# GKSTR35 Count_Even

```
        // take input

        // func call and collect

    }

    public static int count(int arr[]){
        int count = 0;
        for(int i = 0; i < n;i++){
            if(arr[i] % 2==0){
                count++;
            }
        }
        return count;
    }
}
```

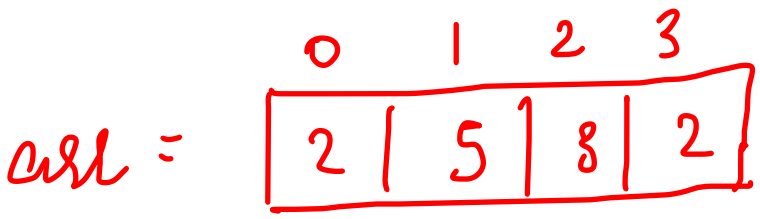
# Maximum of Array

```
        // take input

        // func call and collect

    }

    public static int max(int arr[]){
        int max = -1;
        for(int i = 0; i < arr.length; i++){
            if(arr[i] > max){
                max = arr[i];
            }
        }
        return max;
    }
```

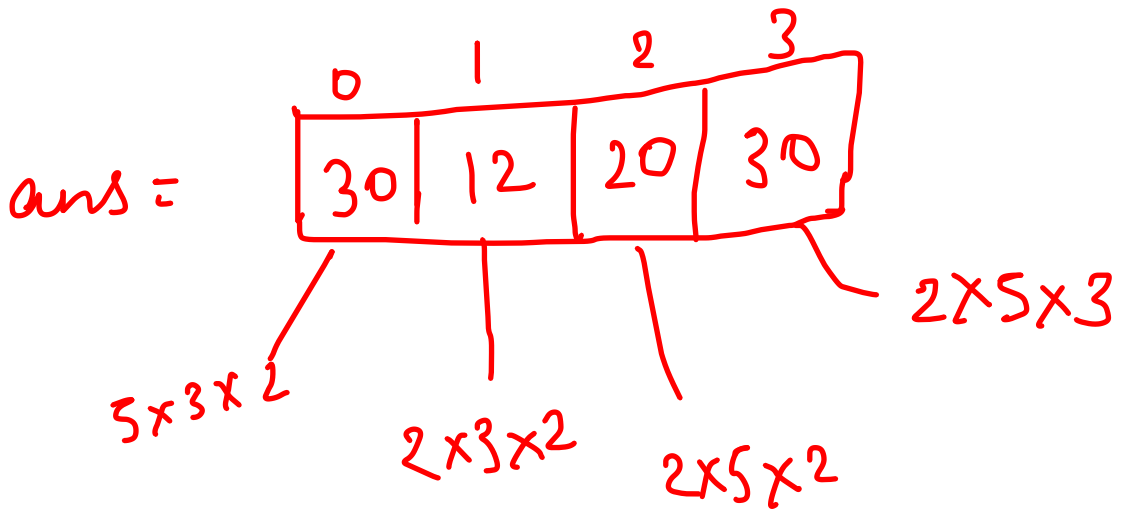
# Product of Elements Except Itself

V. Imp.

$n = 4$








Note → nested loops means running entire array for each element

## pseudo code-

1) i/p array.
2) traverse from 0 to (n-1) → i
   2.1) declare prod = 1
   2.2) traverse from 0 - (n-1) → j
   2.3) check if i != j
        prod x= arr[i]
3) Print prod.

i = 3, j = 0, p = 1 x 2 = 2
j = 1, p = 10
j = 2, p = 30
j = 3, p = 30

## Dry Run.

| 2 | 5 | 3 | 2 |
|---|---|---|---|
| 0 | 1 | 2 | 3 |

i = 0, j = 0, prod. = 1
j = 1, prod. = 5
j = 2, p = 15
j = 3, p = 30

---

i = 1, j = 0, p = 1 x 2 = 2
j = 1, p = 2
j = 2, p = 2 x 3 = 6
j = 3, p = 6 x 2 = 12

