Josephus Problem

We solved the Josephus problem using a recursive function. We established a recursive function then called the function from the main function and sent “n” and “k” as perimeter.

In recursive function there we’ve checked whether the value of “n” is 1 or not. If n = 1 then the function will return 1. Else the recursive case will be executed.

Explaining the recursive case below:

$$f(n,k) \;=\; ((f(n-1,k)+k-1)\ \%\ n)\ +1$$

Suppose the quantity of people in the circle, $n=5$ and the step between two man in the circle, $k=2$.

$$step\ 1:\ \ f(5,2) \;=\; ((f(5-1,2)+2-1)\ \%\ 5)\ +1$$
$$\Rightarrow ((f(4,2)+1)\ \%\ 5)\ +1$$

Now the function will call it’s recursively and solve $f(4,2)$:

$$step\ 2:\ \ f(4,2) \;=\; ((f(4-1,2)+2-1)\ \%\ 4)\ +1$$
$$\Rightarrow ((f(3,2)+1)\ \%\ 4)\ +1$$

Now solve $f(3,2)$:

$$step\ 3:\ \ f(3,2) \;=\; ((f(3-1,2)+2-1)\ \%\ 3)\ +1$$
$$\Rightarrow ((f(2,2)+1)\ \%\ 3)\ +1$$

Now solve $f(2,2)$:

$$step\ 4:\ \ f(2,2) \;=\; ((f(2-1,2)+2-1)\ \%\ 2)\ +1$$
$$\Rightarrow ((f(1,2)+1)\ \%\ 2)\ +1$$

Here the recursive function could find $n = 1,$ after calling $f(1,2).$ As we set the statement if $n = 1$ then the function will return 1.

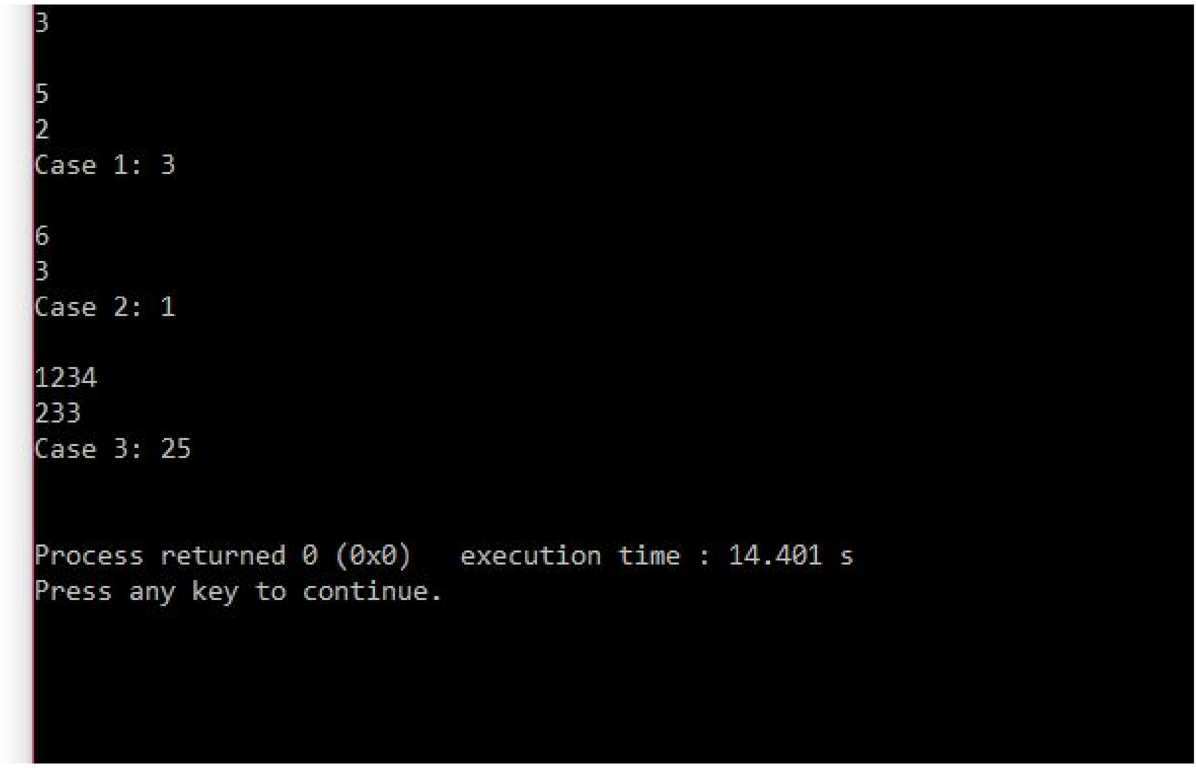
Then we have got the main result is:

$f(2,2) = 1$
$f(3,2) = 3$
$f(4,2) = 1$
$f(5,2) = 3 \ [Final\ result]$

CODE

```
#include<stdio.h>
int position(int n, int k)
{
        if (n == 1)
                return 1;
        else
                return (position(n - 1, k) + k - 1) % n + 1;
}

int main(void)
{
        int testcase,n,k;
        scanf("%d",&testcase);

        printf("\n");
        for(int i=1;i<=testcase;i++){

                scanf("%d%d",&n,&k);

                int josephus = position(n,k);
                printf("Case %d: %d",i,josephus);
                printf("\n\n");
        }

        return 0;
}
```

Output:



```
3

5
2
Case 1: 3

6
3
Case 2: 1

1234
233
Case 3: 25


Process returned 0 (0x0)   execution time : 14.401 s
Press any key to continue.
```