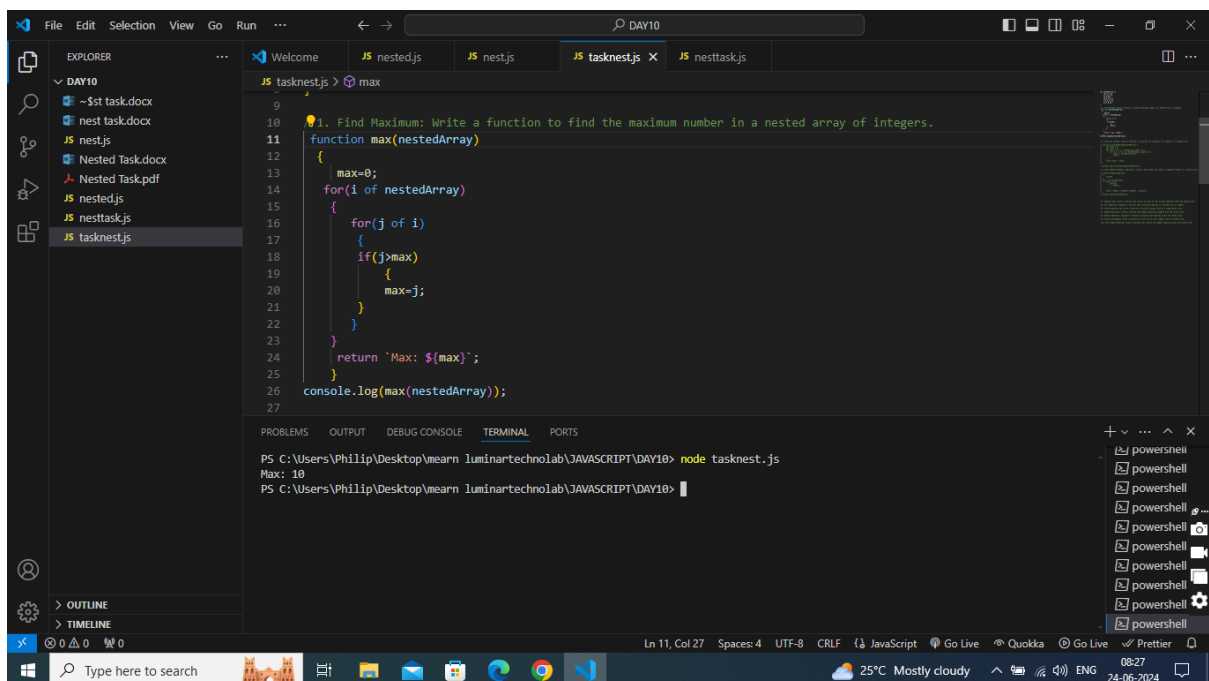


TASK2

```
var nestedArray =[
    [5,6,7,-2],
    [-5,-6,-7],
    [8,9,10],
    [3,5,2,1,4],
    [-3,5,2,1,],
    [4,2,6,8],
]
```

//1. Find Maximum: Write a function to find the maximum number in a nested array of integers.