---

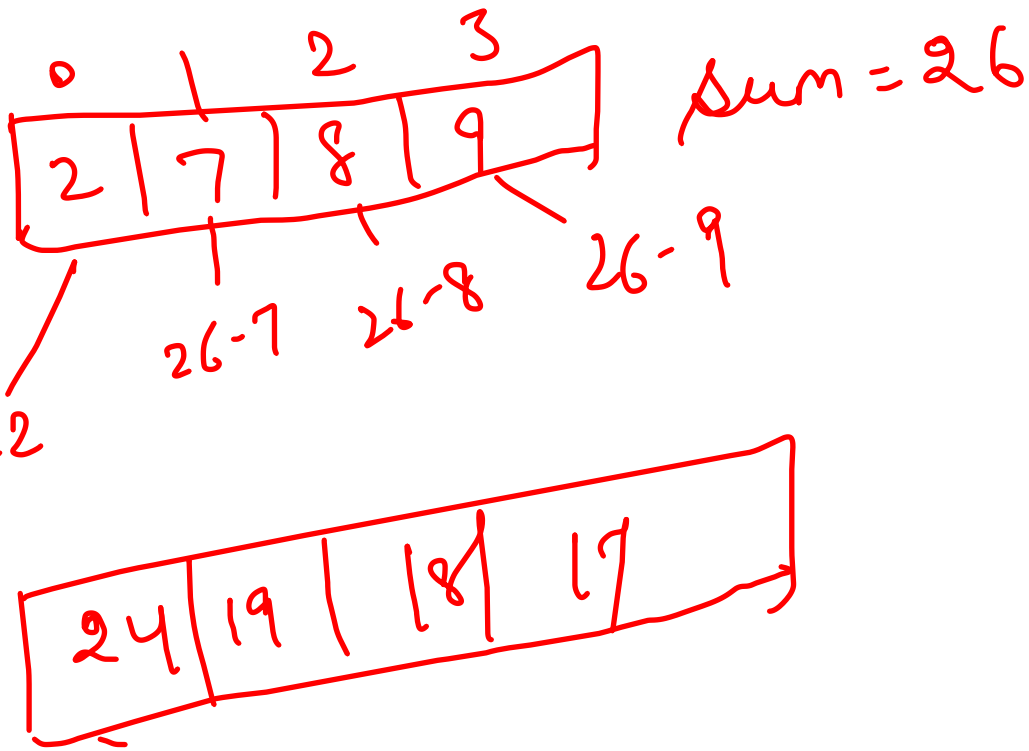
i = 2, j = 0, p = 1 x = 2
j = 1, p = 10
j = 2, p = 40
j = 3, p = 20

Error TLE

Time limit Exceed.

```
        Scanner s = new Scanner(System.in);
        int n = s.nextInt();
        int arr[] = new int[n];
        for(int i = 0 ;i < n; i++){
            arr[i] = s.nextInt();
        }

        product(arr,n);
    }

    public static void product(int arr[], int n){

        for(int i = 0; i < n; i++){
            int prod = 1;
            for(int j = 0; j < n; j++){
                if(i != j){
                    prod *= arr[j];
                }
            }
            System.out.println(prod);
        }
```

# Sum except itself







i=0, 26-2

i=1, 26-7

i=2, 26-8

i=3, 26-9

// take i/p

int totalsum = 0;

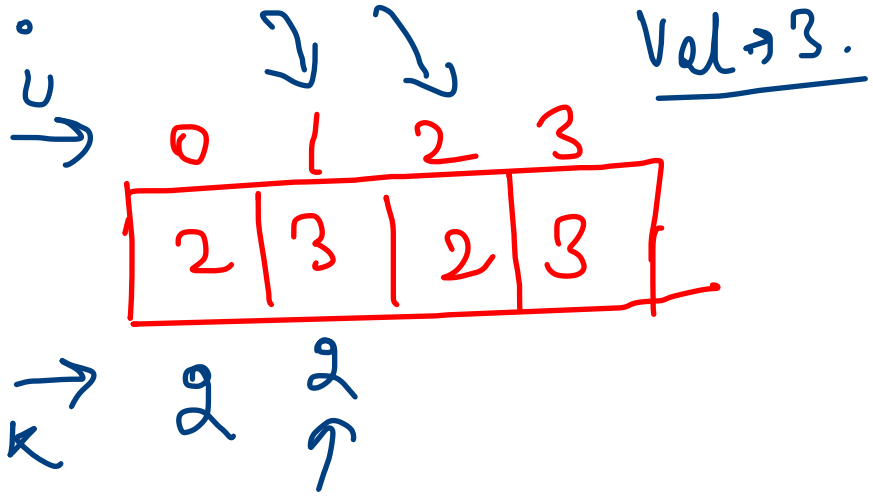
for( i=0 ; <n ; i++)

sum += arr[i] ; → 26

//output

for(i=0; <n ; i++)

