HW_Print All Composite Number of Array

non-prime no.

5%1 == 0

5%5 == 0

n=5

1	4	8	11	14
0	1	2	3	4

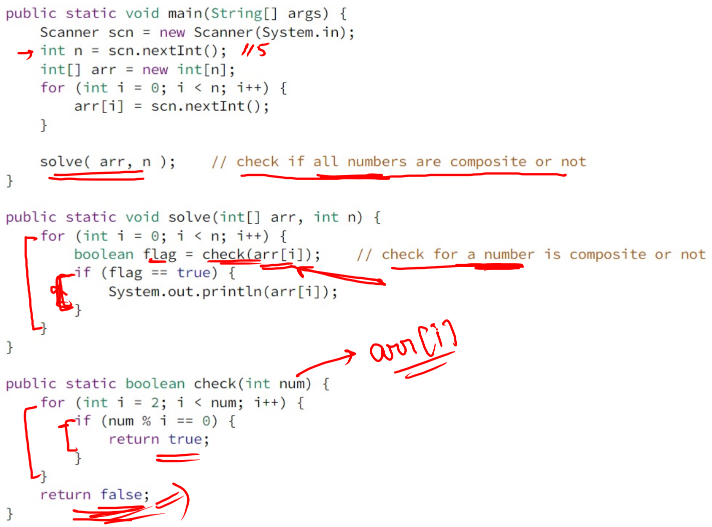
1%1 == 0

1%1 == 0

4
8
14

prime no. ⇒ only divide with 1 & self

⇒ means prime no. only have 2 factors



```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();  //5
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    solve( arr, n );    // check if all numbers are composite or not
}

public static void solve(int[] arr, int n) {
    for (int i = 0; i < n; i++) {
        boolean flag = check(arr[i]);    // check for a number is composite or not
        if (flag == true) {
            System.out.println(arr[i]);
        }
    }
}

public static boolean check(int num) {
    for (int i = 2; i < num; i++) {
        if (num % i == 0) {
            return true;
        }
    }
    return false;
}
```

arr[i]

console

8
6

n=5

2	8	3	6	11
0	1	2	3	4

i=0, false

i=1, true

i=2, false

i=3, true

i=4, false

~~1~~, 2, 3, - - - - - - , 10, ~~11~~

⇒ Max of array

↑ max = ~~-∞~~ ~~2~~ ~~3~~ 7

arr

2	-2	3	0	7	6	1
0	1	2	3	4	5	6

current value

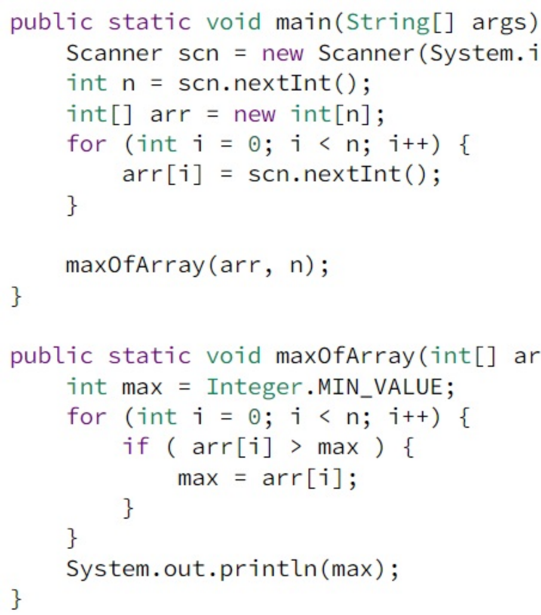
current maxi value

if (arr[i] > max)

max = arr[i]

max = -∞

i=0,	2 > -∞	true,	max = 2
i=1,	-2 > 2	false,	max = 2
i=2,	3 > 2	true,	max = 3
i=3,	0 > 3,	false,	max = 3
i=4,	7 > 3,	true,	max = 7
i=5,	6 > 7	false	max = 7
i=6,	1 > 7	false	$\boxed{max = 7}$



```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    maxOfArray(arr, n);
}

public static void maxOfArray(int[] arr, int n) {
    int max = Integer.MIN_VALUE;
    for (int i = 0; i < n; i++) {
        if ( arr[i] > max ) {
            max = arr[i];
        }
    }
    System.out.println(max);
}
```

//MAX_VALUE

if (arr[i] < min)

Integer.MIN_VALUE = -∞

Integer.MAX_VALUE = +∞

→ Integer $[-2^{31}, +2^{31}-1]$

Product of Elements Except Itself

arr	2	5	3	1	4
	0	1	2	3	4

$i=0,\ ans = 5\times3\times1\times4 = 60$

$i=1,\ ans = 2\times3\times1\times4 = 24$

$i=2,\ ans = 2\times5\times1\times4 = 40$

$i=3,\ ans = 2\times5\times3\times4 = 120$

$i=4,\ ans = 2\times5\times3\times1 = 30$

logic

step1 $product = 2\times5\times3\times1\times4$

$= 120$

step2

$i=0,\ ans = 120/2 = 60$

$i=1,\ ans = 120/5 = 24$

$i=2,\ ans = 120/3 = 40$

$i=3,\ ans = 120/1 = 120$

$i=4,\ ans = 120/4 = 30$

$\rightarrow \dfrac{\cancel{2}\times5\times3\times1\times4}{\cancel{2}}$

$\rightarrow \dfrac{2\times\cancel{5}\times3\times1\times4}{\cancel{5}}$

$i=0,\ \dfrac{\cancel{2}\times5\times3\times1\times4}{\cancel{2}} = 5\times3\times1\times4$

