

Ques Add one

arr

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

possibilities \Rightarrow 0 to 9

arr

(n)

0	...	n-3	n-2	n-1	
5	3	9	2 3	9 0	9 0
0	1	2	3	4	5

(Note: Red arrows point from index 4 to 3 and from index 5 to 4)

num = 539299

num+1 = ??

1 1
5 3 9 2 9 9
↓ ↓ ↓ +1
5 3 9 3 0 0

approach :- 1) Iterate from last to start

2) if val is b/w 0 to 8

[2.1) add 1 in the index
2.2) Stop the fⁿ (return)

3) if val is 9 then

3.1) update value to 0

3.2) Check for previous index

code

↓
5 2 ~~9~~ 0 0 0
↑ ↑ ↑ ↑
5 2 4 0 0 0

9 9 9 9 9
↑ ↑
1 0 0 0 0 0

```
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;
}
```

0-8
0-9

9	9	9
---	---	---

3

1	0	0	0
---	---	---	---

4

⇒ Nested loops

arr

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

pairs
↑ ↑
i j

(15)

1, 2

2, 3

3, 4

4, 5

5, 6

1, 3

2, 4

3, 5

4, 6

↑ ↑
i j

1, 4

2, 5

3, 6

↑ ↑
i j

1, 5

2, 6

↑ ↑
i j

1, 6

↑ ↑
i j

↑ ↑
i j

(N²) n = size

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
---	---	---

n=3

0 1 2

$$i=0, \text{ ans} = \underline{5} * 3 = \underline{15}$$

$$i=1, \text{ ans} = 2 * \underline{3} = 6$$

$$i=2, \text{ ans} = 2 * 5 = \underline{10}$$

$$i=0, j = \overset{\times}{0}, \overset{\checkmark}{1}, \overset{\checkmark}{2}$$

$$i=1, j = \checkmark 0, \overset{\times}{1}, \checkmark 2$$

$$i=2, j = \checkmark 0, \checkmark 1, \overset{\times}{2}$$

```
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

```

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]);
        }
    }
}

```

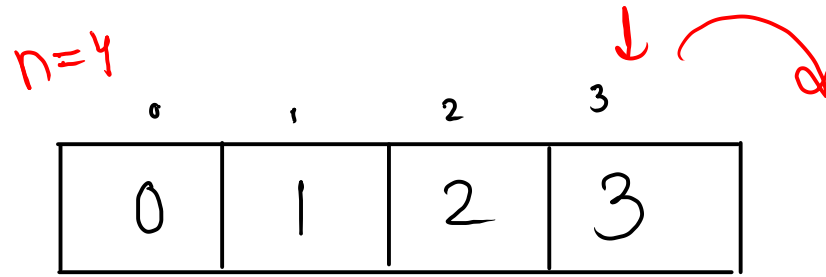


Diagram illustrating the nested loop iterations for the solve function:

- For $i=0$:
 - $j=1$ (0, 1)
 - $j=2$ (0, 2)
 - $j=3$ (0, 3)
- For $i=1$:
 - $j=2$ (1, 2)
 - $j=3$ (1, 3)
- For $i=2$:
 - $j=3$ (2, 3)

The pairs (i, j) are grouped by a large red bracket on the right, indicating the sequence of pairs generated by the nested loops.

```

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

0	0	0	0	0
--------------	---	---	---	---

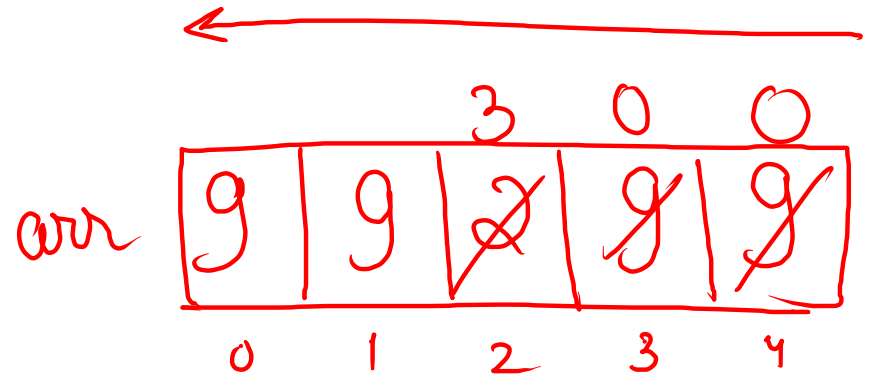
0

$n = 5$

```

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

$arr[2] = arr[2] + 1$

~~$i=1$~~
 ~~$i=0$~~