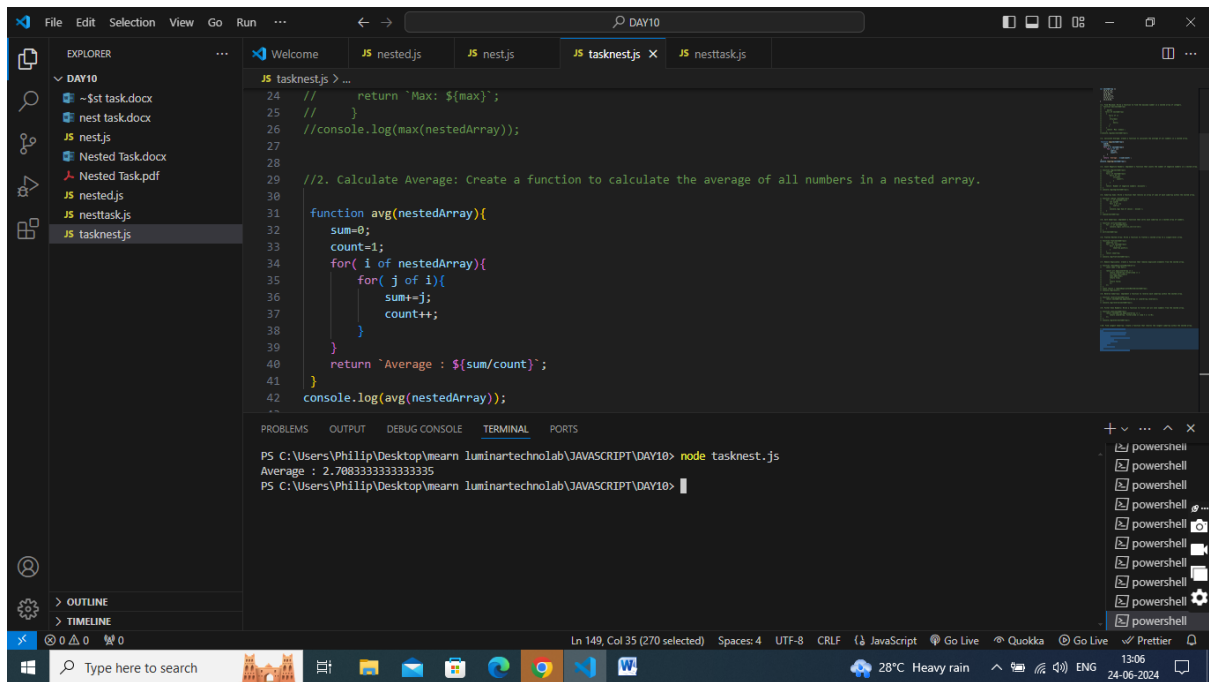
The screenshot shows the Visual Studio Code editor with a file named `tasknest.js` open. The code defines a function `max` that iterates through a nested array to find the maximum value. The terminal output shows the function being called with the nested array, resulting in `Max: 10`.

```
9
10
11 //1. Find Maximum: Write a function to find the maximum number in a nested array of integers.
12 function max(nestedArray)
13 {
14     max=0;
15     for(i of nestedArray)
16     {
17         for(j of i)
18         {
19             if(j>max)
20             {
21                 max=j;
22             }
23         }
24     }
25     return `Max: ${max}`;
26     console.log(max(nestedArray));
27 }
```

Terminal Output:

```
PS C:\Users\Philip\Desktop\learn_luminartechnolab\JAVASCRIPT\DAY10> node tasknest.js
Max: 10
PS C:\Users\Philip\Desktop\learn_luminartechnolab\JAVASCRIPT\DAY10>
```

//2. Calculate Average: Create a function to calculate the average of all numbers in a nested array.

