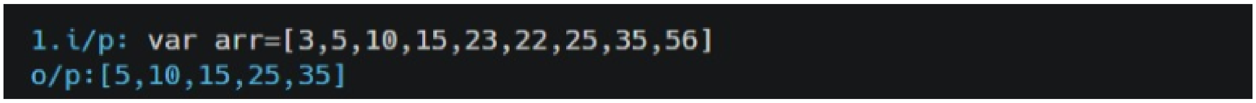
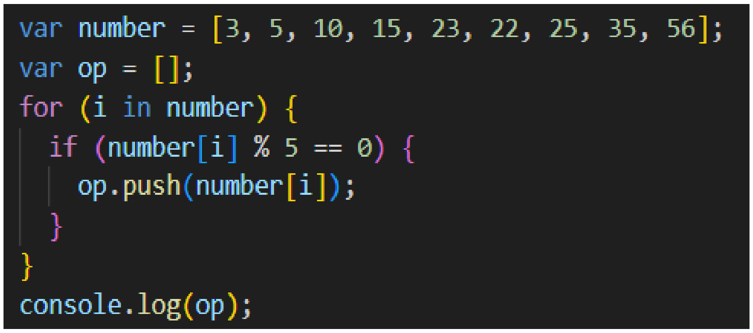
TASK-8

1.



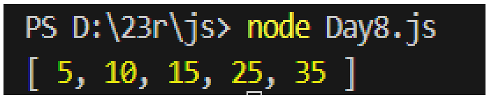
```
1.i/p: var arr=[3,5,10,15,23,22,25,35,56]
o/p:[5,10,15,25,35]
```

Code:



```
var number = [3, 5, 10, 15, 23, 22, 25, 35, 56];
var op = [];
for (i in number) {
  if (number[i] % 5 == 0) {
    op.push(number[i]);
  }
}
console.log(op);
```

Output:



```
PS D:\23r\js> node Day8.js
[ 5, 10, 15, 25, 35 ]
```

Explanation:

This JavaScript code filters numbers divisible by 5 from the array number and stores them in a new array op. Here's how it works step-by-step:

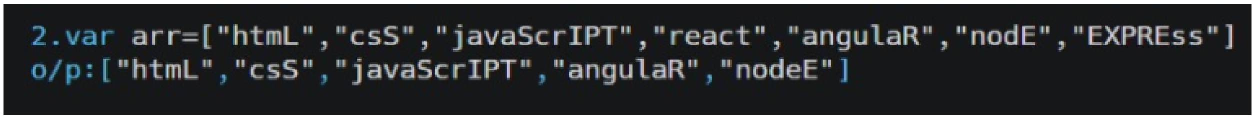
1. The number array contains the values [3, 5, 10, 15, 23, 22, 25, 35, 56].
2. An empty array op is initialized to store numbers that are divisible by 5.
3. The for...in loop iterates over each index i in the number array.

Inside the loop, each value in the number array is checked to see if it's divisible by 5 (number[i] % 5 == 0).

If a value satisfies this condition, it is added to the op array using op.push(number[i]);.

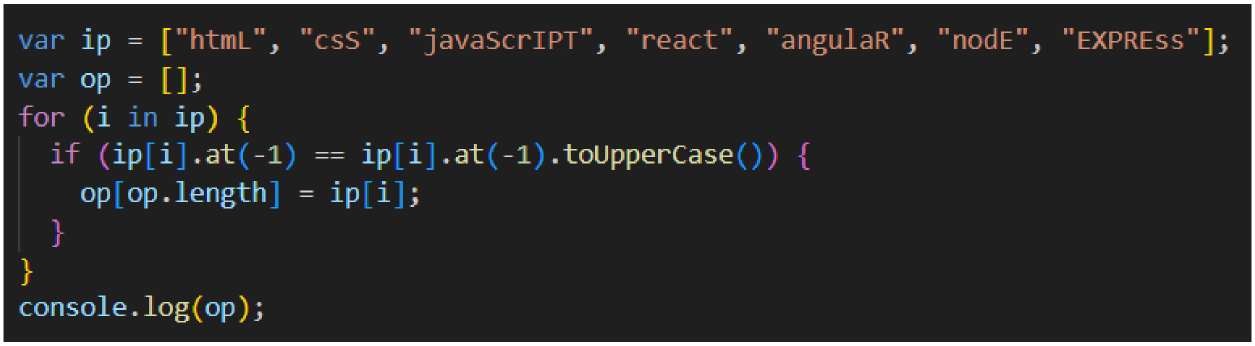
4. After the loop finishes, op contains all numbers from number that are divisible by 5.
5. console.log(op); outputs the contents of op.

2.



