

Assignment III — Linked list

TA: Hank(s0963360045@gmail.com)

Deadline: 2019 Nov. 12, 11:59pm

In this homework, you will be asked to implement an elimination game by using the operations of the linked list. The operations including **odd/even order change** and **rotation**. After changing the order and rotate the list by n nodes, start the elimination game to eliminate nodes in the **linked list**. The description of the elimination game is shown below.

Elimination game

There are n players standing in a circle waiting to be killed. The counting out begins at some point in the circle and proceeds around the circle in a direction. In each step, k number of players are skipped, and the next player of the last skipped player is killed. The elimination proceeds around the circle (which is becoming smaller and smaller as the killed players are removed), until only the last player remains, who is the final survivor of the elimination game. Figure 1 illustrates the elimination game where $k = 2$ and $n = 5$.

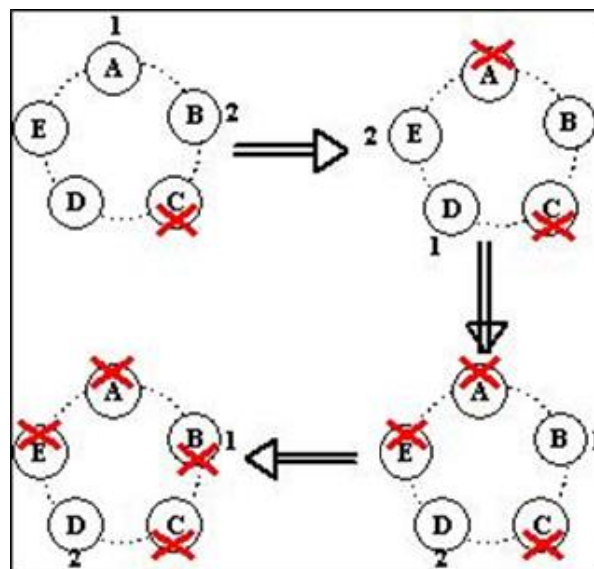


Figure 1. An illustration of the elimination game.

Implementation Step

1. Reading input data

Please write a program that can read data from standard input (stdin). In this program, you are asked to read multiple input data from a given file. The first three inputs are the **action codes** to manipulate the linked list. As the example is shown in Figure 2, the first number “2” represents that the linked list needs to rotate counter-clockwise by 2 nodes. For example, if the given linked list is 10->20->30->40->50->60, the list should be modified to 30->40->50->60->10->20. The second number “1” represents the clockwise search direction of the elimination game, while 2 represents the counter-clockwise direction. The third number “1” is the number of players to be skipped in the elimination game. The fourth line “Yuling” to the last line “Jim” are the players to participate in the elimination game, and please store them into **a linked list**.



```
2
1
1
Yuling
Tom
Henry
Hank
WenJ
Cody
Sine
Ben
Edward
Peter
Helen
Patty
Haoze
Jim
```

Figure 2: The required input data

2. Performing odd/even order change and rotation

operations

Next, you are asked to implement two operations of the linked list. The first operation is **odd/even order change**. In this operation, you need to put the odd nodes sequentially to the head of the list and put the even nodes sequentially to the backside. Figure 3 illustrates the process of the odd/even order change operation of the linked list.



Figure 3: The illustration of odd/even order change operation.

The second one is **the rotation operation**. Given a k value, rotate the linked list in a counter-clockwise direction by k nodes. Figure 4 illustrates the rotation operation when k = 4.

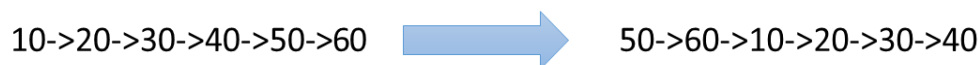


Figure 4: The illustration of the rotate operation.

For this assignment, after reading the participants from the given file (the format is as illustrated in Figure 2), print them out in the first line. Subsequently, perform the odd/even order change operation on the linked list. After changing the order of the linked list, run the rotation operation by using k = 2, which is read from the first line of the given input file. The results of performing the two operations are shown in Figure 5.

```
Yuling -> Tom -> Henry -> Hank -> WenJ -> Cody -> Sine -> Ben -> Edward ->
Peter -> Helen -> Patty -> Haoze -> Jim
Yuling -> Henry -> WenJ -> Sine -> Edward -> Helen -> Haoze -> Tom -> Hank
-> Cody -> Ben -> Peter -> Patty -> Jim
WenJ -> Sine -> Edward -> Helen -> Haoze -> Tom -> Hank -> Cody -> Ben ->
Peter -> Patty -> Jim -> Yuling -> Henry
```

Figure 5: The results of performing the odd/even order change and rotation operations.

3. Eliminating the players from the linked list

In this step, perform the elimination game on the resulting linked list. Before starting the elimination game, set the search direction and the number of skipped players based on the settings in the second line and the third line of the input file, respectively for the elimination game. In this case, the number of skipped players is 1, so the first eliminated player is “Sine”. Please print out the killed player in every step. After the elimination, print out the final survivor in the last line. Figure 6 shows the result of the elimination game.

```
Sine died
Helen died
Tom died
Cody died
Peter died
Jim died
Henry died
Edward died
Hank died
Patty died
WenJ died
Ben died
Haoze died
Yuling
```

Figure 6: The execution result of the elimination game.

Test case examples

Sample Input:	Sample Output:
<p>Read input data.</p> <p>The input format is shown as follow:</p> <p><NumRoation></p> <p><NumDirection></p> <p><NumSkip></p> <p><Players></p>	<pre>2 1 1 Yuling Tom Henry Hank WenJ Cody Sine Ben Edward Peter Helen Patty Haoze Jim</pre>

Execution results.	<pre>Yuling -> Tom -> Henry -> Hank -> WenJ -> Cody -> Sine -> Ben -> Edward -> Peter -> Helen -> Patty -> Haoze -> Jim</pre>
The output format is shown as follow:	<pre>Yuling -> Henry -> WenJ -> Sine -> Edward -> Helen -> Haoze -> Tom -> Hank -> Cody -> Ben -> Peter -> Patty -> Jim</pre>
<Origin list> <Odd/Even list> <Rotated list> <Killed Players> <Final Survivor>	<pre>WenJ -> Sine -> Edward -> Helen -> Haoze -> Tom -> Hank -> Cody -> Ben -> Peter -> Patty -> Jim -> Yuling -> Henry Sine died Helen died Tom died Cody died Peter died Jim died Henry died Edward died Hank died Patty died WenJ died Ben died Haoze died Yuling</pre>

4. Readme, comments and style (5%)

An indicator for good source code is readability. To keep source code maintainable and readable, you should add comments to your source code where reasonable. A consistent coding style also helps a lot when tracing the source code. For this assignment, please also compose a readme file in *.txt format and name it as "README.txt". This file should contain a brief explanation of how to use your program. Please remember to have your source code comments and readme file in English.

5. Submission

To submit your files electronically, login DomJudge website through the following url :

<https://108-1-ds-judge.ate.cs.ccu.edu.tw/>

Press the submit button and choose the homework questions you want to submit. After submitting your code, DomJudge will give you a result to tell you whether your code is correct or not. However, during the demo time, your code will be evaluated by different sets of test cases. Please make sure your code can work correctly based on the description above. Additionally, you must compress your code and the README file into a **zip** file and upload it to Ecourse2.

6. Grading policy

40% - Operations of the linked list.

55% - Elimination game.

5% - Readme file, coding style, and comments in the source code.

A document named "readme.txt". You should describe the details of your project in your readme file in English.