/\*\*

\* Sum all numbers till the given one

\*/

function sum(n) {

if (n <= 1) {

return 1;

}

return n + sum(n - 1);

}

console.log(sum(6))

/\*\*

\* use recursion to implement print() method

\*/

function print(item){

if(item.length === 0){

return "";

}

return item.charAt(0) + item.substring(1);

}

let s = print("Awaab");

console.log(s);

"use strict";

/\*\*

\* Use Object literal to create an object named linkedlist which has the

\* following methods:

\* linkedlist.add(1)

\* linkedlist.remove(2) – remove the first one

\* linkedlist.print() – format: LinkedList{ 1, 2, 5, 7 } \*

\*

\*/

let linkedlist = {

head: {},

add: (item) => {

let element = { value: item, next: null };

element.next = this.head;

this.head = element;

},

remove: (item) => {

if (item === undefined) {

this.head = this.head.next;

} else {

if (this.head.value === item) {

this.head = this.head.next;

} else {

let current = this.head;

while (current.next != undefined) {

if (current.next.value === item) {

current.next = current.next.next;

} else {

current = current.next;

}

}

}

}

},

print: () => {

let result = [];

while (this.head != null) {

result.push(this.head.value);

this.head = this.head.next;

}

// return Object.values(this.head);

return "LinkedList{" + result.reverse().join() + "}";

}

};

linkedlist.add(1);

linkedlist.add(2);

// linkedlist.remove(2);

linkedlist.add(3);

// linkedlist.remove(3);

linkedlist.add(6);

console.log(linkedlist.print());