1. Do the below programs in anonymous function & IIFE

a. Print odd numbers in an array.

Code:

```
const readline =
require('readline'); const inp =
readline.createInterface({ input:
process.stdin
});
const userInput = [];
inp.on("line", (data)
=> {
userInput.push(data);
});
inp.on("close", () => {

let numbers=[2,3,4,5,6,7,8,9];

// IIFE creation and execution
(function() {
   for (i = 0; i < numbers.length;
     i++) {
    if (numbers[i] % 2 !== 0) {
       console.log(numbers[i]);
```

```
    }}} )();
});
```

Output:

```
3
5
7
9
```

Execution Time:

0.075s

Memory Used:

8408kb

b. Convert all the strings to title caps in a string array

Code:

```
const readline =
require('readline'); const inp =
readline.createInterface({ input:
process.stdin
});
const userInput = [];
inp.on("line", (data)
=> {
userInput.push(data);
});
inp.on("close", () => {

let str =
["akhila","alekhya","akshaya"];

// IIFE creation and execution
(function () {
  for (var i = 0; i < str.length; i++)
{
   str[i] =
str[i].charAt(0).toUpperCase() +
```

```
    str[i].slice(1);

      }

      return str.join(' ');

    })();

     console.log(str);

    });
```

Output:

['Akhila', 'Alekhya', 'Akshaya']

Execution Time:

0.074s

Memory Used:

8324kb

C. Sum of all numbers in an array

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
userInput.push(data);
});
inp.on("close", () => {



let numbers = [10, 20, 30, 40, 50];
let sum = 0;

(function (){
for (let i = 0; i < numbers.length; i++) {
  sum += numbers[i];
}
  console.log(sum);
})();

});
```

Output:

```
150
Execution Time:
0.073s
Memory Used:
8408kb
```

d. Return all the prime numbers in an array

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
userInput.push(data);
});
inp.on("close", () => {



var numArray = [2, 3, 4, 5, 6, 7, 8, 9, 10];

numArray = numArray.filter((number) => {
  for (var i = 2; i <= Math.sqrt(number); i++) {
    if (number % i === 0) return false;
  }
  return true;
});

console.log(numArray);

});
```

Output:

```
[ 2, 3, 5, 7 ]
Execution Time:
0.074s
Memory Used:
8324kb
```

e. Return all the palindromes in an array

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
userInput.push(data);
});
inp.on("close", () => {


var words = ['dad', 'racecar', 'mom', 'sis', 'bro'];

(function (){
var arr = [];
var str = words.slice(0);
var pal = str.toString().split("").reverse().join("").split(",");


for (let i = 0; i < words.length; i++) {
  for (let k = 0; k < pal.length; k++) {
    if (words[i] == pal[k]) {
      arr.push(words[i])
    }
  }
}


console.log(arr);
})();


});
```

Output:

['dad', 'racecar', 'mom', 'sis']
Execution Time:
0.075s
Memory Used:
8288kb

f. Return median of two sorted arrays of same size

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
userInput.push(data);
});
inp.on("close", () => {

let ar1 = [ 1, 2, 3, 6 ];
let ar2 = [ 4, 6, 8, 10 ];
   let n1 = ar1.length;
   let n2 = ar2.length;
   let n=n1;
(function ()
   {
     let j = 0;
     let i = n - 1;
```

```
    while (ar1[i] > ar2[j] && j < n && i > -1)
    {
      let temp = ar1[i];
      ar1[i] = ar2[j];
      ar2[j] = temp;
      i--; j++;
    }
    ar1.sort(function(a, b){return a - b});
    ar2.sort(function(a, b){return a - b});
    console.log( “Median is”+parseInt((ar1[n - 1] + ar2[0]) / 2, 10));
  })();
});
```

Output:

Median is5
Execution Time:
0.072s
Memory Used:
8320kb

g. Remove duplicates from an array

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
userInput.push(data);
});
inp.on("close", () => {


(function (){
   const arr = [1, 2, 3, 2, 3, 4, 5];

   let uniqueArr = [...new Set(arr)];

   console.log(uniqueArr);
})();

});
```

Output:

[1, 2, 3, 4, 5]
Execution Time:
0.073s
Memory Used:
8420kb

h. Rotate an array by k times

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
userInput.push(data);
});
inp.on("close", () => {

let nums=[1, 2, 3, 4, 5, 6];
let k=3;

(function(){
   let i=0;
   while (i<k) {
      nums.unshift(nums.pop())
      i++
   }
   return nums
})();

   console.log(nums);

});
```

Output:

```
[ 4, 5, 6, 1, 2, 3 ]
Execution Time:
0.074s
Memory Used:
8344kb
```

3. Do the below programs in arrow functions

a. **Print odd numbers in an array**

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
userInput.push(data);
});
inp.on("close", () => {

let numbers=[1, 2, 3, 4, 5, 6];
 let odd=  numbers.filter(number => number % 2);
 console.log(odd);

});
```

Output:

```
[ 1, 3, 5 ]
Execution Time:
0.074s
Memory Used:
8412kb
```

b.Convert all the strings to title caps in a string array

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
userInput.push(data);
});
inp.on("close", () => {

var arr = ["akhila", "alekhya", "zen", "fullstack"];
var str = () => {
 for (var i = 0; i <= arr.length; i++) {
 console.log(arr[i][0].toUpperCase() + arr[i].substr(1));
 }
}
str();

});
```

Output:

```
Akhila
Alekhya
Zen
Fullstack
Execution Time:
0.071s
Memory Used:
8332kb
```

C. Sum of all numbers in an array

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
userInput.push(data);
});
inp.on("close", () => {


let numbers = [10, 20, 30, 40, 50];
let sum = 0;

 let sum1 = () =>{
for (let i = 0; i < numbers.length; i++)
  sum += numbers[i];
  console.log(sum);
}
sum1();

});
```

Output:

150
Execution Time:
0.073s
Memory Used:
8388kb

d. Return all the prime numbers in an array

Code:

```
const readline = require('readline');
const inp = readline.createInterface({
input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
userInput.push(data);
});
inp.on("close", () => {


var numArray = [2, 3, 4, 5, 6, 7, 8, 9, 10];

numArray = numArray.filter((number) => {
  for (var i = 2; i <= Math.sqrt(number); i++) {
    if (number % i === 0) return false;
  }
  return true;
});

console.log(numArray);

});
```

Output:

```
[ 2, 3, 5, 7 ]
Execution Time:
0.073s
Memory Used:
8404kb
```

e. Return all the palindromes in an array

Code :

```
const readline = require('readline');
const inp = readline.createInterface({
input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
userInput.push(data);
});
inp.on("close", () => {


var words = ['dad', 'racecar', 'mom', 'sis', 'bro'];

let func = () =>{
var arr = [];
var str = words.slice(0);
var pal = str.toString().split("").reverse().join("").split(",");


for (let i = 0; i < words.length; i++) {
  for (let k = 0; k < pal.length; k++) {
    if (words[i] == pal[k]) {
      arr.push(words[i])
    }
  }
}


console.log(arr);
}
func();

});
```

Output:

['dad', 'racecar', 'mom', 'sis']
Execution Time:
0.072s
Memory Used:
8292kb