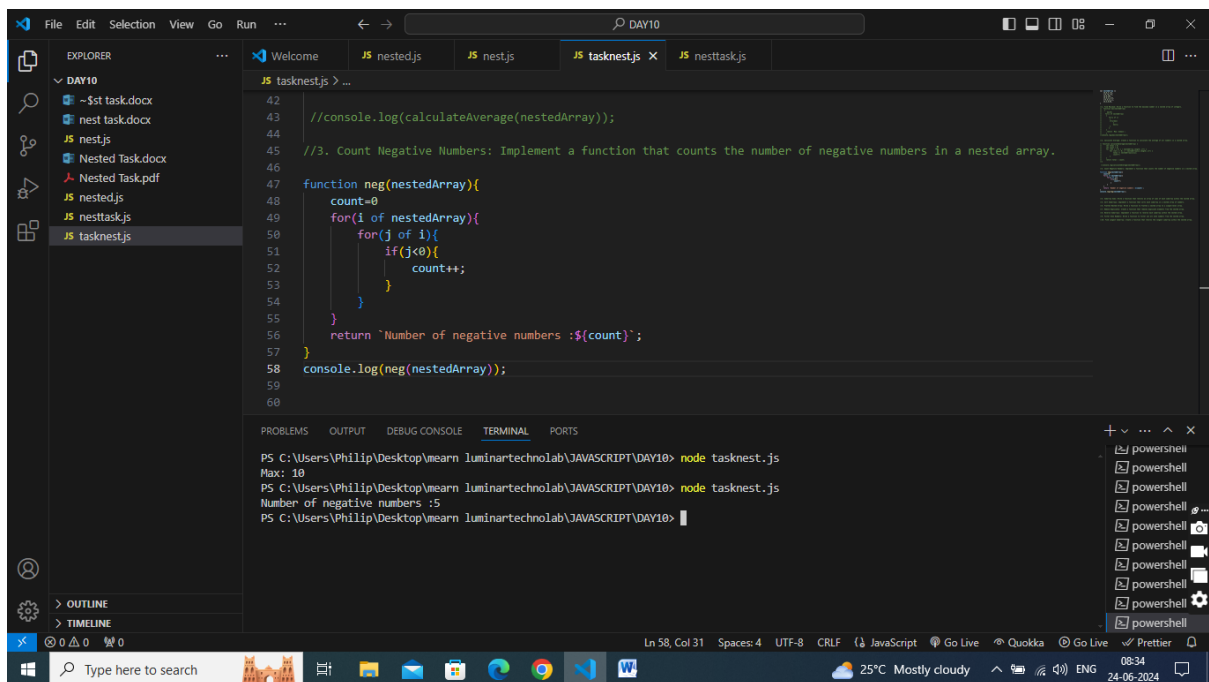


```
24 // return `Max: ${max}`;
25 // }
26 //console.log(max(nestedArray));
27
28
29 //2. Calculate Average: Create a function to calculate the average of all numbers in a nested array.
30
31 function avg(nestedArray){
32     sum=0;
33     count=1;
34     for( i of nestedArray){
35         for( j of i){
36             sum+=j;
37             count++;
38         }
39     }
40     return `Average : ${sum/count}`;
41 }
42 console.log(avg(nestedArray));
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Philip\Desktop\learn_luminartechnolab\JAVASCRIPT\DAY10> node tasknest.js
Average : 2.7083333333333335
PS C:\Users\Philip\Desktop\learn_luminartechnolab\JAVASCRIPT\DAY10>

//3. Count Negative Numbers: Implement a function that counts the number of negative numbers in a nested array.



