Ques Add one

arr

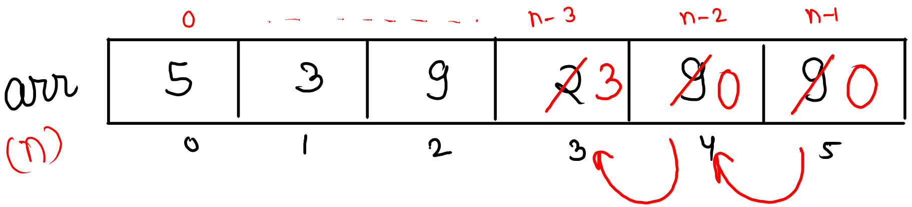
5	3	9	2	9	9
0	1	2	3	4	5

num = 539299

num+1 = ??

```
    1 1
5 3 9 2 9 9
↓ ↓ ↓    +1
-----------
5 3 9 3 0 0
```

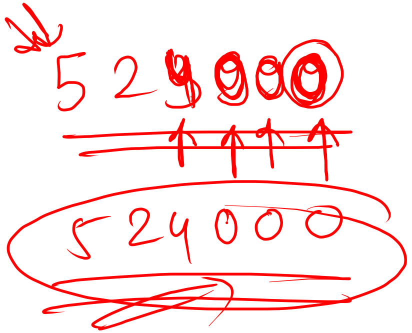
possibilities ⇒ 0 to 9

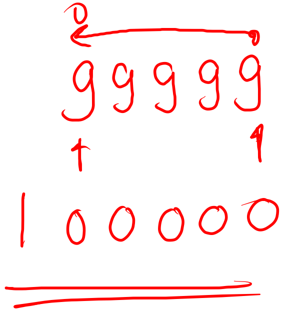


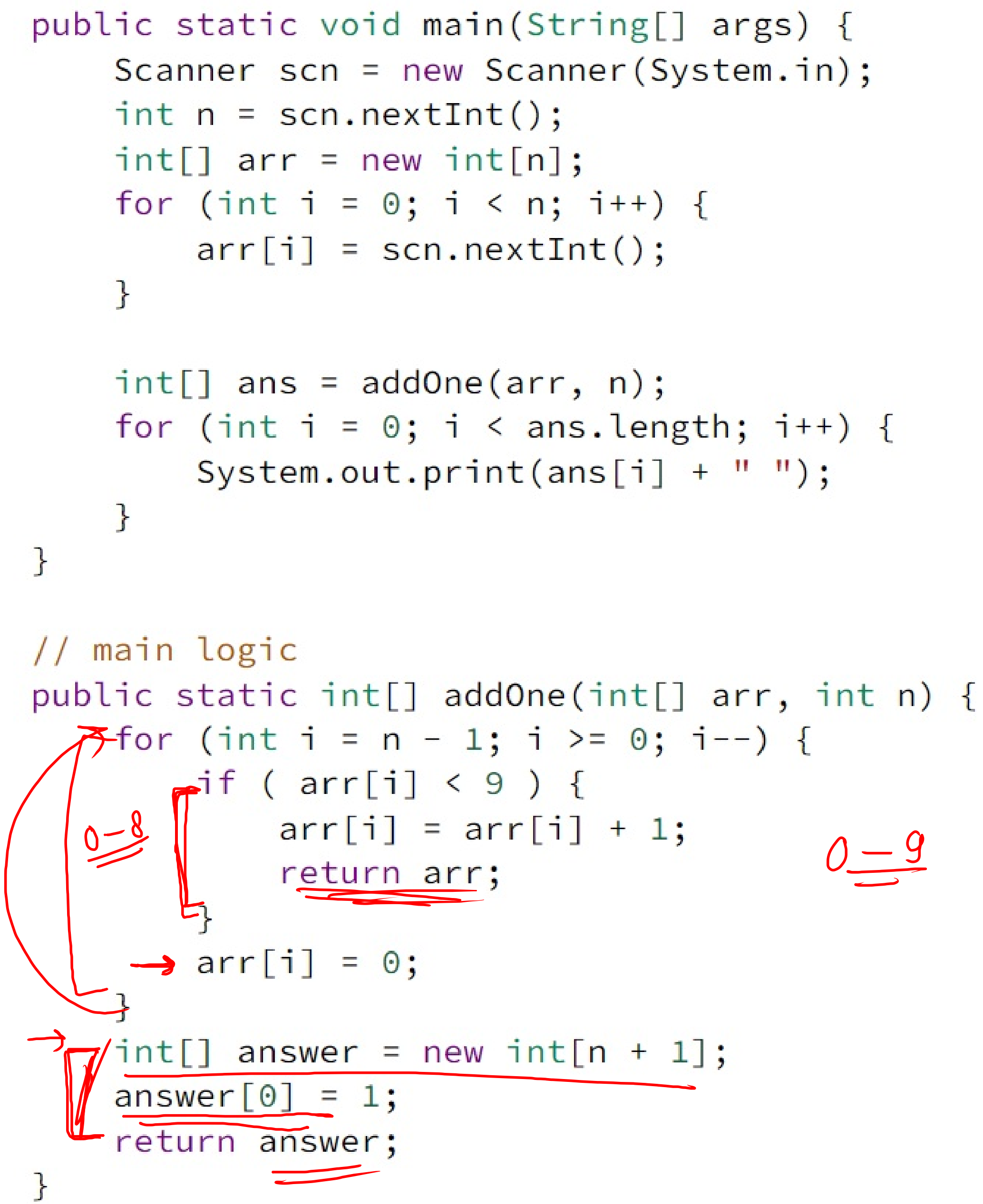
approch :-

1) Iterate from last to start
2) if val is b/w 0 to 8
   2.1) add 1 in the idex
   2.2) Stop the f^n (return)
3) if val is 9 then
   3.1) update value to 0
   3.2) Check for previous index

Code



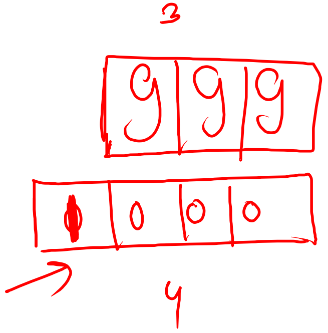




```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

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

// main logic
public static int[] addOne(int[] arr, int n) {
    for (int i = n - 1; i >= 0; i--) {
