Name: Sahil jain Roll No: 19 Class: TY-B

Josephus Problem

1. Recursive Approach

package Josephus;

import java.util.Scanner;

class recursive {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter number of players and the number of skips");

        int n = sc.nextInt();

        int k = sc.nextInt();

        System.out.println(findTheWinner(n, k));

        sc.close();

    }

    public static int findTheWinner(int n, int k) {

        return helper(n, k) + 1;

    }

    public static int helper(int n, int k) {

        if (n == 1)

            return 0;

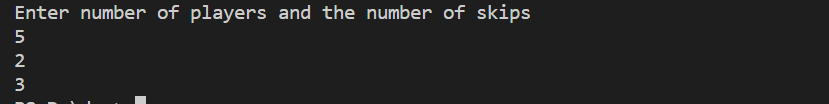
        else

            return (helper(n - 1, k) + k) % n;

    }

}

Output:



1. Iterative:
2. package Josephus;
3. import java.util.LinkedList;
4. import java.util.Queue;
5. import java.util.Scanner;
6. public class iterative {
7. public static void main(String[] args) {
8. Scanner sc = new Scanner(System.in);
9. System.out.println("Enter number of players and the number of skips");
10. int n = sc.nextInt();
11. int k = sc.nextInt();
12. System.out.println(findTheWinner(n, k));
13. sc.close();
14. }
15. public static int findTheWinner(int n, int k) {
16. Queue<Integer> q = new LinkedList<>();
17. for (int i = 1; i <= n; i++) {
18. q.offer(i);
19. }
20. while (q.size() != 1) {
21. for (int i = 1; i < k; i++)
22. q.offer(q.poll());
23. q.poll();
24. }
25. return q.poll();
26. }
27. }

Output:

