1 ES6 Assignment 1

1.1 Task 1: Create User Profiles

Create a `createUserProfile` function with the following:

- Use **destructuring with default values** to extract `name` (default: `"Anonymous"`) and a nested `theme` (default: `"light"`).
- Return a greeting string using a **template literal**.

Starter Code:

```
// Example Usage:
const user1 = { name: "Ahmed", preferences: { theme: "dark" } };
const user2 = { name: "Salma" };
const user3 = {};

console.log(createUserProfile(user1)); // Hello Ahmed! Your theme is dark.
console.log(createUserProfile(user2)); // Hello Salma! Your theme is light.
console.log(createUserProfile(user3)); // Hello Anonymous! Your theme is
↪  light.
```

1.2 Task 2: Manage Unique Roles & User Data

Part 1: Unique Roles

- Create a `getUniqueRoles` function that accepts an array of roles.
- Return a new array with duplicates removed.

Part 2: User Database

- Populate the `usersMap` from the `users` array, using the user `id` as the key.
- Create a `findUserById` function to retrieve a user from the map.

Starter Code:

```
// Part 1: Unique Roles
const userRoles = ["admin", "editor", "viewer", "editor", "admin"];

console.log(getUniqueRoles(userRoles)); // ['admin', 'editor', 'viewer']

// Part 2: User Database
const users = [
    { id: "u1", name: "Nour", email: "nour@example.com" },
    { id: "u2", name: "Karim", email: "karim@example.com" },
];

const usersMap = new Map(); // Your code here

console.log(findUserById("u2")); // { id: 'u2', name: 'Karim', email:
↪  'karim@example.com' }
```

1.3 Task 3: Safely Update Configuration

Create `updateTheme` function with the following:

- Accept a config object and a new theme.
- Return a **new object** with the updated theme, leaving the original unchanged.

Starter Code:

```javascript
const originalConfig = {
    user: "Admin",
    settings: {
        theme: "dark",
        notifications: {
            email: true,
        },
    },
};

const newConfig = updateTheme(originalConfig, "light");

// Verification:
console.log("New Config Theme:", newConfig.settings.theme); // light
console.log("Original Config Theme:", originalConfig.settings.theme); // dark
```

1.4 Task 4: Destructuring

Create a function `processData` that accepts an object with the following structure:

```javascript
const data = {
    id: 1,
    user: {
        name: "Ali",
        address: {
            city: "Cairo",
            postal: "12345",
        },
    },
    items: [
        { id: "i1", name: "Item 1", price: 100 },
        { id: "i2", name: "Item 2", price: 200 },
    ],
};
```

The function should:

1. Extract the `name` and `city` properties.
2. Extract the first item's `name` and `price`.
3. Provide default values for `postal` (default: `"00000"`) and the second item's `price` (default: `0`).
4. Return a string summarizing the extracted data.

```
console.log(processData(data));
// User Ali from Cairo bought Item 1 for 100 and Item 2 for 0.
```