$1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7 \to 8$, $k = 3$.

↑
h

↑ ↑
P_2 r

↑
P_1

↳ $6 \to 7 \to 8 \to 1 \to 2 \to 3 \to 4 \to 5$

n = get length.

k = n % k.

```
const fn = ( head, k ) => {

    const n = getLength ( head );

    const r = n % k ;

    let p1 = head ;

    let p2 = head ;

    for ( let i = 0 ; i < r ; i++ ) {

        p2 = p2. next ;
    }

    while ( p2. next ) {

        p1 = p1. next ;

        p2 = p2. next ;
    }

    const newHead = p1. next ;
```

← hande edge cases

if (n <= 1 || r <= 0) {

    return head ;

}.

```
    p1.next = null;

    p2.next = head;

    return newHead;

}

function getLength(head) {
    let ptr = head;

    let n = 0;

    while(ptr) {
        ptr = ptr.next;

        n += 1;
    }

    return n;

}
```