

National University of Computer and Emerging Sciences, Lahore Campus



Course: Data Structures

Program: BS(CS), BS(SE)

Due Date 22nd October 2023 at
11:59 pm

Course Code: CS 2001

Semester: Fall 2023

Total Marks: 30

Important Instructions:

1. Submit your code in a zip file named as your roll number.
2. You are not allowed to copy solutions from other students. We will check your code for plagiarism using plagiarism checkers. If any sort of cheating is found, negative marks will be given to all students involved.
3. Late submission is not allowed.

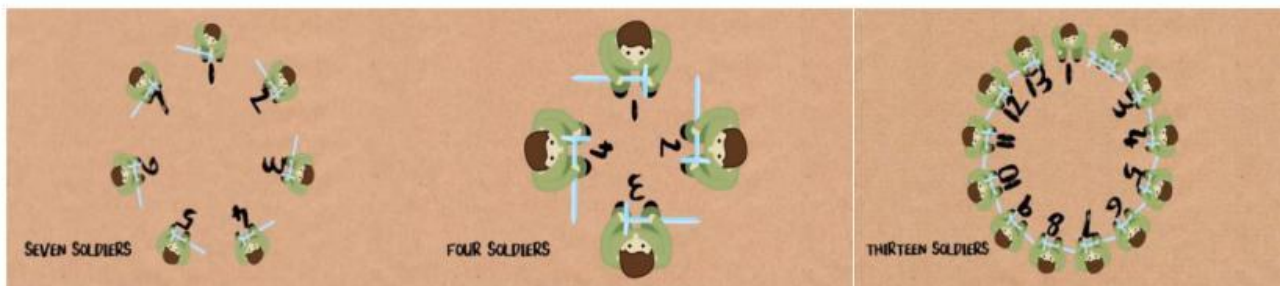
Problem 1: Josephus Problem

History: This problem is named after Flavius Josephus, a Jewish historian living in the 1st century. According to Josephus' account of the siege of Yodfat, he and his 40 soldiers were trapped in a cave by Roman soldiers. They chose suicide over capture, and settled on a serial method of committing suicide by drawing lots. Josephus states that by luck or possibly by the hand of God, he and another man remained until the end and surrendered to the Romans rather than killing themselves. This is the story given in Book 3, Chapter 8, part 7 of Josephus' The Jewish War (writing of himself in the third person)

In computer science and mathematics, the Josephus problem (or Josephus permutation) is a theoretical problem related to a certain counting-out game.

https://en.wikipedia.org/wiki/Josephus_problem

1. People (any number $N > 1$) are standing in a circle waiting to be executed.
2. Counting begins at a specified point (S selected randomly) in the circle and proceeds around the circle in a specified direction. You can assume that it is clockwise.



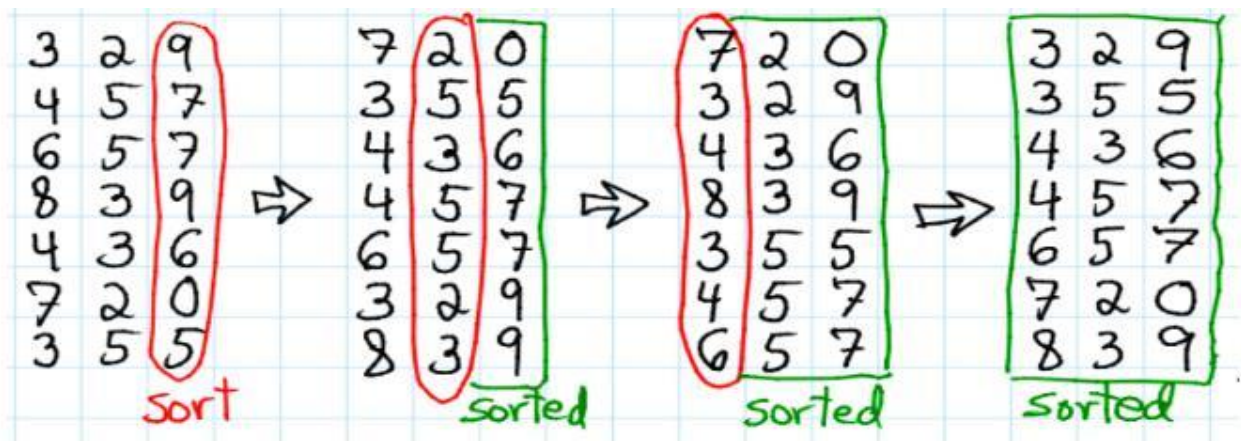
3. After a specified number of people are skipped, say $(k-1)$, the next (kth) person is executed. The procedure is repeated with the remaining people, starting with the next person, going in the same direction and skipping the same number of people $(k-1)$, until only one person remains, and is freed.

Write a C++ program that solves the Josephus problem by using a Queue data structure. Your program should take N and K input from the user. Design a graphical user interface that will update with every move as shown in figure above.

Problem 2: (Radix Sort)

Radix sort is a sorting algorithm, which sorts the keys based on the values of digits in keys. It takes a queue containing n keys to be sorted, where each key consists of k number of digits, and there could be m possible values for each digit 0 through $m-1$. Radix sort uses an array consisting of (m) queues for sorting of these keys.

For example, if each key contains $k = 3$ digits and each individual digit has $m=10$ possible values 0 to 9 , then it will use an array consisting of 10 queues $0-9$ in the sorting process of considering all digits one by one as follows.



You have to implement two Radix sorts which can sort **integer** and **string** keys with any value of n and k .

CODE DESIGN GUIDELINES

- Do template-based programming
- Code should be properly indented and commented (2 marks for this)
- Make sure there are no memory leaks or dangling pointers
- Don't cheat or take too much unnecessary help from your friends