Lab 04

Linked List

Tran Thanh Tung

1. **Introduction:**

   You have learnt about Linked List data structure. This lab will help you go through some practice manners of Linked List data in Java programming language.

2. **Problems:**

   Problem I: Circular Linked list

   A circular list is a linked list in which the last link points back to the first link. There are many ways to design a circular list. Sometimes there is a pointer to the "start" of the list. However, this makes the list less like a real circle and more like an ordinary list that has its end attached to its beginning.

   Make a class for a singly linked circular list that has no end and no beginning. The only access to the list is a single reference, current, that can point to any link on the list. This reference can move around the list as needed. (See next problem for a situation in which such a circular list is ideally suited.).

   Your list should handle insertion, searching, and deletion. You may find it convenient if these operations take place one link downstream of the link pointed to by current. (Because the upstream link is singly linked, you can't get at it without going all the way around the circle.) You should also be able to display the list (although you'll need to break the circle at some arbitrary point to print it on the screen). A step() method that moves current along to the next link might come in handy too.

   Problem II: Josephus Problem

   The Josephus Problem is a famous mathematical puzzle that goes back to ancient times. There are many stories to go with the puzzle. One is that Josephus was one of a group of Jews who were about to be captured by the Romans. Rather than be enslaved, they chose to commit suicide. They arranged themselves in a circle and, starting at a certain person, started counting off around the circle. Every nth person had to leave the circle and commit suicide. Josephus decided he didn't want to die, so he arranged the rules so he would be the last person left. If there were (say) 20 people, and he was the seventh person from the start of the circle, what number should he tell them to use for counting off? The problem is made much more complicated because the circle shrinks as the counting continues. Create an application that uses a circular linked list (like that in Programming Project 5.3) to model this problem. Inputs are the number of people in the circle, the number used for counting off, and the number of the person where counting starts (usually 1). The

output is the list of persons being eliminated. When a person drops out of the circle, counting starts again from the person who was on his left (assuming you go around clockwise).
Here's an example. There are seven people numbered 1 through 7, and you start at 1 and count off by threes. People will be eliminated in the order 4, 1, 6, 5, 7, 3. Number 2 will be left