Syso(totalsum - arr[i]);




Val → 3.

i = 0 < 4

a[i] != 3 T

arr[k] = a[i]

k++;

i = 1 < 4

a[i] != 3 F

i = 2 < 4

a[i] != 3 T

arr[k] = a[i]

i = 3 < 4

a[i] != 3 F

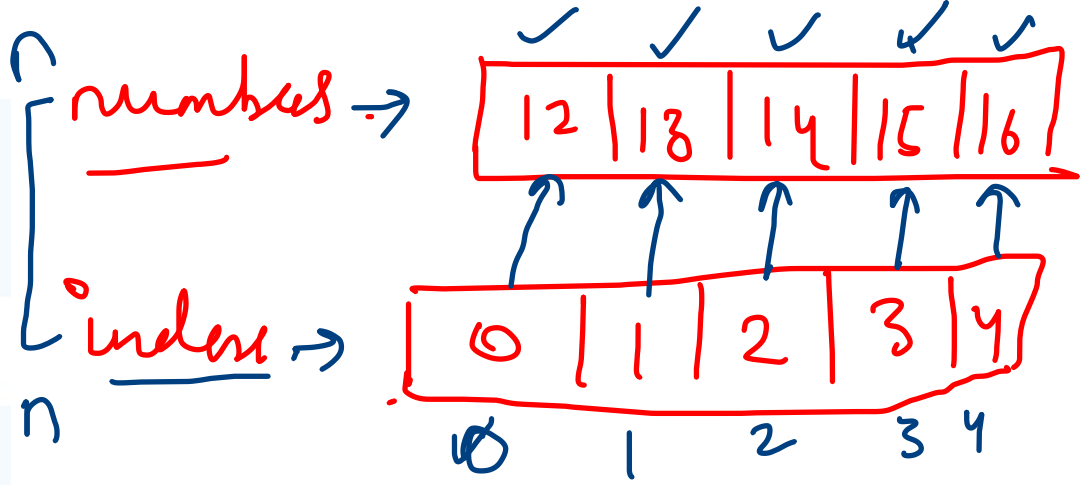
k = 2

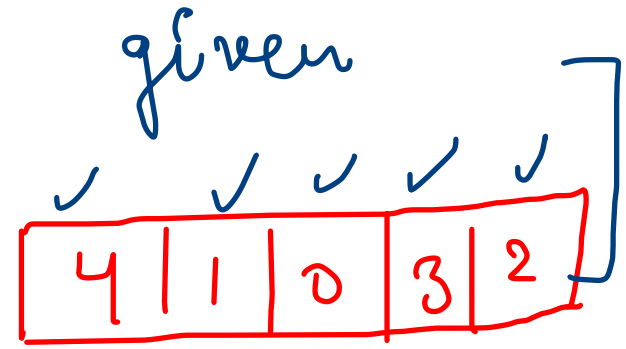
```
    // arr take input
    // val input
        // func call and collect

    }

    public static int remove(int arr[], int val){

        int k = 0;
        for(int i = 0; i < arr.length; i++){
            if(arr[i] != val){
                arr[k] = arr[i];
                k++;
            }
        }
        return k;
    }
}
```

