1. GLOBAL LEXICAL ENVIRONMENT

globalEC : creation → execution	army
LE : { outer : null, makeArmy : fn, army : makeArmy [fn, fn] }	

2. LE FOR MAKE ARMY

TDZ

makeArmyEC : creation → execution	shooters
LE : { arguments : {length : 0}, outer : global, shooters : ~~[]~~ ~~[fn(){alert(i)}]~~, [fn(){alert(i)}, fn(){alert(i)}], i : ~~0~~ ~~1~~ 2 }	i

3. LE FOR LE OF THE WHILE LOOP

TDZ

while (i=0) EC : creation → execution	shooter

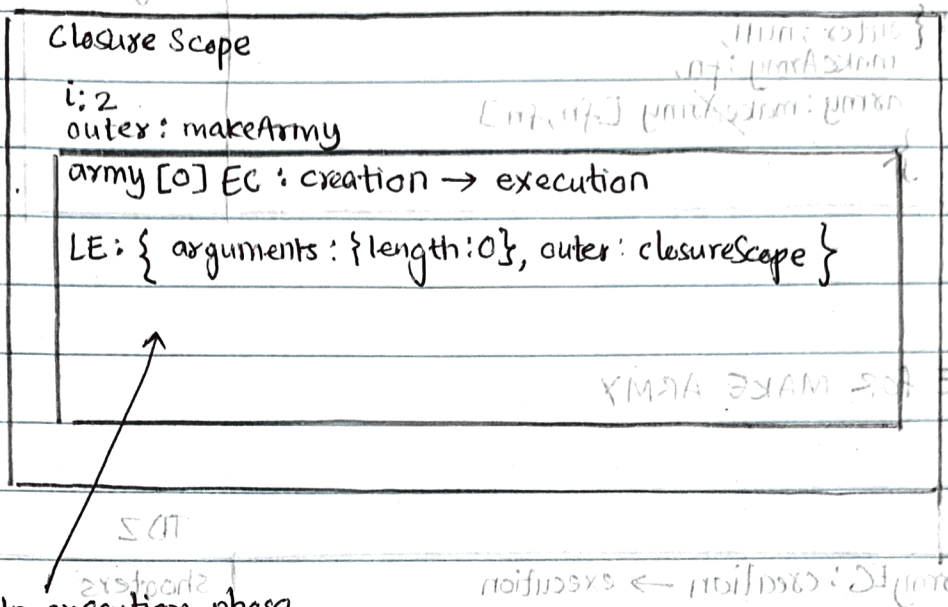
LE : {
outer : makeArmy(),
shooter : fn(){alert(i)},
}	

TDZ

while (i=1) EC : creation	shooter

LE : {
outer : makeArmy(),
shooter : fn(){alert(i)}
}	

4. LE for ~~army~~ army[0]



In execution phase

army[0] ⟶ fn(){alert(i)} is taken from army that returns an array. Here free variable 'i' is available through closure scope, which means i = 2.

5. WHAT WILL army[0] alert?

→ Doing 'army[0]()' → this will alert 2.

6. CAN YOU FIX THE CODE

→ DONE IN GITHUB'S Q1 folder's 'MakeArray.js' file

7. ~~AFTE~~ HOW WILL THE DIAGRAM CHANGE?

AFTER THE FIX, ~~IN~~ IN THE DIAGRAM, THE GLOBAL EXECUTION WILL ALSO BE POPPED OUT ONCE THE 'forEach' FUNCTION IS DONE EXECUTING.