        if ( arr[i] < 9 ) {
            arr[i] = arr[i] + 1;
            return arr;
        }
        arr[i] = 0;
    }
    int[] answer = new int[n + 1];
    answer[0] = 1;
    return answer;
}
```



⇒ Nested loops

arr	1	2	3	4	5	6
	0	1	2	3	4	5
	↑ i	↑ j				

(N^2) n = size

pairs

(15)

1, 2
1, 3
1, 4
1, 5
1, 6
↑ ↑
i j

2, 3
2, 4
2, 5
2, 6
↑ ↑
i j

3, 4
3, 5
3, 6
↑ ↑
i j

4, 5
4, 6
↑ ↑
i j

5, 6
↑ ↑
i j

i<=n-2

```
for(int i=0; i<n-1; i++){
    for(int j= i+1; j<n; j++){
        Syso( arr[i] + "  "+ arr[j]);
    }
}
```

Product of Elements Except Itself

arr	2	5	3
	0	1	2

$n=3$

$i=0,\quad ans = 5*3 = 15$

$i=1,\quad ans = 2*3 = 6$

$i=2,\quad ans = 2*5 = 10$

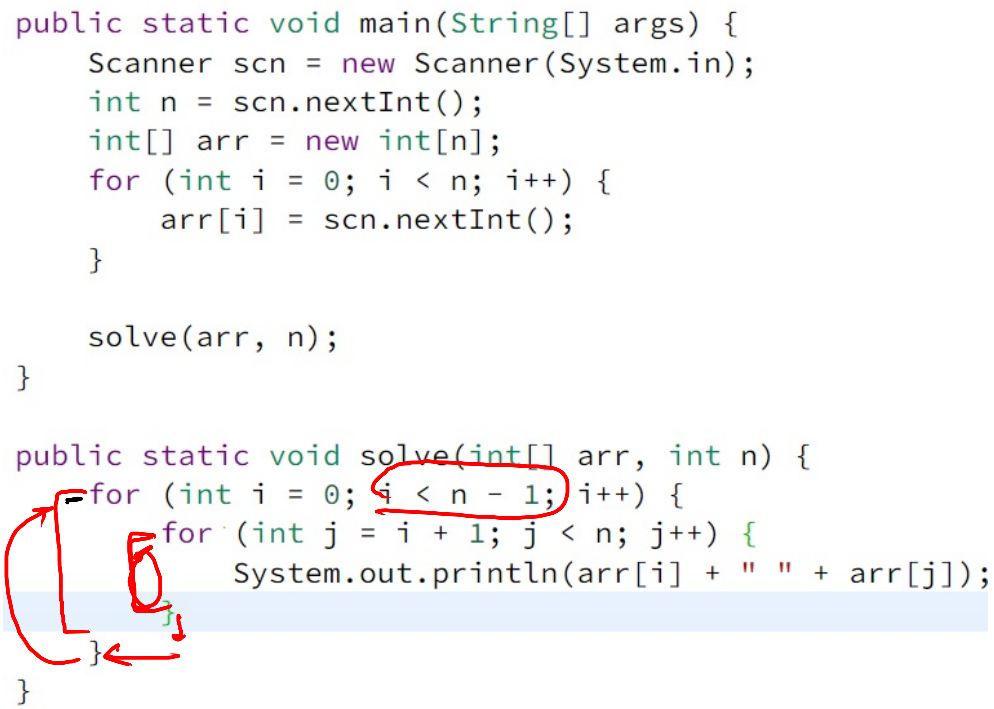
$i=0,\quad j = \overset{✗}{0}, \overset{✓}{1}, \overset{✓}{2}$