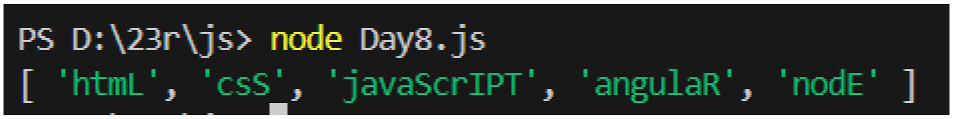
```
2.var arr=["htmL","csS","javaScrIPT","react","angulaR","nodE","EXPREss"]
o/p:["htmL","csS","javaScrIPT","angulaR","nodeE"]
```

Code:



```
var ip = ["htmL", "csS", "javaScrIPT", "react", "angulaR", "nodE", "EXPREss"];
var op = [];
for (i in ip) {
  if (ip[i].at(-1) == ip[i].at(-1).toUpperCase()) {
    op[op.length] = ip[i];
  }
}
console.log(op);
```

Output:

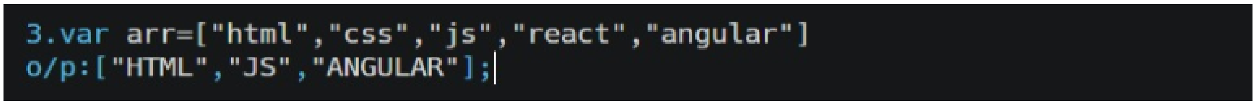


```
PS D:\23r\js> node Day8.js
[ 'htmL', 'csS', 'javaScrIPT', 'angulaR', 'nodE' ]
```

Explanation:

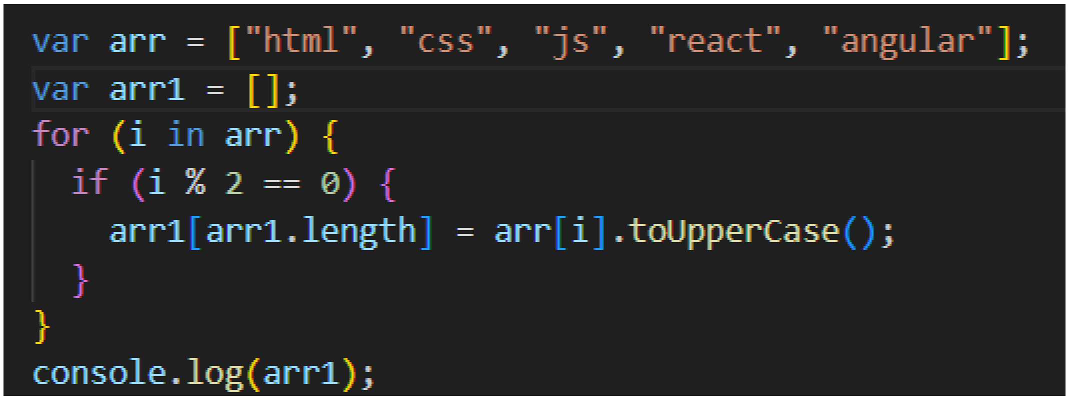
The code iterates through the ip array and checks if the last character of each string is uppercase. If it is, the string is added to the op array. Here's how it works step-by-step:
1. Input Array (ip): ["html", "css", "javaScrIPT", "react", "angular", "node", "EXPRESS"]
2. Initialization: An empty array op is created to store the selected strings.
3. Loop Execution:
"html": Last character 'l' is lowercase – not added to op.
"css": Last character 's' is lowercase – not added to op.
"javaScrIPT": Last character 'T' is uppercase – added to op.
"react": Last character 't' is lowercase – not added to op.
"angular": Last character 'r' is lowercase – not added to op.
"node": Last character 'e' is lowercase – not added to op.
"EXPRESS": Last character 'S' is uppercase – added to op.
4. Output Array (op):
After the loop, op contains only the strings where the last character is uppercase:
["javaScrIPT", "EXPRESS"].

3.



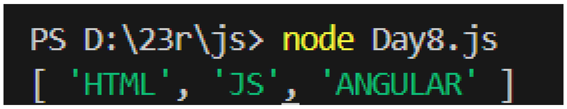
```
3.var arr=["html","css","js","react","angular"]
o/p:["HTML","JS","ANGULAR"];
```

Code:



```
var arr = ["html", "css", "js", "react", "angular"];
var arr1 = [];
for (i in arr) {
  if (i % 2 == 0) {
    arr1[arr1.length] = arr[i].toUpperCase();
  }
}
console.log(arr1);
```

Output:



```
PS D:\23r\js> node Day8.js
[ 'HTML', 'JS', 'ANGULAR' ]
```

Explanation:

The code iterates through the arr array and selects elements at even indices (0, 2, 4, ...) to convert to uppercase. These uppercase strings are then added to the arr1 array.
Here’s a step-by-step breakdown:
1. Input Array (arr): ["html", "css", "js", "react", "angular"]
2. Initialization: An empty array arr1 is created to store the transformed strings.
3. Loop Execution:
Index 0 ("html"): 0 % 2 == 0 (even index), so "html" is converted to "HTML" and added to arr1.
Index 1 ("css"): 1 % 2 != 0 (odd index), so it's skipped.
Index 2 ("js"): 2 % 2 == 0 (even index), so "js" is converted to "JS" and added to arr1.
Index 3 ("react"): 3 % 2 != 0 (odd index), so it's skipped.
Index 4 ("angular"): 4 % 2 == 0 (even index), so "angular" is converted to "ANGULAR" and added to arr1.
4. Output Array (arr1):
After the loop, arr1 contains the uppercase versions of the elements at even indices: ["HTML", "JS", "ANGULAR"].