**Student Name： studentID:**

**Experiment 1:** List, Stack and Queue

**Purpose:**

1. Master the methods of implementing List, Stack and Queue ADTs;

2. Learn to use List, Stack and Queue ADTs and their implementations to solve problems;

3. Understand the way of selecting appropriate ADTs for solving given problems.

**Problems and requirements:**

**【Problem 1】: Implement the linked list and array list.**

**Requirements:**

(1) The linklist.h is provided for you, and you should implement all of its functions and put them into .c files. (define and implement auxiliary operations when needed)

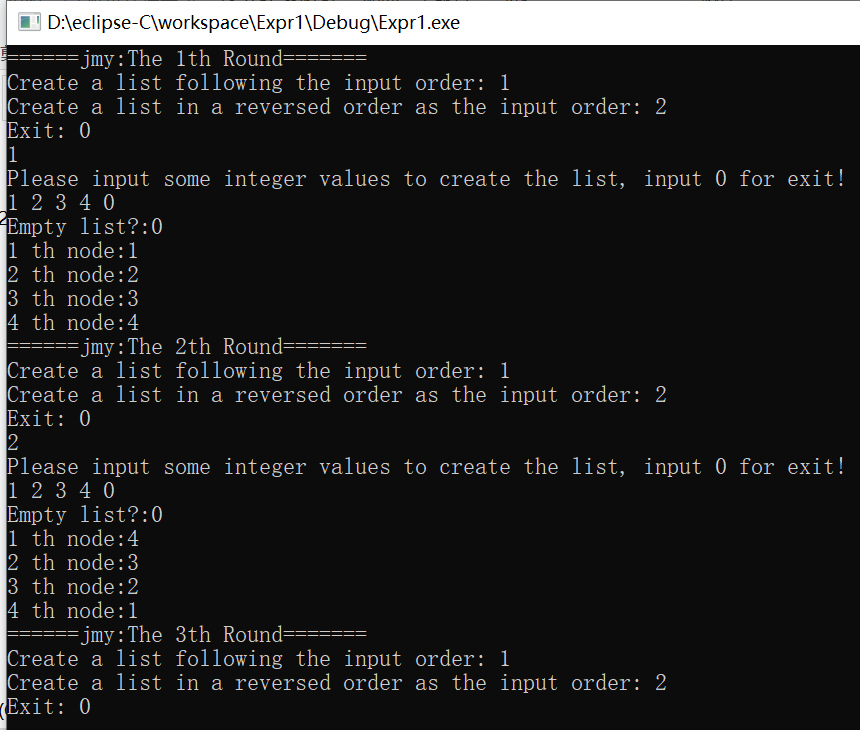
(2) Write a program to create a linked list based on user input data (construct a list according to the order of numbers typed in), and then print out the list. In this program, you should use functions of the linked list.

(3) The arraylist.h is provided for you, and you should implement all of its functions and put them into .c files. (define and implement auxiliary operations when needed)

(4) Write a program to create an array list based on user input data (construct a list according to the order of numbers typed in), and then print out the list. In this program, you should use functions of the array list.

**(5) Discuss the differences between the implementations of array list and linked list.**

Hints: you can read the code snippets in Chapter 3 for references. The sample output can be:



**【Problem 2】: Implement a desk calculator.**

**Requirements:** (1) The input to your calculator is an infix expression.

(2) The expression may include the parenthesis and the exponentiation operator.

Hints: The stackli.h stackli.c are provided for you, which implements the stack ADT based on the linked list. **You can extend the current ADT when needed (i.e., adding more functions).**

**Experimental report:**

**[Task 1]**

1. **Task description:** *//clarify your understanding of the tasks to solve problem1*
2. **Method:** *//describe your ideas to solve the problem, you can describe your idea by text, or depict your algorithm.*
3. **Implementation:** *//present and explain your key code here*
4. **Results:** *//run your implementation with some input, then show and explain your outputs.*

**[Task 2]**

1. **Task description:** *//clarify your understanding of the tasks to solve problem1*
2. **Method:** *//describe your ideas to solve the problem, you can describe your idea by text, or depict your algorithm.*
3. **Implementation:** *//present and explain your key code here*
4. **Results:** *//run your implementation with some input, then show and explain your outputs.*

**[Discussions]:** //*talk about your observations from this experiments, for example, what kind of experiences you have learned? Are there any key points that you should pay attention to in your study or programming exercises in future?*

**Evaluation:**

Functionality

Completeness

Readability