1-) Create an application app.ts which will import the following three modules:

1. Module 01 - module01.ts

   Complete the code for isWeekend() function and return the string "weekend" or "weekday" without using if-statement.

```
export function isWeekend(): boolean {
      const todayDate: Date = new Date();
      const day: number = todayDate.getDay(); // 0 - 6 (0 is
Sunday)

      // your code here
}
```

2. Module 02 - module02.ts

   We want to create a curryable function that takes one argument:

   - An item object

   The returned function takes another one argument:

   - Discount value between 0 and 100 (a $100 item with a 10% discount will cost $90).

   Which will return the item with the price after we apply the discount.

   Complete the code for applyCoupon() as a curryable function, write your solution without mutating the original object.

```
      interface Iitem{
          name: string, type: string, category: string, price: number
      }
const item: Iitem = {
      "name": "Avocado",
      "type": "Fruit",
      "category": "Food",
      "price": 200
}
let applyCoupon: (i: Iitem) => (discount: number) => Iitem;

applyCoupon(item)(10).price === 180
```

3. Module 03 - module03.ts

   Write a function function isDual(arr: number[]): boolean{} that accepts an Array of numbers. An array is said to be dual if it has an even number of elements and each pair of consecutive elements sum to the same value.
   Return true if the array is dual, otherwise return false.

   Examples:

```
[1,2,3,0] // returns 1 (because 1+2 == 3+0 == 3)
[1, 2, 2, 1, 3, 0] // returns 1 (because 1+2 == 2+1 == 3+0 ==
3)
[1,1,2,2] // returns 0 (because 1+1 == 2 != 2+2)
[1,2,3] // returns 0 (because array does not have an even
number of elements)
[] //returns 1
```

   Transpile and bundle the application into a single JS file.

4. Complete the code for the ImmutableStorage class that will override setItem() to work with an immutable Storage object.

```
type Obj = { [key: string]: string };

class MyStorage {
    private storage: Obj = {};

    getItem(key: string): string {
        return this.storage[key]
    }
    setItem(key: string, value: string): void {
        this.storage[key] = value
    }
    getStorage(): Obj {
        return this.storage;
    }
}

class ImmutableStorage extends MyStorage {
    constructor() {
        super();
        Object.freeze(this.storage);
    }
    // complete the code here
}
```

5. Write a function function removeDuplicates(): number[] {} that will work on any Array object, the function will remove all the duplicate numbers from an array.

```
[4, 1, 5, 7, 2, 3, 1, 4, 6, 5, 2].removeDuplicates();

// returns: [4, 1, 5, 7, 2, 3, 6]
```

6. Write a function filterWords(string[]): string {} that will work on any String object. It accepts an array of strings that specifies a list of banned words. The function returns the string after replacing all the banned words with three stars.

```
console.log("This course is awesome".filterWords(['course',
'awesome']));

//"This *** is ***"
```

https://classroom.github.com/a/JYTYmZQT