a. Creation phase

LE: { makeArmy: fun(), outer: null}, thisBinding: window	army

Execution phase

LE: { makeArmy: fun(), outer: null ,army:[fun(), fun()]} thisBinding: window makeArmy()	

b. LE for makeArmy()

Creation phase

LE: { makeArmy:Fn, outer: global} Arguments: {length:0}	shooters

Execution phase

LE: { makeArmy:Fn, outer: global} Arguments: {length:0}	

b. Creation phase

LE: { outer: makeArmy }	shooter

Execution phase

LE: { outer: makeArmy, shooter: fun()}	

d. LE for army[0]

Creation phase

```
Closure
i = 2 , outer: makeArmy

army[0]()

LE: { arguments: { length: 0 }, outer: closure}
```

Execution phase

```
Closure
i = 2 , outer: makeArmy

army[0]()

LE: { arguments: { length: 0 }, outer: closure}
alert(i); i=2;
```

e. What will army[0] alert?

2

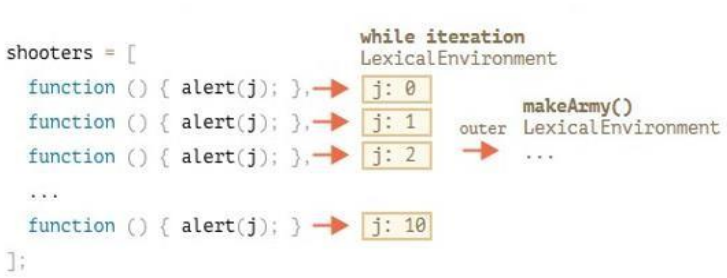
f. Can you fix the code?

```
function makeArmy() {
  let shooters = []
  let i = 0
  while (i < 2) {
   let j = i;
    let shooter = function(){
      alert(j)
    }
    shooters.push(shooter)
    i++
  }
  return shooters
}

let army = makeArmy()
```

army.forEach(f => f())

g. How will the diagram change?



1. Write a function printNumbers(*from*,*to*) that outputs a number every second, starting from *from* and ending with *to.*

```
function printNumbers(from, to) {
    let x = setInterval(() => {
        if(from === to){
            clearTimeout(x)
        }
        console.log(from++)
    }, 1000)
}

// printNumbers(5, 10)
```

2. When will the schedules function run?
   - After the loop
   - will print 100000000