Detect Loops in Linked List:

```
class Node {
    constructor(data){
        this.data = data;
        this.next = null;
    }
}
class Solution {

    detectLoop(head)
    {
        let tortoise = head
        let hare = head
        while(hare!==null && tortoise.next !== null){
            tortoise = tortoise.next
            hare = hare.next.next
            if(tortoise === hare){
                return true
            }
        }
        return false
    }

}

let head = new Node(1);
let second = new Node(2);
let third = new Node(3);
let fourth = new Node(4);

head.next = second;
second.next = third;
third.next = fourth;
fourth.next = second; //toggle

let solution = new Solution();
console.log(solution.detectLoop(head));
```

Cyclically Rotate an Array:

```
class Solution{
    rotate(arr,n){
        let element = arr.pop()
        arr.unshift(element)
        return arr
    }
}
```

```
let TestSolution = new Solution()
console.log(TestSolution.rotate([1, 2, 3],3))
```

Reversing An Array:

```
class ReadlineConsole {
  constructor() {
    this.numbers = [];
    this.readline = require("readline").createInterface({
      input: process.stdin,
      output: process.stdout,
    });
  }

  async getNumbers() {
    const ask = async (question) => {
      return new Promise((resolve) => {
        this.readline.question(question, resolve);
      });
    };

    let input = await ask(
      "Enter required number of integers separated by spaces and then press enter: "
    );
    input = input.trim().replace(/\s+/g, " "); // Remove leading/trailing spaces and
consecutive spaces

    let numbersArray = input.split(" ");

    for (let i = 0; i < numbersArray.length; i++) {
      let number = parseInt(numbersArray[i]);
      if (!isNaN(number)) {
        this.numbers.push(number);
      }
    }

    this.readline.close();
  }

  async showNumbers() {
    for (let i = 0; i < this.numbers.length; i++) {
      console.log(this.numbers[i]);
    }
  }

  async reverseArray() {//O(n)
    let arr = this.numbers;

    let arr2 = [];
    for (let i = arr.length - 1; i >= 0; i--) {
      arr2.push(arr[i]);
    }
```

```
    console.log("Reversed =====>", arr2);
  }
  async reverseArrayOnby2() {//O(n/2)
    let arr = this.numbers;
    let start = 0
    let end = arr.length-1
    while (start<end) {
        var temp = arr[start];
        arr[start] =  arr[end]
        arr[end] = temp
        start++;
        end--;
    }
    console.log("Reversed =====>", arr);
  }
}

(async () => {
  const readConsole = new ReadlineConsole();
  await readConsole.getNumbers();
  readConsole.showNumbers();
  //readConsole.reverseArray();
  readConsole.reverseArrayOnby2();
})();
```