```
42 //console.log(calculateAverage(nestedArray));
43
44
45 //3. Count Negative Numbers: Implement a function that counts the number of negative numbers in a nested array.
46
47 function neg(nestedArray){
48     count=0
49     for(i of nestedArray){
50         for(j of i){
51             if(j<0){
52                 count++;
53             }
54         }
55     }
56     return `Number of negative numbers :${count}`;
57 }
58 console.log(neg(nestedArray));
59
60
```

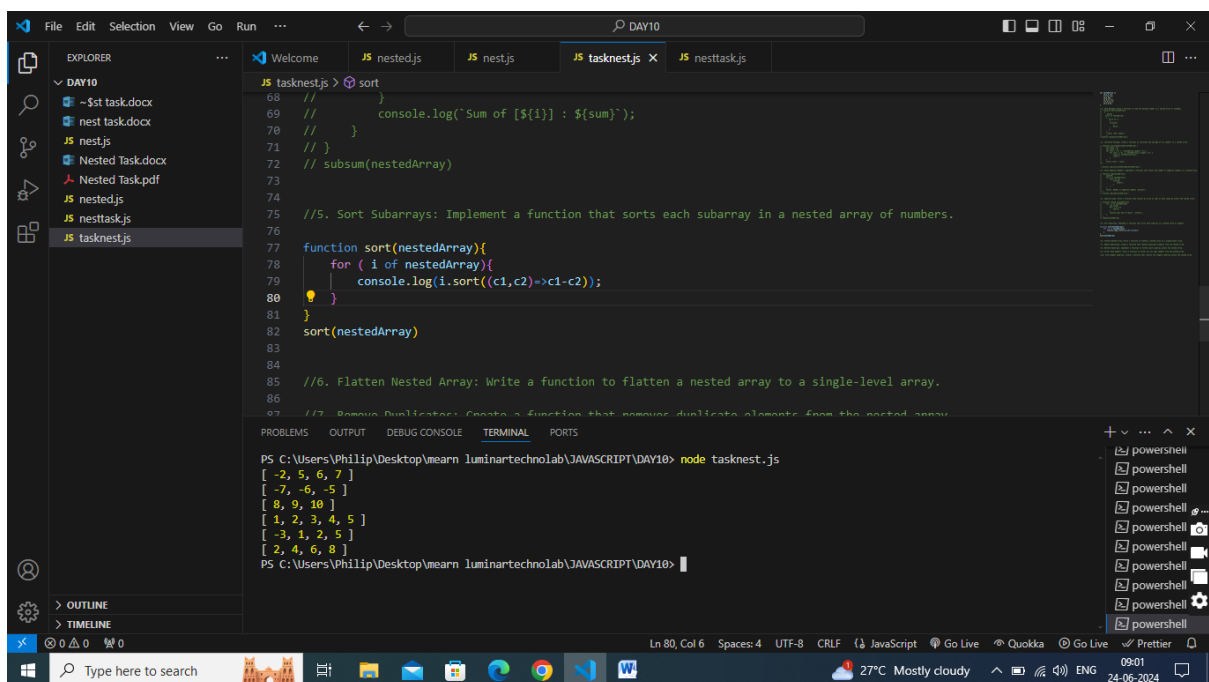
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Philip\Desktop\learn_luminartechnolab\JAVASCRIPT\DAY10> node tasknest.js
Max: 10
PS C:\Users\Philip\Desktop\learn_luminartechnolab\JAVASCRIPT\DAY10> node tasknest.js
Number of negative numbers :5
PS C:\Users\Philip\Desktop\learn_luminartechnolab\JAVASCRIPT\DAY10>

//4. Subarray Sums: Write a function that returns an array of sums of each subarray within the nested array.

```
59 console.log(nestedArray);
60
61 //4. Subarray Sums: Write a function that returns an array of sums of each subarray within the nested array.
62
63 function subsum (nestedArray){
64     for ( i of nestedArray){
65         let sum=0;
66         for( j of i){
67             sum+=j;
68         }
69         console.log(`Sum of [${i}] : ${sum}`);
70     }
71 }
72 subsum(nestedArray)
73
74
75
```

//5. Sort Subarrays: Implement a function that sorts each subarray in a nested array of numbers.



//6. Flatten Nested Array: Write a function to flatten a nested array to a single-level array.

```
81 // }
82 // sort(nestedArray)
83
84
85 //6. Flatten Nested Array: Write a function to flatten a nested array to a single-level array.
86
87 function flat(nestedArray){
88   newArray = [];
89   for( i of nestedArray){
90     for( j of i){
91       newArray.push(j);
92     }
93   }
94   return newArray;
95 }
96 console.log(flat(nestedArray));
97
98
99 //7. Remove Duplicates: Create a function that removes duplicate elements from the nested array.
```

```
[ -7, -6, -5 ]
[ 8, 9, 10 ]
[ 1, 2, 3, 4, 5 ]
[ -3, 1, 2, 5 ]
[ 2, 4, 6, 8 ]
PS C:\Users\Philip\Desktop\mearn_luminartechnolab\JAVASCRIPT\DAY10> node tasknest.js
[
  5, 6, 7, -2, -5, -6, -7, 8,
  9, 10, 3, 5, 2, 1, 4, -3,
  5, 2, 1, 4, 2, 6, 8
]
```

//7. Remove Duplicates: Create a function that removes duplicate elements from the nested array.

