Given the total number of persons **n** and a number **k** which indicates that **k-1** persons are skipped and **k^th** person is killed in circle in a fixed direction.

The task is to choose the **safe place in the circle** so that when you perform these operations starting from **1st place** in the circle, you are the last one remaining and survive.

Basically the last remaining person survives.
∴ every call reduces the no. of people (n-1)
and at n == 1 → return 1.

Now this is to be adjusted to original the solution function.

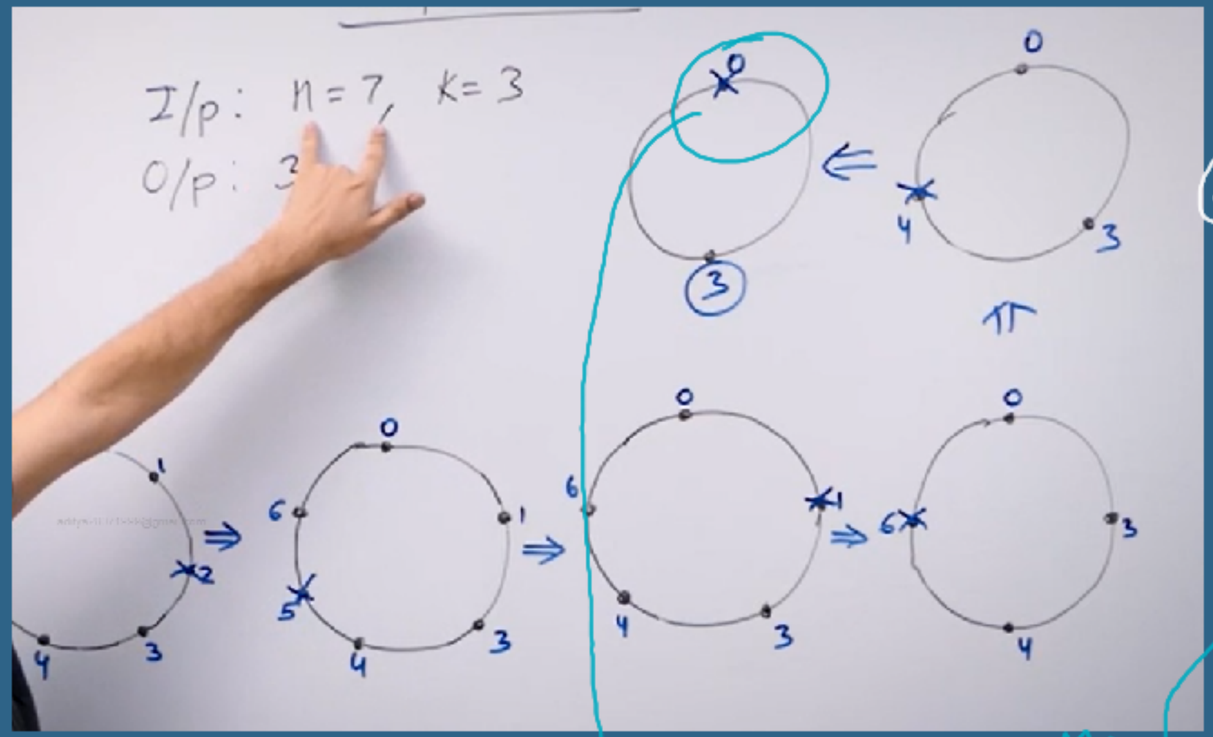
If n = 7 and k = 3, then the safe position is 4. The persons at positions 3, 6, 2, 7, 5, 1 are killed in order, and person at position 4 survives.

```
josephus(n, k) = (josephus(n - 1, k) + k-1) % n + 1

josephus(1, k) = 1
```

