

## Circular Linked List Solution

```
class Node {
  constructor(data, next = null) {
    this.data = data;
    this.next = next;
  }
}
// Initialise the linked List with clockwise length
let head = new Node(1)
let current = head

function startToEnd(start, end){
  for (let i = 2 ; i <= 12; i++){
    let newNode = new Node(i)
    current.next = newNode
    current = current.next
    if (i === 12){
      current.next = head
    }
  }

  current = head
  let clockwiseLength;
  let isStartFound = false
  while (current){
    if (current.data === start){
      if (!isStartFound){
        clockwiseLength = 1;
        isStartFound = true
      }
    }
    if (current.data === end){
      if (!isStartFound){
        clockwiseLength = 1;
      } else {
        clockwiseLength--
        break
      }
    }
    if (clockwiseLength){
      clockwiseLength++
    }
    current = current.next
  }
  const antiClockwiseLength = 12 - clockwiseLength;
  return Math.min(clockwiseLength, 12 - antiClockwiseLength)
}
```