$i=1,\quad j = \underset{✓}{0}, \underset{✗}{1}, \underset{✓}{2}$

$i=2,\quad j = \underset{✓}{0}, \underset{✓}{1}, \underset{✗}{2}$

```
for(int i = 0; i<n; i++){
    int ans=1;
    for (int j=0; j<n; j++){
        if ( i!=j){
            ans= ans* arr[j];
        }
    }
    syso(ans);
}
```

TLE:- Time limit exceed

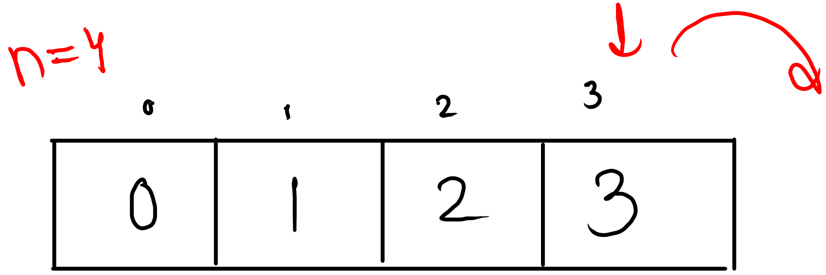


```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    solve(arr, n);
}

public static void solve(int[] arr, int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            System.out.println(arr[i] + " " + arr[j]);
        }
    }
}
```

n=4



i=0, j=1 (0, 1)

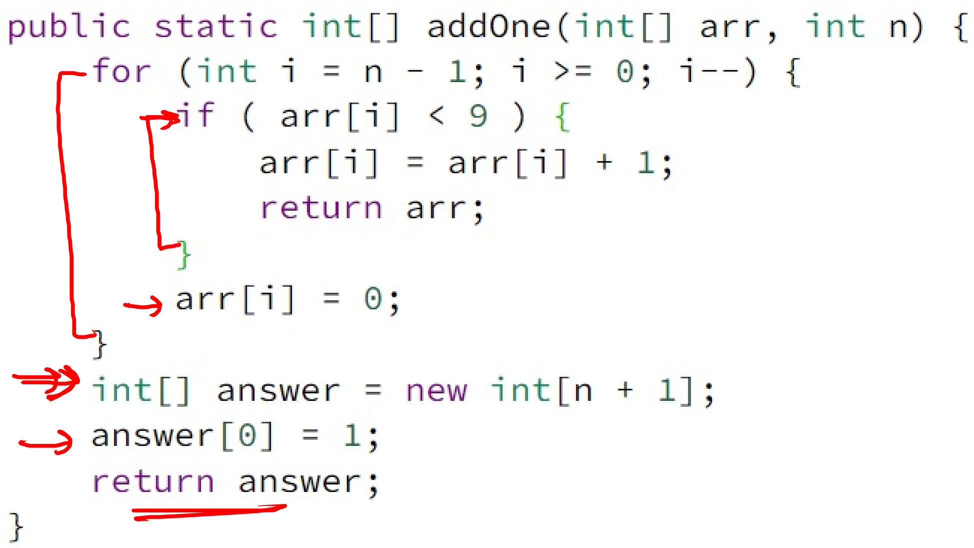
j=2 (0, 2)

j=3 (0, 3)

i=1, j=2 (1, 2)

j=3 (1, 3)

i=2, j=3 (2, 3)



```
public static int[] addOne(int[] arr, int n) {
    for (int i = n - 1; i >= 0; i--) {
        if ( arr[i] < 9 ) {
            arr[i] = arr[i] + 1;
            return arr;
        }
        arr[i] = 0;
    }
    int[] answer = new int[n + 1];
    answer[0] = 1;
    return answer;
}
```