```
99 ///. Remove Duplicates: Create a function that removes duplicate elements from the nested array.
100
101 function removeDuplicatesNested(arr){
102   const seen = new Set();
103
104   return arr.map(innerArray => {
105     return innerArray.filter(item => {
106       if(!seen.has(item)){
107         seen.add(item);
108         return true;
109       }
110       return false;
111     });
112   });
113 }
114 const result = removeDuplicatesNested(nestedArray);
115 console.log(result);
116
117 //8. Reverse Subarrays: Implement a function to reverse each subarray within the nested array.
118
```

```
9, 10, 3, 5, 2, 1, 4, -3,
5, 2, 1, 4, 2, 6, 8
]
[
  [ 5, 6, 7, -2 ],
  [ -5, -6, -7 ],
  [ 8, 9, 10 ],
  [ 3, 2, 1, 4 ],
  [ -3 ],
  []
]
```

//8. Reverse Subarrays: Implement a function to reverse each subarray within the nested array.

The screenshot shows the VS Code editor with the file explorer on the left displaying a project structure for 'DAY10'. The main editor window shows the 'tasknestjs.js' file with the following code:

```
107 //      seen.add(item);
108 //      return true;
109 //    }
110 //    return false;
111 //  });
112 //  });
113 //  }
114 // const result = removeDuplicatesNested(nestedArray);
115 // console.log(result);
116
117 //8. Reverse Subarrays: Implement a function to reverse each subarray within the nested array.
118
119 function reverse(nestedArray){
120   return nestedArray.map(innerArray => innerArray.reverse());
121 }
122 console.log(reverse(nestedArray));
123
124
125 //9. Filter Even Numbers: Write a function to filter out all even numbers from the nested array.
```

The terminal at the bottom shows the command `node tasknest.js` and its output:

```
PS C:\Users\Philip\Desktop\learn_luminartechnolab\JAVASCRIPT\DAY10> node tasknest.js
[
  [ -2, 7, 6, 5 ],
  [ -7, -6, -5 ],
  [ 10, 9, 8 ],
  [ 4, 1, 2, 5, 3 ],
  [ 1, 2, 5, -3 ],
  [ 8, 6, 2, 4 ]
]
```

//9. Filter Even Numbers: Write a function to filter out all even numbers from the nested array.

The screenshot shows the VS Code editor with the file explorer on the left displaying a project structure for 'DAY10'. The main editor window shows the 'tasknestjs.js' file with the following code:

```
117 //8. Reverse Subarrays: Implement a function to reverse each subarray within the nested array.
118
119 // function reverse(nestedArray){
120 //   return nestedArray.map(innerArray => innerArray.reverse());
121 // }
122 // console.log(reverse(nestedArray));
123
124
125 //9. Filter Even Numbers: Write a function to filter out all even numbers from the nested array.
126
127 function even(nestedArray){
128   return nestedArray.map(innerArray => {
129     return innerArray.filter(item => item % 2 !== 0);
130   });
131 }
132 console.log(even(nestedArray));
133
134
135 //10. Find Longest Subarray: Create a function that returns the longest subarray within the nested array.
```

The terminal at the bottom shows the command `node tasknest.js` and its output:

```
PS C:\Users\Philip\Desktop\learn_luminartechnolab\JAVASCRIPT\DAY10> node tasknest.js
[
  [ -2, 7, 6, 5 ],
  [ -7, -6, -5 ],
  [ 10, 9, 8 ],
  [ 4, 1, 2, 5, 3 ],
  [ 1, 2, 5, -3 ],
  [ 8, 6, 2, 4 ]
]
PS C:\Users\Philip\Desktop\learn_luminartechnolab\JAVASCRIPT\DAY10> node tasknest.js
[ [ 5, 7 ], [ -5, -7 ], [ 9 ], [ 3, 5, 1 ], [ -3, 5, 1 ], [] ]
PS C:\Users\Philip\Desktop\learn_luminartechnolab\JAVASCRIPT\DAY10>
```

//10. Find Longest Subarray: Create a function that returns the longest subarray within the nested array.