$i=1,\ \dfrac{2\times\cancel{5}\times3\times1\times4}{\cancel{5}} = 2\times3\times1\times4$

$i=2,\ \dfrac{2\times5\times\cancel{3}\times1\times4}{\cancel{3}} = 2\times5\times1\times4$

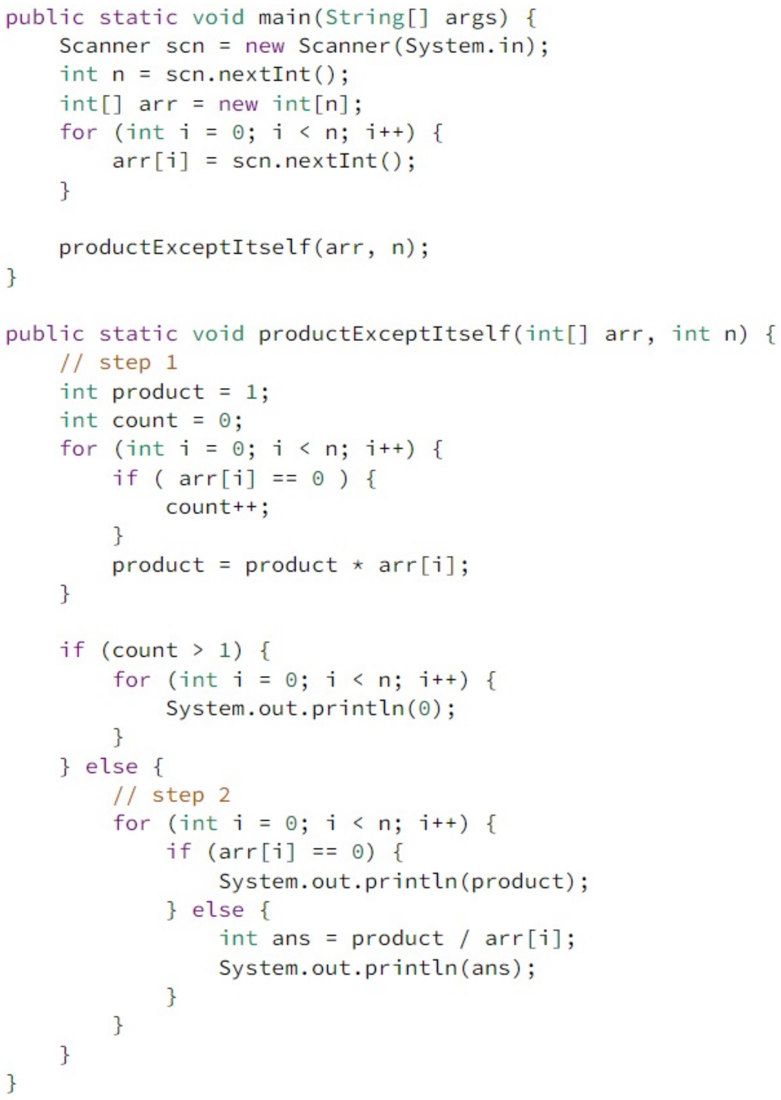
$i=3,\ \dfrac{2\times5\times3\times\cancel{1}\times4}{\cancel{1}} = 2\times5\times3\times4$

$i=4,\ \dfrac{2\times5\times3\times1\times\cancel{4}}{\cancel{4}} = 2\times5\times3\times1$

logic

product of every element except itself $\Leftrightarrow$ product of all elements / current element

TLE = time limit exceed



```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    productExceptItself(arr, n);
}

public static void productExceptItself(int[] arr, int n) {
    // step 1
    int product = 1;
    int count = 0;
    for (int i = 0; i < n; i++) {
        if ( arr[i] == 0 ) {
            count++;
        }
        product = product * arr[i];
    }

    if (count > 1) {
        for (int i = 0; i < n; i++) {
            System.out.println(0);
        }
    } else {
        // step 2
        for (int i = 0; i < n; i++) {
            if (arr[i] == 0) {
                System.out.println(product);
            } else {
                int ans = product / arr[i];
                System.out.println(ans);
            }
        }
    }
}
```

HW_Print Sum of Elements Except Itself

```
public static void sumExceptItself(int[] arr, int n) {
    // step 1
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum = sum + arr[i];
    }

    // step 2
    for (int i = 0; i < n; i++) {
        int ans = sum - arr[i];
        System.out.println(ans);
    }
}
```