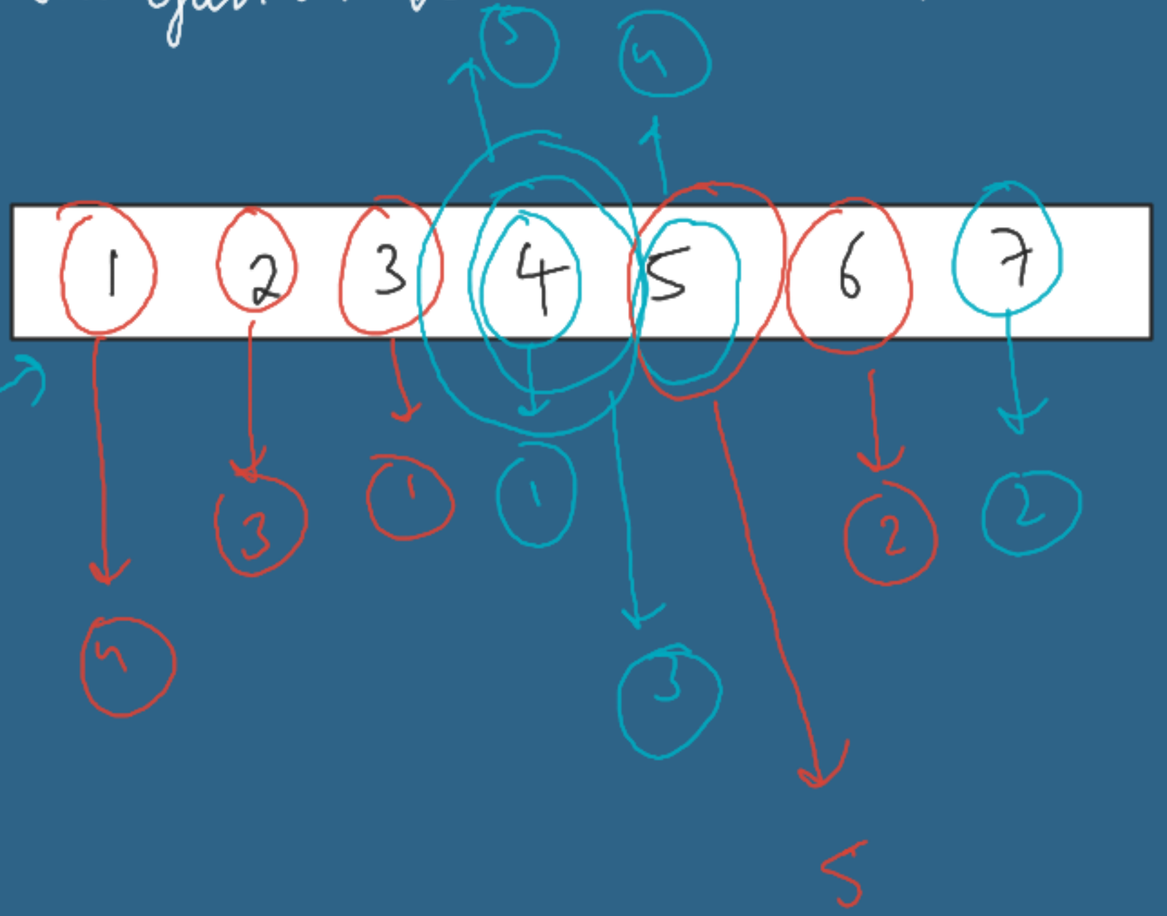
will return 1 after every call.

