1. Global Execution Context Lexical environment
   a) Creation phase

LE: {makeArmy:fn, outer:null} thisBinding:window	TDZ army

   b) Execution phase

LE: {makeArmy:function, outer:null,army:[fn,fn]} thisBinding:window

2. **MakeArmy** Function Execution Context – Lexical Environment
   a. Creation

	TDZ
LE:{outer:global}, thisBinding:window	Shooters i

   b. Execution

LE:{outer:global,shooters:[fn,fn],i:0}, thisBinding:window

3. While loop Execution Context
   a. Creation phase

	TDZ
LE:{outer:makeArmy,anonymous:fn}, thisBinding:window	Shooter

   NB: There are three iteration with the same lexical execution.

   b. Execution

   The first loop execution context

LE:{outer:makeArmy,anonymous:fn,shooter: fn}, thisBinding:window

   NB: The anonymous function has to lexical execution environment.
4. Lexical execution for **army[0]**

a. Creation phase

b. Execution

5. What will army[0] alert? army[0] alert is not function call but is function. To make it function call we should correct it army[0](). And basically after making it function the method will always alert **i result after increment**.
6. Here below is the correction.

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

7. Changes in diagram after correction is all about having local variable **j** in addition.

Question 2:printNumber function that can print every number as scheduled.

```
function printNumbers(from,to){
setInterval(()=>{
    if(from<=to){
        console.log(from);
        from++;
    }
},1000)};
printNumbers(5,7);
```

Question 3: Explanation about setTimeout scheduled function.

```
let i=0;
setTimeout(()=>alert(i),100);
for(let j=0;j<100000000;j++){
    i++;
}
```

The scheduled function run after the loop. Because setTimeout simply queues the code to run once the current call stack is finished executing. Once done, the setTimeout will execute the alert which will be popout window with the value of i (i.e. 100000000).