

CP 3: Data Structures and Algorithms

Expected time to solve: 2 hours.

In computer science and mathematics, the Josephus Problem (or Josephus permutation) is a theoretical problem related to a certain counting-out game. There are people standing in a circle waiting to be executed (to be killed that is). The counting out begins at some point in the circle and proceeds around the circle in a fixed direction. In each step, a certain number of people are skipped and the next person is executed. The elimination proceeds around the circle (which is becoming smaller and smaller as the executed people are removed), until only the last person remains, who is given freedom. The task is to choose the place in the initial circle so that you are the last one remaining and so survive.



In the example shown above, 41 people are standing in a circle and the game has started from **person 1**. Suppose that we have to execute every third person. Starting with person 1, counting is started 1, 2, and the third person (in this case **person 3**) is killed. Killing means that his place has become empty. After killing person 3, person 2 and person 4 will be standing adjacent to each other. The process continues from the last killed person. Hence person 4 and 5 are saved and the **person 6** is killed. At the end of the algorithm, only 1 person will be left. The person who is left in the end is the one who survives, all other are killed.

- I. Solve this problem using:
 1. **Any Data Structure of your choice**
 2. The name of your main .cpp should be the name of the data structure you used. For example, the names of your file could be any of these: array.cpp, singlyLinkedList.cpp, singlyCircularLinkedList.cpp, doublyCircularLinkedList.cpp, bst.cpp etc... (you may opt for any other data structure as well)
- II. In the first few lines of the main program, explain the rough algorithm of your solution in comments
- III. Make a menu driven program having the following options:
 1. Start the game
 2. Exit the program
- IV. On pressing 1, the following inputs should be taken from the user:
 - The number of people standing in the circle **[See Hint 2]**
 - The starting person **[In the example, it was 1, it can be different from 1]**
 - How many persons are NOT to be killed starting from the initial person that was chosen **[In the example, TWO persons in between were kept alive and 3rd person was killed. So, the input in this case should be 2.]**
 - You should output the person that was the winner (the one who remained alive at the end) before starting the new game.
- V. On pressing 2, the program should exit.

Hints: You have to create the data entries yourself. In case of the example described above, you will create 41 data elements for your chosen data structure in the main function. You may use a loop and place an integer number in the data field for your convenience. The number can be used to get the exact position/number of the person.