Data Structure and Algorithm Practicals

2. STACK implementation using Array with PUSH, POP operations

```
let stack = [];

stack.push(1);
console.log(stack); // [1]

stack.push(2);
console.log(stack); // [1,2]

stack.push(3);
console.log(stack); // [1,2,3]

stack.push(4);
console.log(stack); // [1,2,3,4]

stack.push(5);
console.log(stack); // [1,2,3,4,5]

console.log(stack.pop()); //  5
console.log(stack); // [1,2,3,4];

console.log(stack.pop()); //  4
console.log(stack); // [1,2,3];

console.log(stack.pop()); //  3
console.log(stack); // [1,2];

console.log(stack.pop()); //  2
console.log(stack); // [1];

console.log(stack.pop()); //  1
console.log(stack); // []; -> empty

console.log(stack.pop()); //  undefined
```