class Node {

constructor(data) {

this.data = data;

this.next = null;

}

}

class Queue {

constructor() {

this.front= null;

this.rear = null;

this.length = 0;

}

enqueue(data) {

const node = new Node(data); // creates the node using class Node

if (this.front) { // if the first Node exitsts

this.rear.next = node; // inserts the created node after the tail of our Queue

this.rear = node; // now the created node is the last node

} else { // if there are no nodes in the Queue

this.front = node; // the created node is a head

this.rear = node // also the created node is a tail in Queue because it is single.

}

this.length++; // increases the length of Queue by 1

}

dequeue() {

const temp = this.front;

if(this.length===0){

console.log('queue is empty');

}else{// saves the link to the head which we need to remove

this.front = this.front.next; // moves the head link to the second Node in the Queue

this.length--; // decreaments the length of our Queue

return temp.data; // returns the removed Node's value

}

}

print() {

let temp = this.front; // saves a link to the head of the queue

while(temp) { // goes through each Node of the Queue

console.log(temp.data); // prints the value of the Node in console

temp = temp.next; // moves link to the next node after head

}

}

}

const qu=new Queue();

qu.enqueue(100);

qu.enqueue(200);

qu.enqueue(300);

qu.print();

qu.dequeue();

qu.print();

qu.dequeue();

qu.print();

qu.dequeue();

qu.print();

qu.dequeue();

qu.print();