

Assignment 6

CS329e - Elements of Software Design

Circular Linked List

(100 points)

Due Date on Canvas and Gradescope

1 Description

The problem that you are going to solve is known as Josephus problem and it is stated as follows. There is a group of soldiers surrounded by an overwhelming enemy force. There is no hope for victory without reinforcements, so they make a pact to commit suicide.

They form a circle and a number n is picked from a hat. One of their names is picked at random. Beginning with the soldier whose name is picked, they begin to count clockwise around the circle. When the count reaches n , that soldier is executed, and the count begins again with the next man. The process continues so that each time the count reaches n , a man is removed from the circle. Once a soldier is removed from the circle he is no longer counted. The last soldier remaining was supposed to take his own life. According to legend the last soldier remaining was Josephus and instead of taking his own life he joined the enemy forces and survived.

The problem is: given a number n , the ordering of the men in the circle, and the man from whom the count begins, to determine the order in which the men are eliminated from the circle and which man escapes. For example, suppose that n equals 3 and there are five men named A, B, C, D, and E. We count three men, starting at A, so that C is eliminated first. We then begin at D and count D, E, and back to A, so that A is eliminated next. Then we count B, D, and E (C has already been eliminated) and finally B, D, and B, so that D is the man who escapes.

You will use a circular linked list. You have worked on the linear linked list in your previous home work. To make a circular linked list you need to make the next field in the last link of the linked list point back to the first link instead of being null. From any point in a circular list it is possible to reach any other point in the list. Thus any link can be the first or last link. One useful convention is to let the external pointer to the circular list point to the last link and to allow the following link be the first link. We also have the convention that a null pointer represents an empty circular list.

Instead of giving the soldiers names you will assign them numbers serially starting from 1 (one).

This way you can use the Link class that we discussed. In your program you will read the data from a file called **josephus.in**.

1. The first line gives the number of soldiers.
2. The second line gives the soldier from where the counting starts.
3. The third line gives the elimination number.

You will create a circular linked list having the number of soldier specified. Your program will print out the order in which the soldiers get eliminated.

You should use the attached python template for implementation program (*File: Josephus.py*).

You will modify the functions *insert()*, *find()*, and *delete()* from the *LinkedList* class. The main workhorse of the program will be the *delete_after()* method. This takes the starting Link and the elimination number *n* and deletes the *n*th Link from the starting Link. It returns the data of the deleted Link and the next Link after the Link that was deleted.

Input:

Suppose the input file was the following:

```
1 12
2 1
3 3
```

Output:

Then your output will be:

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

The last line of your output will be the number of the soldier that escapes.

Pair Programming

For this assignment you may work with a partner. Both of you must read the paper on Pair Programming¹ and abide by the ground rules as stated in that paper. If you are working with a partner then only one of you will be submitting the code. But make sure that your partner's name and UT EID is in the header. If you are working alone then remove the partner's name and eid from the header.

1.1 Turnin

Turn in your assignment on time on Gradescope system on Canvas. For the due date of the assignments, please see the Gradescope and Canvas systems.

1.2 Academic Misconduct Regarding Programming

In a programming class like our class, there is sometimes a very fine line between "cheating" and acceptable and beneficial interaction between students (In different assignment groups). Thus, it is very important that you fully understand what is and what is not allowed in terms of collaboration with your classmates. We want to be 100% precise, so that there can be no confusion.

¹Read this paper about Pair Programming <https://collaboration.csc.ncsu.edu/laurie/Papers/Kindergarten.PDF>

The rule on collaboration and communication with your classmates is very simple: you cannot transmit or receive code from or to anyone in the class in any way – visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading code to someone) or in any other way we have not yet imagined. Any other collaboration is acceptable.

The rule on collaboration and communication with people who are not your classmates (or your TAs or instructor) is also very simple: it is not allowed in any way, period. This disallows (for example) posting any questions of any nature to programming forums such as **StackOverflow**. As far as going to the web and using Google, we will apply the **"two line rule"**. Go to any web page you like and do any search that you like. But you cannot take more than two lines of code from an external resource and actually include it in your assignment in any form. Note that changing variable names or otherwise transforming or obfuscating code you found on the web does not render the "two line rule" inapplicable. It is still a violation to obtain more than two lines of code from an external resource and turn it in, whatever you do to those two lines after you first obtain them.

Furthermore, you should cite your sources. Add a comment to your code that includes the URL(s) that you consulted when constructing your solution. This turns out to be very helpful when you're looking at something you wrote a while ago and you need to remind yourself what you were thinking.

We will use the following Code plagiarism Detection Software to automatically detect plagiarism.

- **Stanford MOSS**

<https://theory.stanford.edu/~aiken/moss/>

- **Jplag - Detecting Software Plagiarism**

<https://github.com/jplag/jplag> and <https://jplag.ipd.kit.edu/>