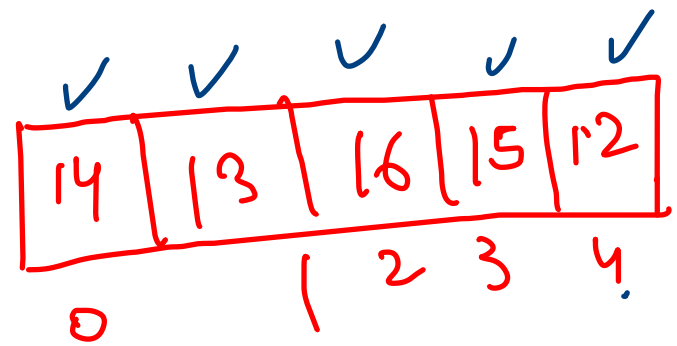
```
5
12 13 14 15 16
0 1 2 3 4
```

Sample Output 0

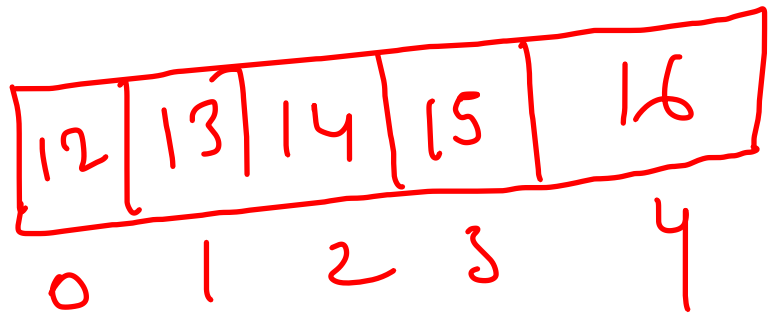
```
12 13 14 15 16
```

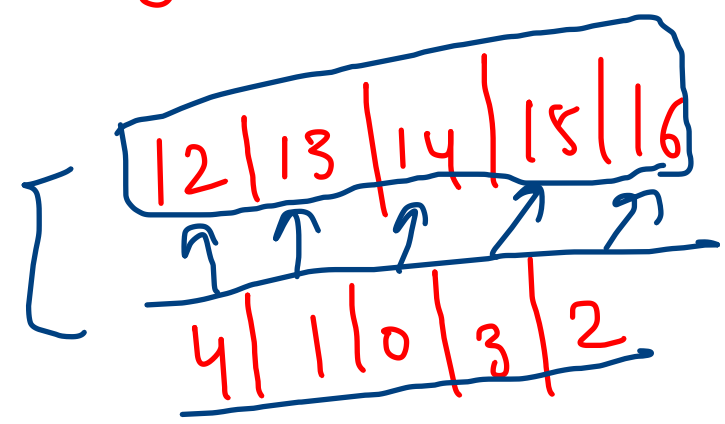



o/p. n [target




target[index[i]] = numbers[i]

target[index[0]] = num[0]

target[0] = 12

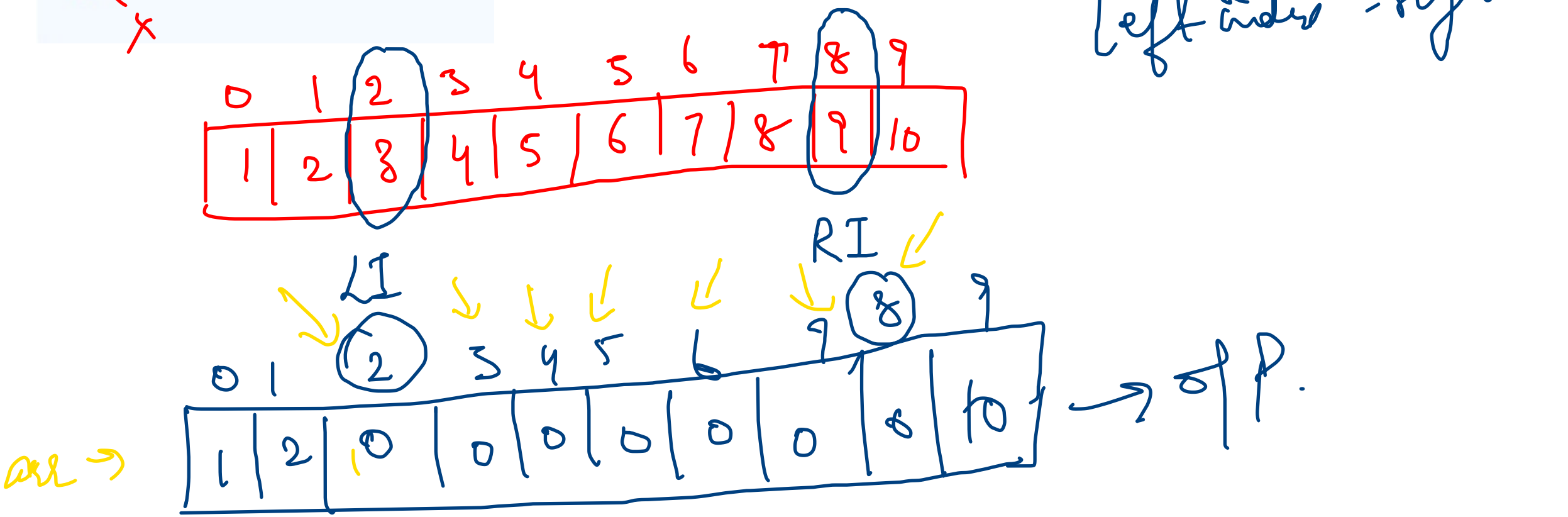
target[index[0]] = nums[0]

