*firstNode; // point to the first item  int size; // number of items in the list  public:  // constructor  List();  // destructor  ~List();  *// add an item to the back of the list (append)*  bool add(ItemType item);  *// add an item at a specified position in the list (insert)*  bool add(int index, ItemType item);  *// remove an item at a specified position in the list*  void remove(int index);  *// get an item at a specified position of the list (retrieve)*  ItemType get(int index);  *// check if the list is empty*  bool isEmpty();  *// check the size of the list*  int getLength();  *// display all the items in the list*  void print();  }; |

Implement the operations of the List ADT

*Note: You should implement (and test) one operation at a time.*

2. Write a sample program, ListApp.cpp, to do the following:

1. Create an empty list, **nameList**.
2. Add the name, Annie, to the list
3. Add the name, Jacky, to the list
4. Add the name, Wendy, to the list
5. Display all the names in the list
6. Add the name, Brenda, to the second position in the list
7. Display the all the names in the list
8. Remove the name in the third position of the list
9. Display the all the names in the list
10. Remove the name in the first position of the list
11. Display all the names in the list