adjustment



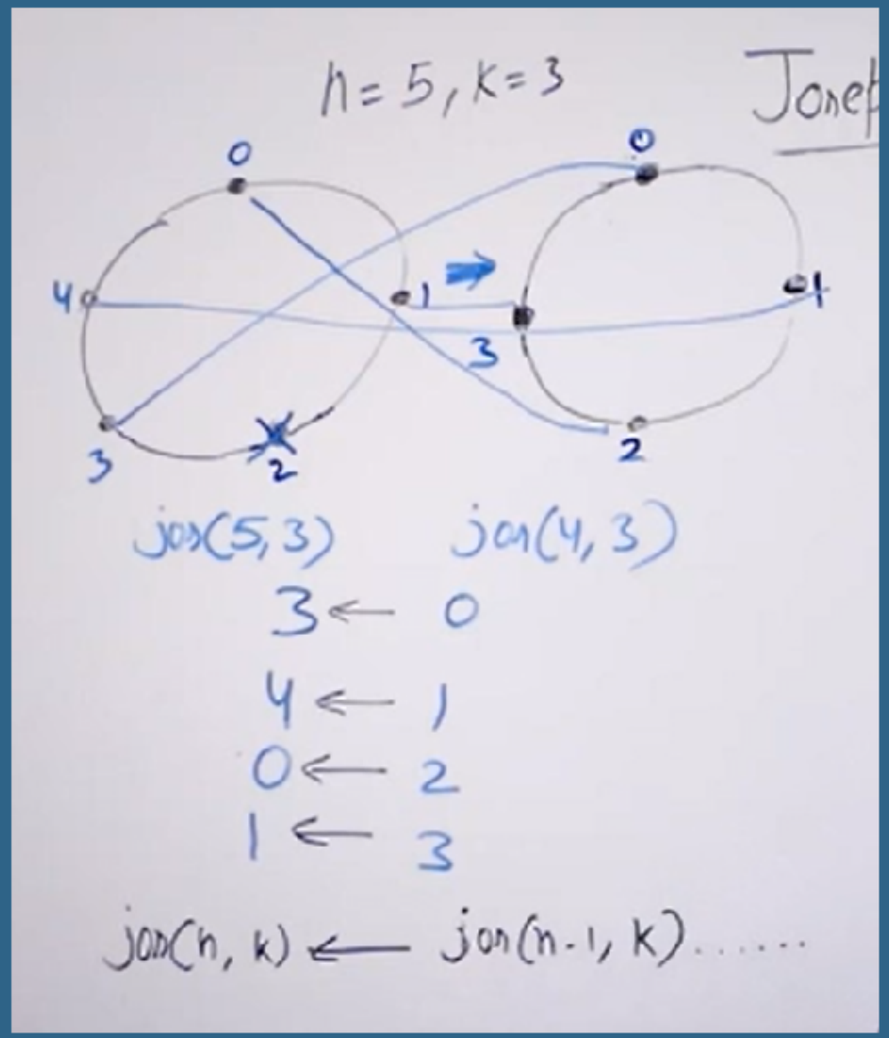
① Everytime when a person is killed, say n^{th}

