

# The Josephus Problem

Daniel Yee  
CIS 2168

## Purpose

In this assignment, you will solve the Josephus Problem by implementing a circular linked list.

## Background

The Josephus problem is named after the Jewish historian Titus Flavius Josephus (37 - 100 CE), who led the Jewish forces in Galilee during the First Jewish-Roman War.

According to his account, Josephus and forty other soldiers were about to be captured by the Romans during the siege of Yodfat. They decided they would rather die than be captured and committed suicide by drawing lots. Josephus and one other soldier remained in the end, decided that a life under the Romans as a slave was better than no life, and surrendered. Josephus became a slave and gained his freedom a few years later. He later went on to write several documents which give insight to the religions of that time period.

While the exact details of the lot-drawing process are unknown and still debated by historians, the following approach is generally believed to be how it was arranged.

The soldiers formed a circle and counted around the circle some predetermined number. When this number is reached, that soldier was then killed. The count starts over from the soldier that was just killed.

## The Problem

Implement a circular linked list with the provided code to represent the circle of soldiers and simulate the problem. Your program should use two parameters,  $n$  soldiers and  $k$ , where  $k$  is the number of counts. In other words, we successively remove the  $k$ th person from the list.

The problem is considered solved when there are only two soldiers left.

If you are interested in more information, the Numberphile Youtube channel discusses the scenario and the underlying math with animated graphics.

## The Code

The CircularLinkedList.java file has been provided you give you a starting point. You need to finish implementing the circular linked list and the iterator.

In your main() method, prompt the user for the number of soldiers, n, and how many to count, k. After each round, print out which soldier was killed and the resulting circle of surviving soldiers. Consider the problem solved when only two soldiers remain: Josephus and his companion. At this point, print out a message indicating which two soldiers survived.

The Iterator should be implemented to allow wrapping around the CircularLinkedList. Be sure to create the iterator by calling the appropriate CircularLinkedList method after you added all the nodes to the list.

## Examples

For group of soldiers n = 5 and count k = 2:

```
1 ==> 2 ==> 3 ==> 4 ==> 5 ==>
1 ==> 3 ==> 4 ==> 5 ==>
1 ==> 3 ==> 5 ==>
3 ==> 5 ==>
```

For a group of soldiers n = 13 and k = 2:

```
1 ==> 2 ==> 3 ==> 4 ==> 5 ==> 6 ==> 7 ==> 8 ==> 9 ==> 10 ==> 11 ==> 12 ==> 13 ==>
1 ==> 3 ==> 4 ==> 5 ==> 6 ==> 7 ==> 8 ==> 9 ==> 10 ==> 11 ==> 12 ==> 13 ==>
1 ==> 3 ==> 5 ==> 6 ==> 7 ==> 8 ==> 9 ==> 10 ==> 11 ==> 12 ==> 13 ==>
1 ==> 3 ==> 5 ==> 7 ==> 8 ==> 9 ==> 10 ==> 11 ==> 12 ==> 13 ==>
1 ==> 3 ==> 5 ==> 7 ==> 9 ==> 10 ==> 11 ==> 12 ==> 13 ==>
1 ==> 3 ==> 5 ==> 7 ==> 9 ==> 11 ==> 12 ==> 13 ==>
1 ==> 3 ==> 5 ==> 7 ==> 9 ==> 11 ==> 13 ==>
3 ==> 5 ==> 7 ==> 9 ==> 11 ==> 13 ==>
3 ==> 7 ==> 9 ==> 11 ==> 13 ==>
3 ==> 7 ==> 11 ==> 13 ==>
3 ==> 7 ==> 11 ==>
3 ==> 11 ==>
```

## Rubric

### 90 points

The correct soldier is killed each round. The last two soldiers remaining are the correct one to have survived.

### 10 points

Code is neat and properly indented.

## **Presentation**

We'll ask you some questions to make sure you understand circular linked lists and iterators.