

```

import java.util.ArrayDeque;
import java.util.Queue;
public class PotatoQueue {
    public static String hotPotato(String[] names, int mN) {
        Queue<String> queue = new ArrayDeque<>();
        for (String name : names) {
            queue.add(name);
        }
        while (queue.size() > 1) {
            for (int i = 0; i < mN - 1; i++) {
                String currentPlayer = queue.poll();
                queue.add(currentPlayer);
            }
            String removedChild = queue.poll();
            System.out.println(removedChild + " is out!");
        }
        return queue.poll();
    }
    public static void main(String[] args) {
        String[] names = {"Circuit", "Prachi", "Jirimali", "Arashdeep", "Kaur",
"Abhijeet"};
        int mN = 3;
        String winner = hotPotato(names, mN);
        System.out.println("The winner is " + winner + "!");
    }
}

```

```

import java.util.Stack;
public class ParenthesesStacks {
    public static boolean isBalanced(String parentheses) {
        Stack<Character> stack = new Stack<>();
        String opening = "{[";
        String closing = ")]";
        for (char c : parentheses.toCharArray()) {
            if (opening.indexOf(c) != -1) {
                stack.push(c);
            } else if (closing.indexOf(c) != -1) {
                if (stack.isEmpty()) {
                    return false;
                }
                char top = stack.pop();
                if (opening.indexOf(top) != closing.indexOf(c)) {
                    return false;
                }
            }
        }
        return stack.isEmpty();
    }
    public static void main(String[] args) {

```

```
String[] bracketsArr = {
    "(() () () ())",
    "((((())))",
    "(() (((()) ()))",
    "(((((((())",
    "())",
    "(() () () ("
};
for (String test : bracketsArr) {
    if (isBalanced(test)) {
        System.out.println(test + ": Balanced");
    } else {
        System.out.println(test + ": Not Balanced");
    }
}
}
```