ES6 Exercises

1. Spread Operator & Rest Parameter

I. Create a JavaScript function that takes a string as argument and returns a string with letters in alphabetical order.
Example string*:* 'dbca'
Expected Output*: 'abcd'*

Note: Use Arrow function with spread operator to achieve solution

II. **Solve using arguments object and spread operator**

Define a function that concatenates several strings in below format

– Function should take a separator as first argument and multiple strings arguments

– Function should return strings concatenated and separated by separator

Input: test(", ", "Hello Sir", "How are you", "My name is Mike");

Output - "Hello Sir, How are you, My name is Mike"

III. **Solve using arrow function and …rest param**

Define a function that concatenates several strings in below format

– Function should take a separator as first argument and multiple strings arguments

– Function should return strings concatenated and separated by separator

Input: test(", ", "Hello Sir", "How are you", "My name is Mike");

Output - "Hello Sir, How are you, My name is Mike"

IV. What will be the output?

const myfunc = (...arg, separator)=>arg.join(separator);

const myVal = myfunc(10,20,30, ",");

console.log(myVal);

V.   What will be the output?

```
const myfunc = ...arg =>arg.join(",");
const myVal = myfunc(10,20,30);
console.log(myVal);
```

VI.   **Solve using spread operator**
Write an arrow function that takes Integer as argument and returns integer in reverse
Example:
Input – funcReverse(123456);
Output – 654321

VII.   What will be the output?

```
let array1 = [1,2];
let array2 = [3,4,5];
array1.push(array2);
console.info(array1);
```

VIII.   **Solve using native apply method to push array elements into another array**
Write an arrow function that take 3 arrays as argument and appends last 2 arrays elements into first array and returns first array (Don't return new array, return same first array passed in argument after appending last 2 arrays elements to it)

IX.   **Solve using spread operator to push array elements into another array**

Write an arrow function that take 3 arrays as argument and appends last 2 arrays elements into first array and returns first array (Don't return new array, return same first array passed in argument after appending last 2 arrays elements to it)

2. Destructuring

I.   What will be the output?

```
function f() {
return [2, 5, 6];
}
let a, b;
[a , , b] = f();
console.log(a);
console.log(b);
```

II.   Write expression for destructuring below object without variable declaration
        let obj = { a: 2, b: 4 });

III.   Write expression for destructuring below object, extract company, name, age, state, spouse name
        Note : Assign to new variable names and provide default values for company and state


        ```
        let data = {
        role:"Engineer",
        company:"SR",
        personal: {
                name:"Sumit",
                age:25,
                country:"IND",
                state:undefined,
                spouse: {
                        name:"Reena"
                        }
                }
        }
        ```


IV.   Write an example of destructuring inside the for-of loop for below list to create a new list of objects with only name and company properties

        ```
         let arr = [
        {name:"Mike", age :25, role:"Engineer", company:"SR"},
        {name:"Sam", age :27, role:"Senior Engineer", company:"SR"} ,
        {name:"Peter", age :28, role:"Senior Program Manager", company:"SR"}
        ]
        ```

V.   Write expression for destructuring below object, extract children name

        ```
        let data = {
        role:"Engineer",
        company:"SR",
        personal: {
            name:"Sumit",
            age:25,
            country:"IND",
            state:undefined,
            children:[ {
                    name:"Mayank"
                    }]
            }
        }
        ```


VI.   Write a function to show destructuring for objects passed as function parameter

VII.   Write examples to show rest param usage in array and object destructuring