0	1	2	3
~~9~~ 0	~~9~~ 0	~~9~~ 0	~~9~~ 0

n=4

i=3, 9<9 false

i=2, 9<9 false

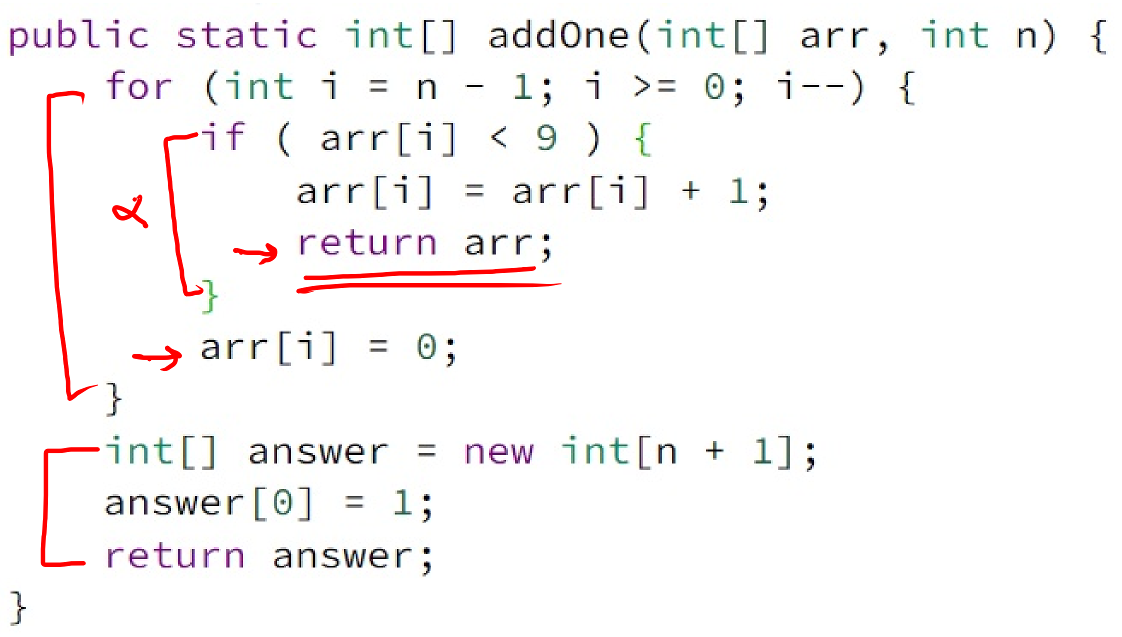
i=1, 9<9 false

i=0, 9<9 false

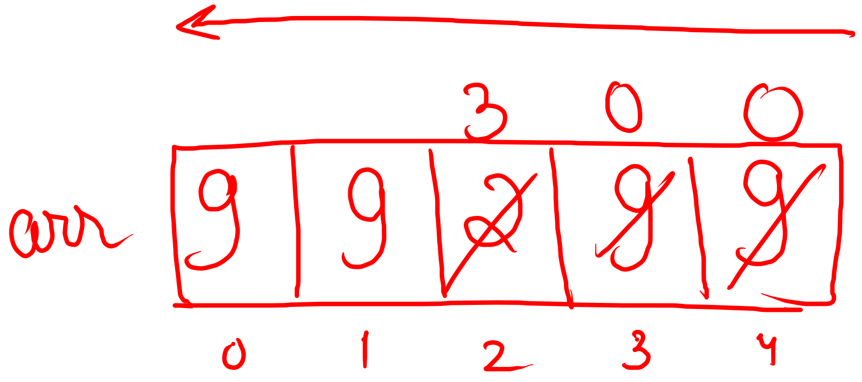
answer

1	0	0	0	0
0				

n=5



```
public static int[] addOne(int[] arr, int n) {
    for (int i = n - 1; i >= 0; i--) {
        if ( arr[i] < 9 ) {
            arr[i] = arr[i] + 1;
            return arr;
        }
        arr[i] = 0;
    }
    int[] answer = new int[n + 1];
    answer[0] = 1;
    return answer;
}
```



i=4, 9<9 false

i=3, 9<9 false

i=2, 2<9 true

arr[2] = arr[2] +1