

CSCI 362: THE JOSEPHUS PROBLEM

OVERVIEW

Suppose N people, numbered 1 through N , stand in a circle. Further, suppose every k 'th person is executed until one remains. Input N and k from the user, then clearly indicate the order of execution and the sole survivor. For $N = 5$ and $k = 3$, the order of execution is: 3, 1, 5, 2 . 4 is the survivor. Be careful to handle cases such as $k \geq N$ and $N = 1$.

IMPLEMENTATION

```

/* Simulate the Josephus problem modeled as a std::list.
 * This function will modify the passed list with only a
 * single survivor remaining.
 *
 * @param circle -- the linked list of people
 * @param k -- skip amount. NOTE:  $k > 0$ 
 *
 * @return a list of those who are executed in chronological order
 */
template <typename T>
std::list<T>
execute (std::list<T>& circle, int k);

/* entry point to the Josephus problem from the autograder
 *
 * @param N -- number of people in the circle
 * @param k -- skip amount. NOTE:  $k > 0$ 
 */
int
josephus (int N, int k);

```

- n – number of people
- k – skip amount **NOTE:** $k > 1$
- You **MUST** use a `std::list` to model the execution ring.

- Use `iterator` s to traverse through the people.
 - To maintain the circle, when the `iterator` equals `end()` set it back to `begin()`
 - When “executing” a person, use the `erase()` function provided by list. NOTE: this will return the “next” iterator – be careful to update it and NOT advance by “K”

STYLE

Use good programming style:

- Write comments
- Choose **mnemonic**, meaningful variable names (e.g. `balance`, `interestRate`)
- Remember to include a comment block at the top of your program that includes:
 - Your name
 - The course
 - The last date of modification
 - The assignment name
 - A brief description of what your program does
- **FORMAT YOUR CODE CONSISTENTLY.**

WRITEUP

In a comment DIRECTLY ABOVE your “execute” function give the complexity of your algorithm along with a DERIVATION and EXPLANATION. Specify function $T(N, k)$, which represents the number of link traversals as a function of N and k . To derive T , add a counter to your code that maintains the number of links traversed. While you are trying to determine T , output the count. When you submit your code, REMOVE the code that OUTPUTS the count, but INCLUDE the code that computes the count. It’s OK if your T function doesn’t EXACTLY match your count, but it should be close — explain any discrepancies.

- [60pts] Correctness
- [15pts] Writeup / Code Formatting / Style / Documentation

- **NOTE:** if your program does not compile/run, the highest score you can earn will be a 10/75