# TY. B. Tech.

## Design & Analysis of Algorithm

### Assignment No: 4

***Date of Submission: 15/04/2023***

| Roll. No. | Gr. No. | Div | Name |
|---|---|---|---|
| 12 | 12011336 | C | Niraj Kirit Patil |

**Academic Year: 2021-22** **Semester: II**

# Assignment No: 4

Josephus Problem

## Code:

```rust
/*

We start at the last step, assuming only 2 people are left. Hence, i
represents total number of people currently in the circle.

With each iteration, we go back in time, to the previous round and try
to determine the winner at that round.

We repeat this until i reaches n, at which point we find the  winner with
n people in the circle

*/

use std::env::Args;

#[derive(Clone)]
1 implementation
struct Params {
    n:  usize,
    k: usize,
}



fn joseph(params: Params) -> usize{
    //let mut prms = params.clone();
    let mut ans = 0;
    let mut i=2;

    //here we calculate ans, which reps next person to kill
    while i <= params.n {
        ans = (ans + params.k) % i;
        i += 1;
    }

    return ans + 1;
}
```

```
▶ Run | Debug
fn main() {
    let args: Vec<String> = std::env::args().collect();
    let n = args.get(1).unwrap().parse::<usize>().unwrap();
    let k = args.get(2).unwrap().parse::<usize>().unwrap();


    let prms = Params {
        n,
        k
    };
    let ans = joseph(prms);

    println!("ans: {}", ans);
}
```

**Output:**

```
t_ws/DAA_59276fed/bin quicksort.QuickSort
Original Array: [4 2 7 3 1 6 ]
Sorted Array: [1 2 3 4 6 7 ]%
```

(This page marks the end of the assignment)