1. Explain what is prototype and what is prototype chain in your own words

Prototype:

A prototype is an object from which other objects inherits properties and methods. When an object is created it inherits properties and methods from it’s prototype. This allows sharing of functionalities among objects without duplicating the code.

Prototype Chain: Prototype Chain is a mechanism by which objects inherits properties and methods of other objects. It is formed by linking an object with another object via the prototype property, creating a chain of linked objects. When a property or method is accessed on an object, the JavaScript engine first looks at the properties and methods of the object, If the property or method is not found, the engine looks for it in the prototype chain until it either finds the desired property or method or reaches the end of the chain.

MAP:

1. Array.prototype.customMap = function(callback){
2. res = []
3. for (let i = 0; i < this.length; i++) {
4. res.push(callback(this[i], this))
5. console.log(res)
6. }
7. return res
9. }
10. nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
11. let mappedArr = nums.customMap((num)=> num \* 2)
12. console.log(mappedArr) // [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

FILTER:

Array.prototype.customFilter = function(callback){

  res = []

  for (let i = 0; i < this.length; i++) {

    if(callback(this[i])){

      res.push(this[i])

    }

  }

  return res

}

nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

let filteredArr2 = nums.customFilter((num)=> num % 2)

console.log(filteredArr2) // [1, 3, 5, 7, 9]

REDUCE:

Array.prototype.customReduce = function(callback, initialValue){

  let accumulator = initialValue

  for(let i = 0; i < this.length; i++){

    accumulator = callback(accumulator, this[i])

  }

  return accumulator

}

nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

let reduceArr2 = nums.customReduce((accumulator, element)=> accumulator + element, 0)

console.log(reduceArr2) // 55

PUSH:

Array.prototype.customPush = function(element){

  this[this.length] = element

  return this.length

}

let arr = [1, 2, 3, 4, 5]

console.log(arr.customPush(10)) // 6

console.log(arr) // [1, 2, 3, 4, 5, 10]

POP:

Array.prototype.customPop = function(){

    if(this.length == 0){

        return undefined

    }

    lastElement = this[this.length - 1]

    this.length = this.length - 1

    return lastElement

}

let nums = [1,2,3,4,5]

let popped = nums.customPop()

console.log(popped) // 5

console.log(nums) //[1,2,3,4]

REVERSE:

Object.prototype.customReverse = function(){

    let reversed = []

    for(i = this.length - 1; i >= 0; i--){

        reversed.push(this[i])

    }

    return reversed

}

let nums = [1,2,3,4,5]

let reverse = nums.customReverse()

console.log(reverse) // [5,4,3,2,1]