target[4] = num[0]

4 → 12

```
        Scanner s = new Scanner(System.in);

        int n = s.nextInt();

        int nums[] = new int[n];
        for(int i = 0; i < n; i++){
            nums[i] = s.nextInt();
        }

         int index[] = new int[n];
        for(int i = 0; i < n; i++){
            index[i] = s.nextInt();
        }

        int arr[] = targetArray(nums, index, n);
        for(int i = 0; i < arr.length; i++){
            System.out.print(arr[i] + " ");
        }
    }

    public static int[] targetArray(int nums[], int index[], int n){
        int target[] = new int[n];
        for(int i = 0; i < n; i++){
            target[index[i]] = nums[i];
        }
        return target;
    }
}
```

```
10
1 2 3 4 5 6 7 8 9 10
2 8
0
```



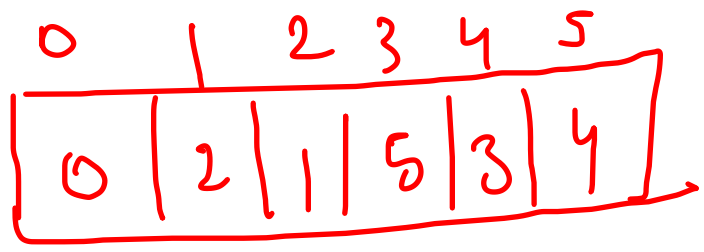

```
public static int[] update(int arr[] , int left, int right, int x){
    for(int i = left; i <= right; i++){
        arr[i] = x;
    }
    return arr;
}
```

```
    // array input

    // int left index input
    // right index input
    // int x input

    // function call, collect and print

}
```

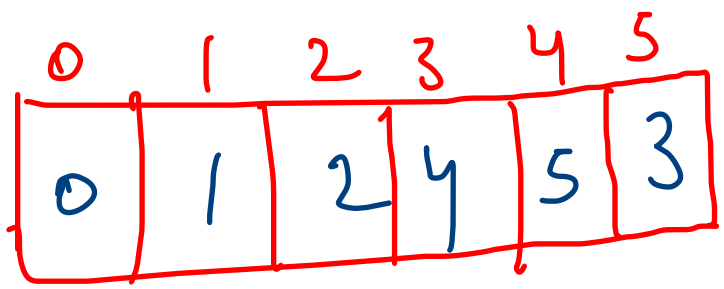
```
6
0 2 1 5 3 4
```

→

nums =




ans =




i = 0, nums[nums[i]]
nums[nums[0]]
nums[0] → 0

i = 1, nums[nums[1]]
nums[2] → 1

i = 2, nums[nums[2]]
nums[1] → 2

i = 3, nums[nums[3]]
nums[5] → 4

i = 4, nums[nums[4]]
nums[3] → 5

i = 5, nums[nums[5]]
nums[4] → 3

✓ ans[i] = nums[nums[i]]

# Print Pair

```
        // array input
        // function call
    }

    public static void pairs(int arr[], int n){
        for(int i = 0; i < n; i++){
            for(int j = i +1; j < n; j++){
                Syso(arr[i] + " " + arr[j]);
            }
        }
    }
}
```

# Find all Combination

```
        // array input
        // k input
        // funct call
    }

    public static void pairs(int arr[], int n, int k){
        for(int i =0 ; i < n; i++){
            for(int j = i; j < n; j++){
                if(arr[i] + arr[j] == k){
                    Syso(arr[i] + " " + arr[j]);
                }
            }
        }
    }
```