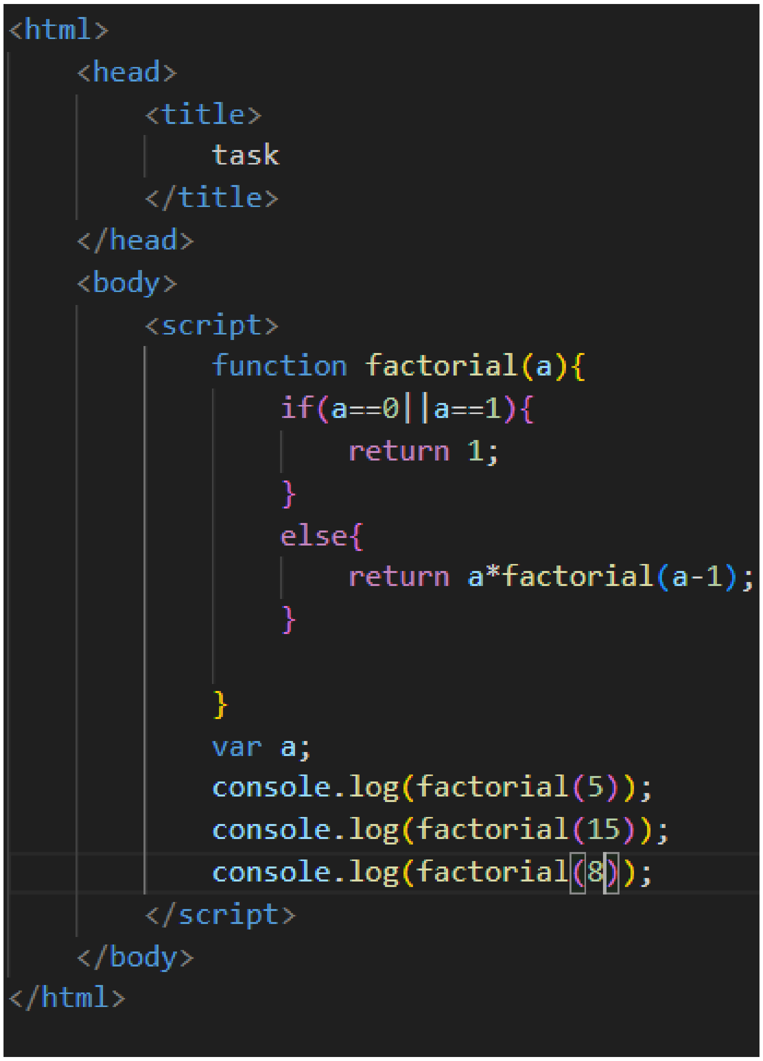
Task 1:

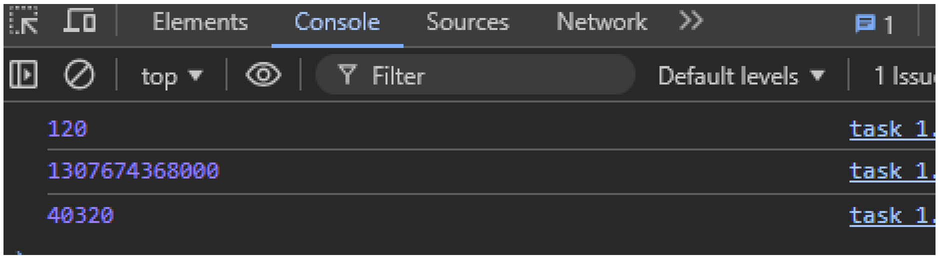
Implement a function to calculate the factorial of a number using recursion.

code:



```html
<html>
    <head>
        <title>
            task
        </title>
    </head>
    <body>
        <script>
            function factorial(a){
                if(a==0||a==1){
                    return 1;
                }
                else{
                    return a*factorial(a-1);
                }

            }
            var a;
            console.log(factorial(5));
            console.log(factorial(15));
            console.log(factorial(8));
        </script>
    </body>
</html>
```

output:

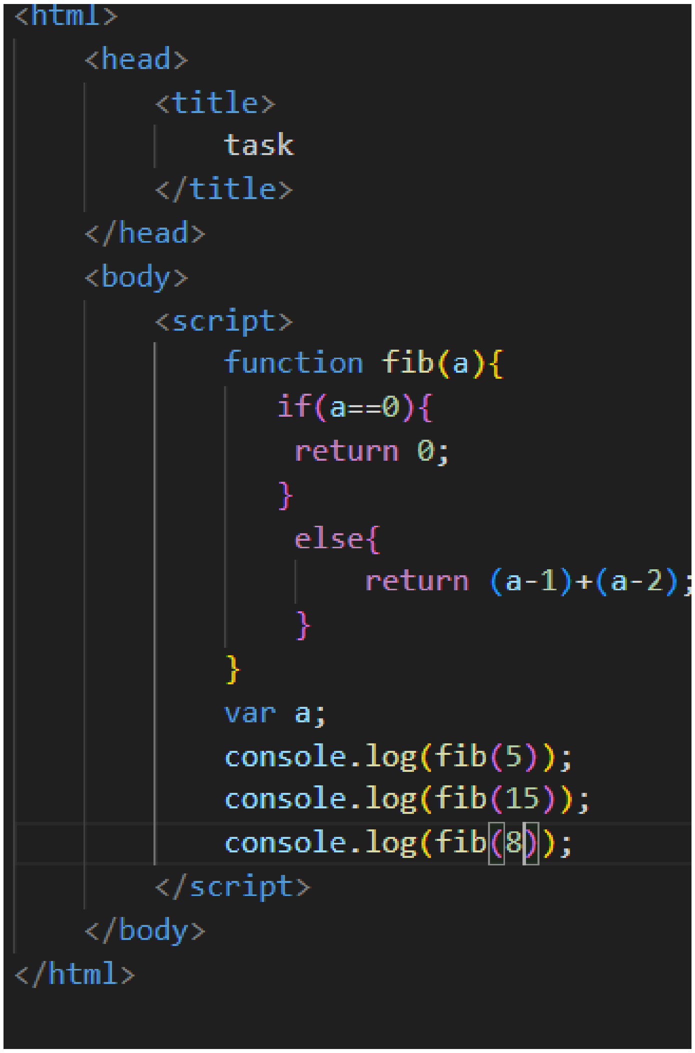


```
120                                      task 1.
1307674368000                            task 1.
40320                                    task 1.
```

Task 2:

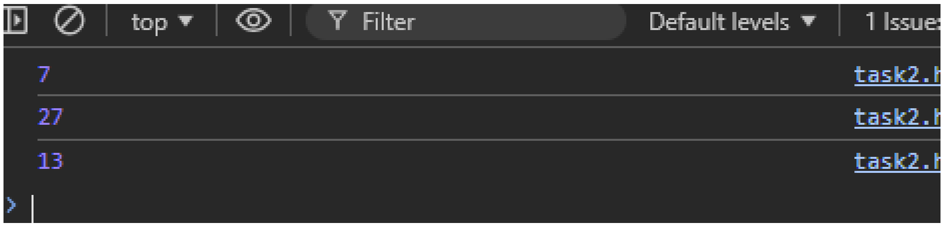
Write a recursive function to find the nth Fibonacci number.

code:



```
<html>
    <head>
        <title>
            task
        </title>
    </head>
    <body>
        <script>
            function fib(a){
               if(a==0){
                return 0;
               }
                else{
                    return (a-1)+(a-2);
                }
            }
            var a;
            console.log(fib(5));
            console.log(fib(15));
            console.log(fib(8));
        </script>
    </body>
</html>
```

output:

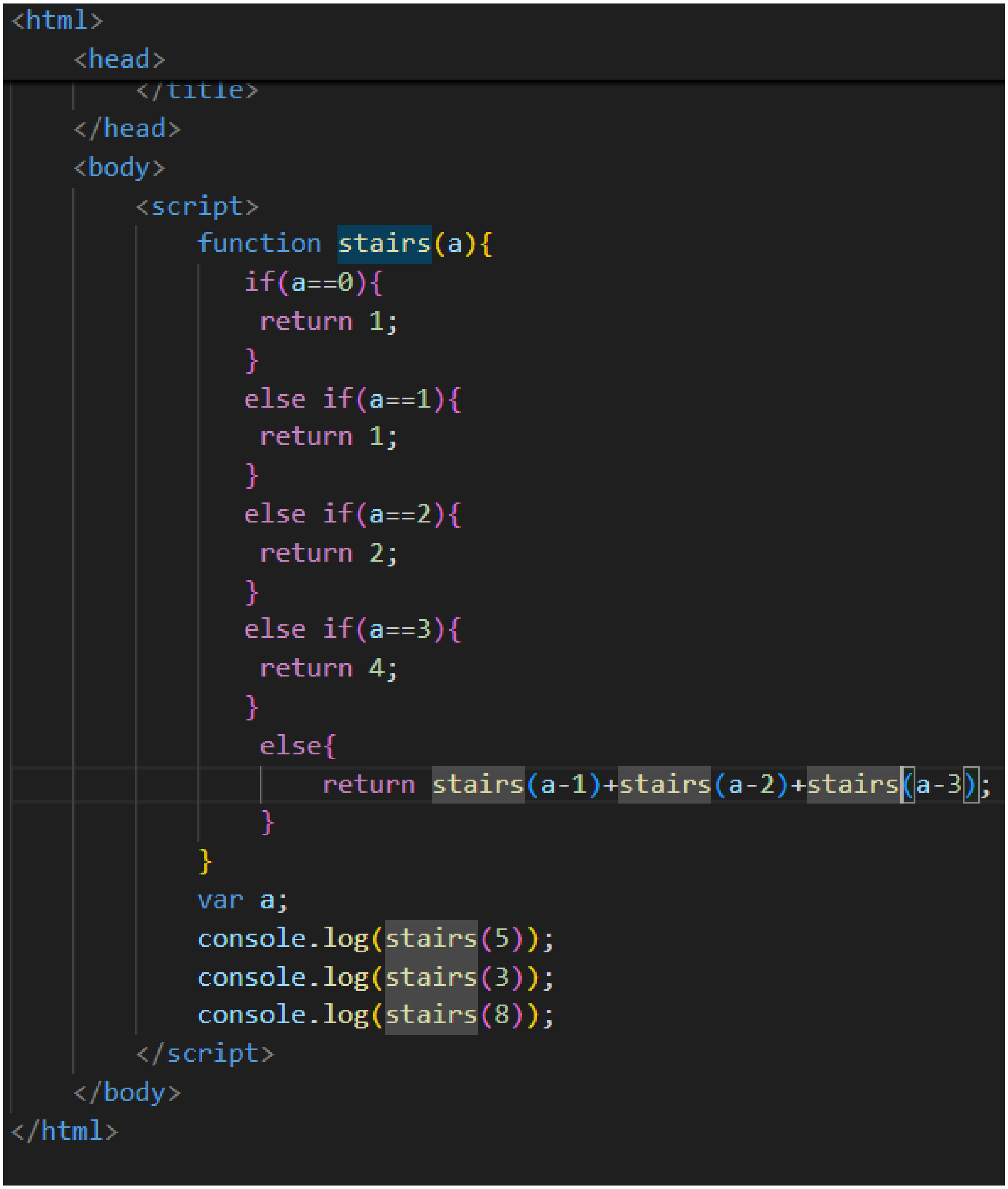


```
  7                                                      task2.h
  27                                                     task2.h
  13                                                     task2.h
>
```

Task 3:

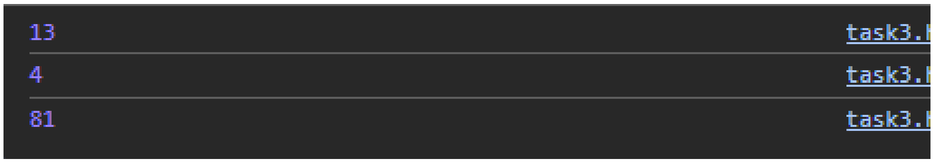
Create a function to determine the total number of ways one can climb astaircase with 1, 2, or 3 steps at a time using recursion.

code:



```
<html>
    <head>
        </title>
    </head>
    <body>
        <script>
            function stairs(a){
               if(a==0){
                return 1;
               }
               else if(a==1){
                return 1;
               }
               else if(a==2){
                return 2;
               }
               else if(a==3){
                return 4;
               }
                else{
                    return stairs(a-1)+stairs(a-2)+stairs(a-3);
                }
            }
            var a;
            console.log(stairs(5));
            console.log(stairs(3));
            console.log(stairs(8));
        </script>
    </body>
</html>
```

output:

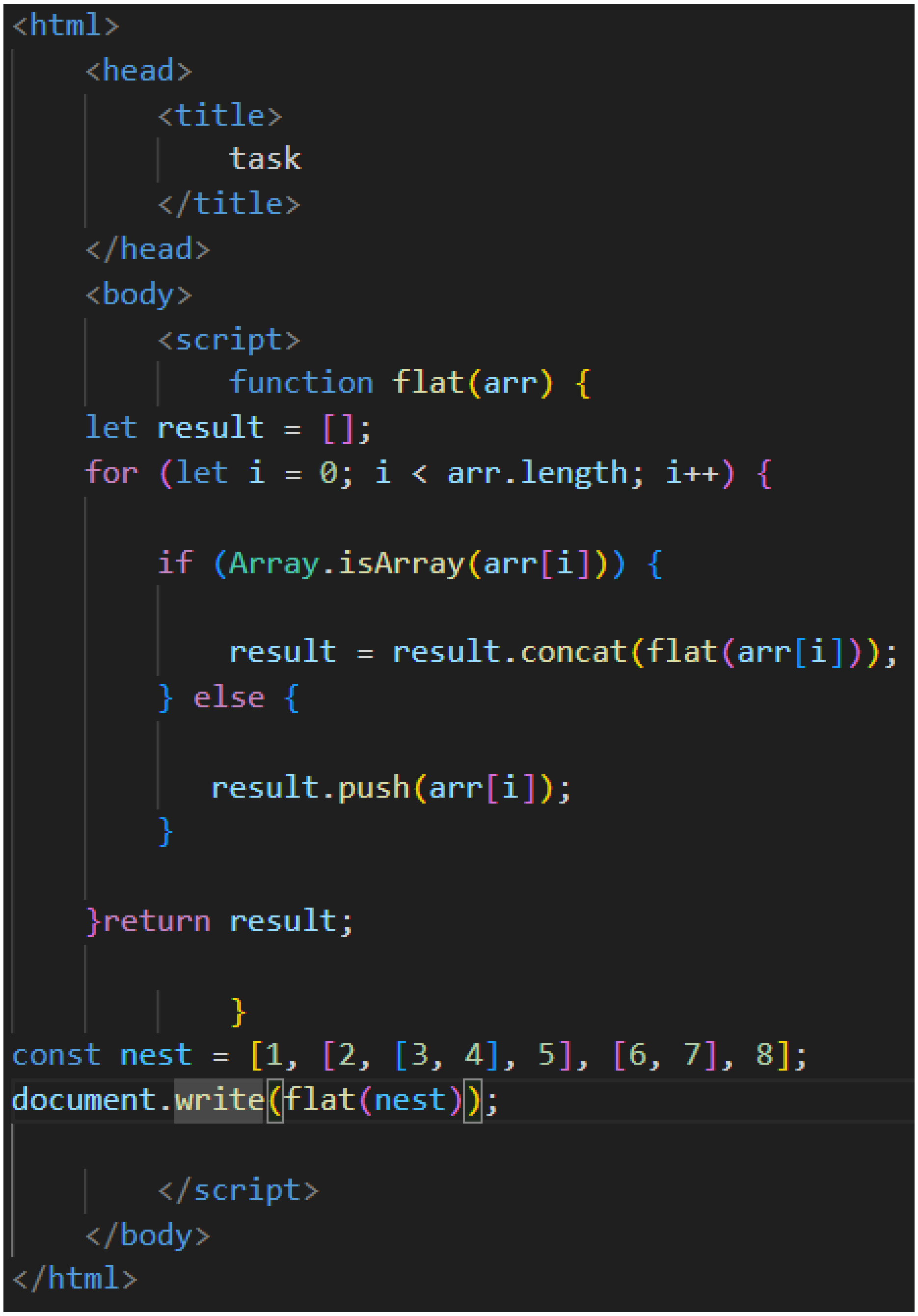


```
13                                                    task3.
4                                                     task3.
81                                                    task3.
```

Task 4:

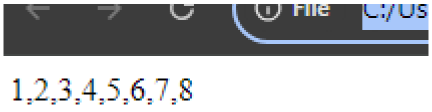
Write a recursive function to flatten a nested array structure.

code:



```
<html>
  <head>
    <title>
      task
    </title>
  </head>
  <body>
    <script>
      function flat(arr) {
  let result = [];
  for (let i = 0; i < arr.length; i++) {

    if (Array.isArray(arr[i])) {

      result = result.concat(flat(arr[i]));
    } else {

     result.push(arr[i]);
    }

  }return result;

      }
const nest = [1, [2, [3, 4], 5], [6, 7], 8];
document.write(flat(nest));

    </script>
  </body>
</html>
```

output:

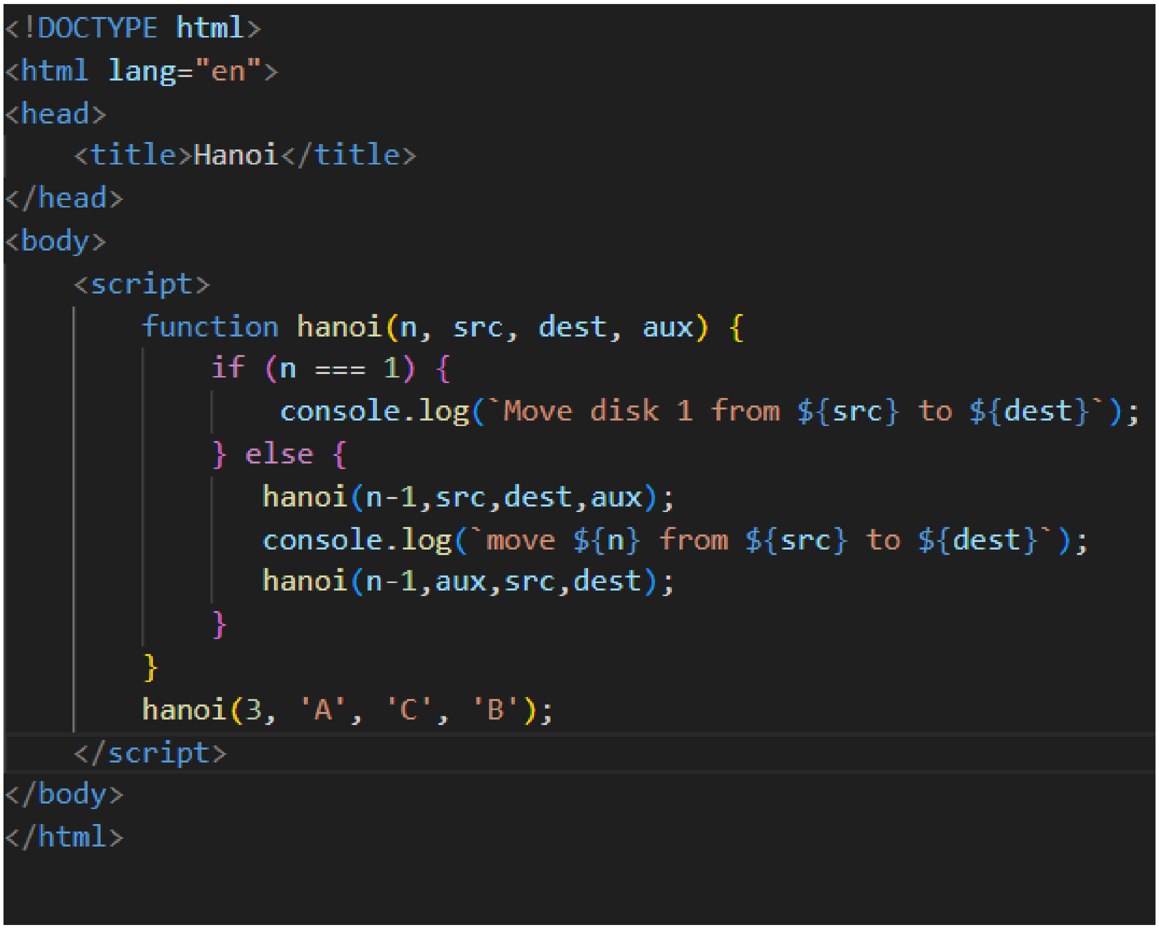


```
1,2,3,4,5,6,7,8
```

Task 5:

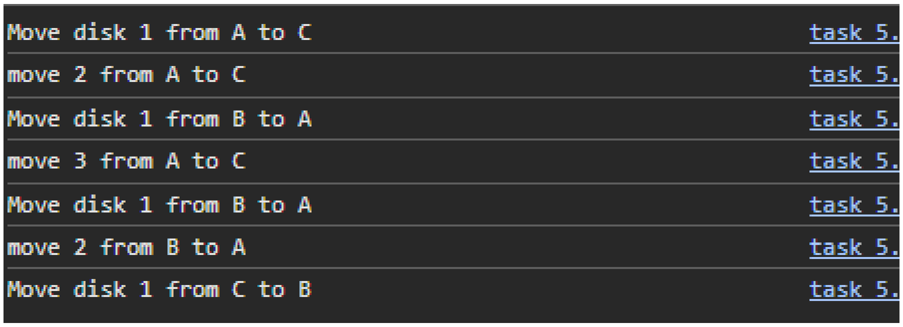
Implement the recursive Tower of Hanoi solution.

code:



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Hanoi</title>
</head>
<body>
    <script>
        function hanoi(n, src, dest, aux) {
            if (n === 1) {
                console.log(`Move disk 1 from ${src} to ${dest}`);
            } else {
               hanoi(n-1,src,dest,aux);
               console.log(`move ${n} from ${src} to ${dest}`);
               hanoi(n-1,aux,src,dest);
            }
        }
        hanoi(3, 'A', 'C', 'B');
    </script>
</body>
</html>
```

output:



```
Move disk 1 from A to C                                         task 5.
move 2 from A to C                                              task 5.
Move disk 1 from B to A                                         task 5.
move 3 from A to C                                              task 5.
Move disk 1 from B to A                                         task 5.
move 2 from B to A                                              task 5.
Move disk 1 from C to B                                         task 5.
```