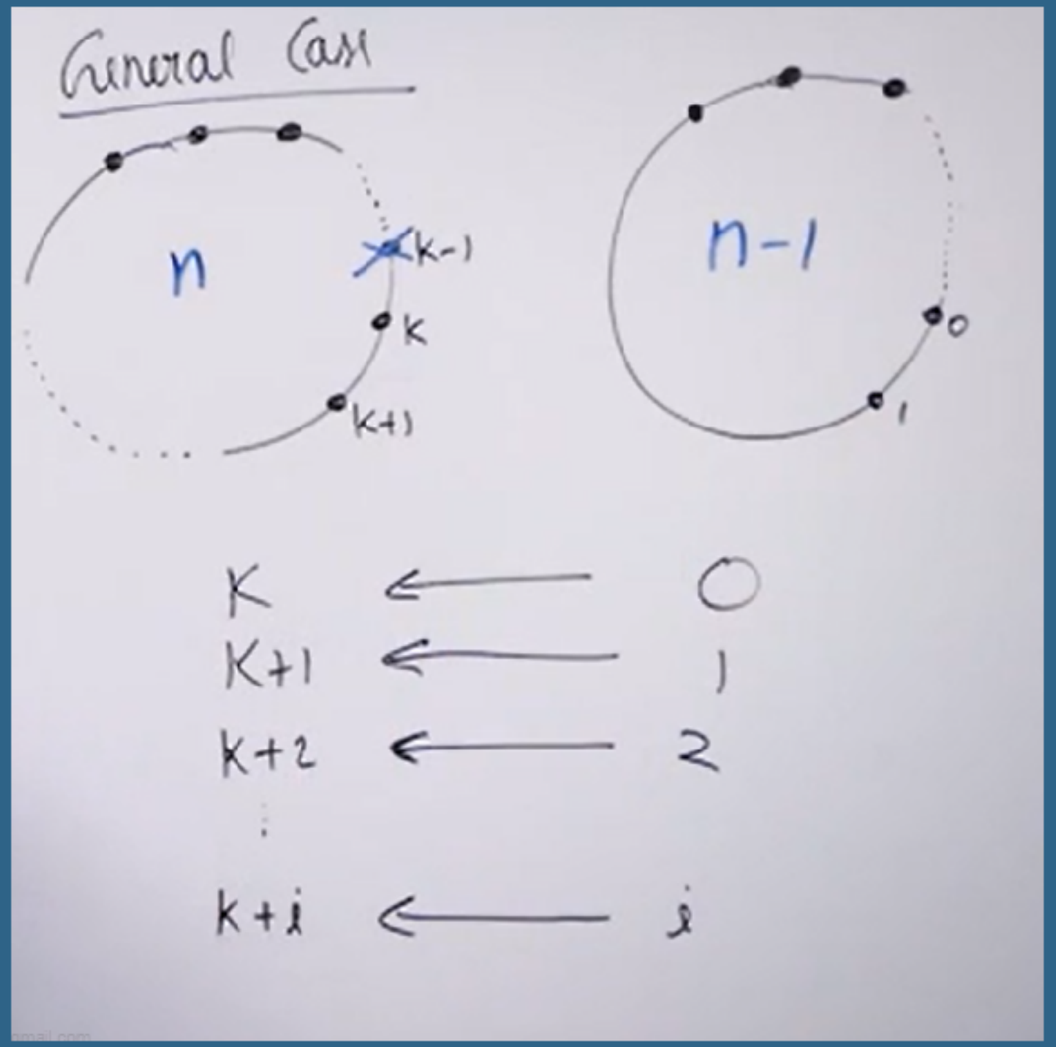
② The gun is passed to $n+1^{th}$ person.



Here we start with zero

Discerning the relationship b/w returned fn value and the original value:





→ Here we're solving our problem on a reduced problem

Connecting the solution of $(n-1)^{th}$ call and n^{th} :

① For 1' based index:

→ ⓐ when k^{th} person is killed → k+1 becomes the new '1'.

$k+2 \rightarrow 2$

```
int josephus(int n, int k)
{
    if (n == 1) //base case
       return 1;
    else
  /* The position returned by josephus(n - 1, k) is adjusted because the
       recursive call josephus(n - 1, k) considers the original position
       k%n + 1 as position 1 */
        return (josephus(n - 1, k) + k-1) % n + 1; //recursion
}
```

Coz we are working on '1' indexed scale.

→ O(P(n-1))

Coz the incoming function is 'k-1' and we need to adjust it.

```c
int josephus(int n, int k)
{
    if (n == 1) //base case
       return 1;
    else
  /* The position returned by josephus(n - 1, k) is adjusted because the
       recursive call josephus(n - 1, k) considers the original position
       k%n + 1 as position 1 */
        return (josephus(n - 1, k) + k-1) % n + 1; //recursion
}
```

Base case, for N = 1, the safest position.

Recursion steps:

(1) Assume we got the answer for (N-1) size problem

(2) Now we need to build upon it till Nth step. i.e, (N-1+1)

(3) Since we need to return, the next surviving location, i.e, the person with whom the pistol rests

(4) Introduce modulo so as we can handle the circular fashion of problem procedure