

# Josephus Problem

ArrayList<Object>

“The **Josephus problem** (or **Josephus permutation**) is a theoretical problem related to a certain counting-out game. The problem is named after Flavius Josephus, a Jewish historian living in the 1st century. According to Josephus’ account of the siege of Yodfat...” ( And see the rest in Wikipedia, [https://en.wikipedia.org/wiki/Josephus\\_problem](https://en.wikipedia.org/wiki/Josephus_problem)).

A simple (and one of the more gentle) descriptions of the problem goes as follows:

In order to determine the next king, all the knights are arranged around a circular-shaped table. The jester then goes around the table. Every knight he taps needs to leave the table. He starts with the first knight, and does NOT tap him. He then taps the second knight. Skips the third, and taps the fourth, and so on. He keeps going around the table, tapping every other knight, until only one knight is left, and is the chosen one to be the king.

In this assignment, you will find the new king using 3 (three!) different methods, which should all give the same result:

1. ArrayList – Using brute-force modeling of the problem.
2. Binary representation - Using binary circular left shift.

Hint: The functions `Integer.toBinaryString(n)` and `Integer.parseInt(str,2)` are available to you.

3. Formula – Yes, there is a formula for that. You can find the formula on the Wiki page.

Hint: The functions `Math.log()`, `Math.pow()`, and `Math.floor()` are available.

Write it all in one class, and use the following as the `main()` function:

```
import java.util.ArrayList;
public class Main
{
    static public void main(String[] args)
    {
        System.out.println(josephus_ArrayList(9)); // 3
        System.out.println(josephus_CircShift(11)); // 7
        System.out.println(josephus_LogN(14)); //13
    }
}
```

## Submit

1. Code listing which includes the 3-different methods
2. The answer for #knights=515.

The following might be of use:

| #Knights | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| King     | 1 | 1 | 3 | 1 | 3 | 5 | 7 | 1 | 3 | 5  | 7  | 9  | 11 | 13 | 15 |