**Longest Increasing Subsequence.**

**Developer**: Diemi Pham

**Date**: Dec 2020

DESCRIPTION

**Project Objective:**

As a developer, write a program to find the longest increasing subsequence.

**Background of the problem statement:**

As a developer, write a Java code to find the longest increasing subsequence from a list of random numbers.

**Implementation:**

* The program is run from the main class (LisMain).
* This program will take an input from user (method setLength()) to define size of the sequence.
* The sequence will be generated from random integers in range of [0,100] with the size defined by user (method generateSequence())
* In this program, the developer implemented the dynamic programming approach with the complexity of O(n^2) (class LisDynamicProgramming. The program finds the longest increasing sub-sequence from the generated list and prints out the result sub-sequence to the screen.
* For comparison purpose, the developer also implemented the recursive approach with the complexity of O(nlogn) (class LisRecursive. This implementation finds the longest increasing sub-sequence from the generated list and will not print out the result sub-sequence to the screen. In some cases, the recursive solution may take longer runtime than the dynamic programming solution (i.e. when the list size is 50).

**Screenshots:**

