### Circular Linked List ( Due 09 Nov 2020 )

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](https://www.cs.utexas.edu/users/mitra/csFall2020/cs313/assgn/josephus.in). The first line gives the number of soldiers. The second line gives the soldier from where the counting starts. 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. The template for your program will look like this:

import sys

class Link(object):

class CircularList(object):

# Constructor

def \_\_init\_\_ ( self ):

# Insert an element (value) in the list

def insert ( self, data):

# Find the link with the given data (value)

def find ( self, data ):

# Delete a link with a given data (value)

def delete ( self, data ):

# Delete the nth link starting from the Link start

# Return the next link from the deleted Link

def delete\_after ( self, start, n ):

# Return a string representation of a Circular List

def \_\_str\_\_ ( self ):

def main():

# read number of soldiers

line = sys.stdin.readline()

line = line.strip()

num\_soldiers = int (line)

# read the starting number

line = sys.stdin.readline()

line = line.strip()

start\_count = int (line)

# read the elimination number

line = sys.stdin.readline()

line = line.strip()

elim\_num = int (line)

# your code

if \_\_name\_\_ == "\_\_main\_\_":

main()

To run this code on the command line on the Mac you will do

python3 Josephus.py < josephus.in

To run the code on the command line on a Windows machine you will do

python Josephus.py < josephus.in

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 nth Link from the starting Link. It returns the ***next*** Link after the Link that was deleted.

Suppose the input file was the following:

12

1

3

Then your output will be:

3

6

9

12

4

8

1

7

2

11

5

10

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

For this assignment you may work with a partner. Both of you must read the paper on [Pair Programming](https://www.cs.utexas.edu/users/mitra/csFall2020/cs313/assgn/PairProg-CACM-1999.pdf) and abide by the ground rules as stated in that paper.

The file that you will be uploading will be called **Josephus.py**. We are looking for clean and structured design. The file will have a header of the following form:

# File: Josephus.py

# Description:

# Student Name:

# Student UT EID:

# Partner Name:

# Partner UT EID:

# Course Name: CS 313E

# Unique Number:

# Date Created:

# Date Last Modified:

If you are working with a partner you will be submitting a single program (from either account) but make sure that you have your partner's name and eid in your program. If you are working alone, then remove the two lines that has the partner's name and eid in the header.

Use the [Canvas](http://canvas.utexas.edu/)system to submit your **Josephus.py** file. We should receive your work by 11 PM on Monday, 09 Nov 2020. There will be substantial penalties if you do not adhere to the guidelines. Remember Python is case sensitive. The name of your file must match exactly what we have specified.

* Your Python program should have the proper header.
* Your code must run before submission.
* You should be submitting your file through the web based *Canvas* program. We will not accept files e-mailed to us.

#### References

* [Josephus Problem in Wikipedia](http://en.wikipedia.org/wiki/josephus_problem)
* [Josephus Problem in Wolfram Mathworld](http://mathworld.wolfram.com/JosephusProblem.html)