Problem 2 Test Plan

Part a: Y

Part b: Y

Part c: Y

Part d: Y

The expected result and the actual result match up.

|  |  |  |
| --- | --- | --- |
| Test input | Expected result | Actual result |
| cout << "\n" << "Assingment 1 Problem 2:" << endl;  cout << "Note: every iteration through the list uses equals() in the loop, therefore it is being tested on every output" << endl;  List problem2list;  problem2list.frontInsert(2);  problem2list.frontInsert(1);  problem2list.rearInsert(3);  problem2list.rearInsert(4);  problem2list.rearInsert(5);  problem2list.rearInsert(6);  //test part c  //declare iterators  Iterator position\_begin;  Iterator position\_end;  //Test List::begin()  //iterate through the list with next  position\_begin = problem2list.begin();  position\_end = problem2list.end();  cout << '\n' << "Testing begin() and next() to iterate through the list " << endl;  while (!position\_begin.equals(position\_end)) {  cout << "postition\_begin = " << position\_begin.get() << endl;  position\_begin.next();  }    //Test List::end()  //Iterate through the list with previous()  position\_begin = problem2list.begin();  position\_end = problem2list.end();  cout << '\n' << "Testing end() and previous() to iterate through the list " << endl;  while (!position\_end.equals(position\_begin)) {  cout << "position\_end = " << position\_end.get() << endl;  position\_end.previous();  }    //Test of the insert method for as node in the middle of the list  position\_begin = problem2list.begin();  for (int i = 0; i < 3; i++) {  position\_begin.next();  }  problem2list.insert(position\_begin, 99);  //Test inserting at the begining and the end  position\_begin = problem2list.begin();  problem2list.insert(position\_begin, 99);  position\_end = problem2list.end();  problem2list.insert(position\_end, 99);  position\_begin = problem2list.begin();  //print the linked list  cout << '\n' << "Testing the insert() on the begining, middle and end of the linked list: " << endl;  while (!position\_begin.equals(position\_end)) {  cout << "postition\_begin = " << position\_begin.get() << endl;  position\_begin.next();  }    //Test delete\_node()  //Test deletitng at the beginging and the end of the list  position\_begin = problem2list.begin();  //problem2list.insert(position\_begin, 99);  position\_end = problem2list.end();  // problem2list.insert(position\_end, 99);  problem2list.delete\_node(position\_begin); //delete the first node in the linked list  problem2list.delete\_node(position\_end); // delete the last node in the linked list  //delet the node that was added to the middle of the list  position\_begin = problem2list.begin();  for (int i = 0; i < 3; i++) {  position\_begin.next();  }  //print the linked list  position\_begin = problem2list.begin();  cout << '\n' << "Test the delete\_node function by deleteing the nodes that were inserted: " << endl;  while (!position\_begin.equals(position\_end)) {  cout << "postition\_begin = " << position\_begin.get() << endl;  position\_begin.next();  } | Assingment 1 Problem 2:  Note: every iteration through the list uses equals() in the loop, therefore it is being tested on every output  Testing begin() and next() to iterate through the list  postition\_begin = 1  postition\_begin = 2  postition\_begin = 3  postition\_begin = 4  postition\_begin = 5  postition\_begin = 6  Testing end() and previous() to iterate through the list  End of the List  position\_end = 6  position\_end = 5  position\_end = 4  position\_end = 3  position\_end = 2  Testing the insert() on the begining, middle and end of the linked list:  postition\_begin = 99  postition\_begin = 1  postition\_begin = 2  postition\_begin = 3  postition\_begin = 99  postition\_begin = 4  postition\_begin = 5  postition\_begin = 6  postition\_begin = 99  Test the delete\_node function by deleteing the nodes that were inserted:  postition\_begin = 1  postition\_begin = 2  postition\_begin = 3  postition\_begin = 99  postition\_begin = 4  postition\_begin = 5  postition\_begin = 6 | Assingment 1 Problem 2:  Note: every iteration through the list uses equals() in the loop, therefore it is being tested on every output  Testing begin() and next() to iterate through the list  postition\_begin = 1  postition\_begin = 2  postition\_begin = 3  postition\_begin = 4  postition\_begin = 5  postition\_begin = 6  Testing end() and previous() to iterate through the list  End of the List  position\_end = 6  position\_end = 5  position\_end = 4  position\_end = 3  position\_end = 2  Testing the insert() on the begining, middle and end of the linked list:  postition\_begin = 99  postition\_begin = 1  postition\_begin = 2  postition\_begin = 3  postition\_begin = 99  postition\_begin = 4  postition\_begin = 5  postition\_begin = 6  postition\_begin = 99  Test the delete\_node function by deleteing the nodes that were inserted:  postition\_begin = 1  postition\_begin = 2  postition\_begin = 3  postition\_begin = 99  postition\_begin = 4  postition\_begin = 5  postition\_begin = 6 |