```
    delete(num)
    {
        if(this.array.indexOf(num) != -1) // determine if it has the
element
        {
            let location = this.array.indexOf(num); //get the location of
specifical elemment
            this.array.splice(location,1); //delete the elemtent from the
location for once
        }
    }

    union (group2)
    {
        let union_group = new Group(); //create a new union array
        union_group.array = this.array.slice(); //deep copy the array to
the union one
        for(let i = 0 ; i < group2.array.length; i++) //loop for
group2.array.length times
        {
            if(!this.has(group2.array[i])) //if group2 has different
emelent that union array does not have
            {
                union_group.add(group2.array[i]); //combine them
            }
        }
        return union_group; // return the union array

    }

    intersection (group2)
    {
        let intersection_group = new Group(); // create a new intersection
array
        for (let i = 0; i < group2.array.length; i++)//loop for
group2.array.length times
        {
            if(this.has(group2.array[i])) //if this array has the same
element whit group2
            {
                intersection_group.add(group2.array[i]); // add this
element to the intersection array
            }
        }
        return intersection_group;// return the intersection array
    }

    difference (group2)
    {
        let diff_group = new Group();// create a new difference array
        diff_group.array = this.array.slice();//deep copy the array to the
diff one
        for (let i = 0; i < group2.array.length; i++)//loop for
group2.array.length times
        {
            if(diff_group.has(group2.array[i])) //if has the same element
whih this array
            {
                diff_group.delete(group2.array[i]); // delete this element
from diff array
            }
        }

        return diff_group;//return diffrence array

    }
}

let group1 = new Group();
let group2 = new Group();
group1.add(1);
group1.add(2);
group1.add(3);
console.log(group1);
group2.add(2);
group2.add(3);
group2.add(5);
group2.add(2);
console.log(group2);
console.log(group1.has(5));
console.log(group2.has(3));
console.log(group1.union(group2));
console.log(group1.intersection(group2));
console.log(group1.difference(group2));
group1.delete(1);
console.log(group1);
group2.delete(1);
console.log(group2);
```

```
▼ Group {array: Array(3)}                                    VM38:88
  ▶ array: (2) [2, 3]
  ▶ [[Prototype]]: Object
▼ Group {array: Array(3)}                                    VM38:93
  ▶ array: (3) [2, 3, 5]
  ▶ [[Prototype]]: Object
false                                                        VM38:94
true                                                         VM38:95
▼ Group {array: Array(4)}                                    VM38:96
  ▶ array: (4) [1, 2, 3, 5]
  ▶ [[Prototype]]: Object
▼ Group {array: Array(2)}                                    VM38:97
  ▶ array: (2) [2, 3]
  ▶ [[Prototype]]: Object
▼ Group {array: Array(1)}                                    VM38:98
  ▶ array: [1]
  ▶ [[Prototype]]: Object
▼ Group {array: Array(2)}                                    VM38:100
  ▶ array: (2) [2, 3]
  ▶ [[Prototype]]: Object
▼ Group {array: Array(3)}                                    VM38:102
  ▶ array: (3) [2, 3, 5]
  ▶ [[Prototype]]: Object
← undefined
>
```