



Algorithm

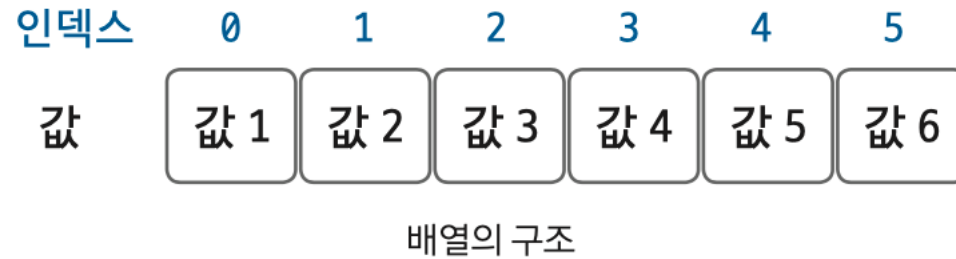
동양미래대학교 강은영

배열과 리스트

배열

배열은 메모리의 연속 공간에 값이 채워져 있는 형태의 자료구조다.

배열의 값은 인덱스를 통해 참조할 수 있으며, 선언한 자료형의 값만 저장할 수 있다.



배열과 리스트

배열의 특징

- ① 인덱스를 사용하여 값에 바로 접근할 수 있다.
- ② 새로운 값을 삽입하거나 특정 인덱스에 있는 값을 삭제하기 어렵다. 값을 삽입하거나 삭제하려면 해당 인덱스 주변에 있는 값을 이동시키는 과정이 필요하다.
- ③ 배열의 크기는 선언할 때 지정할 수 있으며, 한 번 선언하면 크기를 늘리거나 줄일 수 없다.
- ④ 구조가 간단하므로 코딩 테스트에서 많이 사용한다.

배열과 리스트

리스트

리스트는 값과 포인터를 묶는 노드라는 것을 포인터로 연결한 자료구조다.



리스트의 구조

리스트의 특징

- ① 인덱스가 없으므로 값에 접근하려면 Head 포인터부터 순서대로 접근해야 한다. 다시 말해 값에 접근하는 속도가 느리다.
- ② 포인터로 연결되어 있으므로 데이터를 삽입하거나 삭제하는 연산 속도가 빠르다.
- ③ 선언할 때 크기를 별도로 지정하지 않아도 된다. 다시 말해 리스트의 크기는 정해져 있지 않으며, 크기가 변하기 쉬운 데이터를 다룰 때 적절하다.
- ④ 포인터를 저장할 공간이 필요하므로 배열보다 구조가 복잡하다.

문제[001] 숫자의 합 구하기

시간 제한 1초 | 난이도 브론즈 II | 백준 온라인 저지 11720번

N개의 숫자가 공백 없이 써 있다. 이 숫자를 모두 합해 출력하는 프로그램을 작성하시오.

↓ 입력

1번째 줄에 숫자의 개수 N ($1 \leq N \leq 100$), 2번째 줄에 숫자 N 개가 공백 없이 주어진다.

↑ 출력

입력으로 주어진 숫자 N 개의 합을 출력한다.

배열과 리스트

예제 입력 1

1 // 숫자의 개수
1 // 공백 없이 주어진 N개의 숫자

예제 출력 1

1

예제 입력 2

5
54321

예제 출력 2

15

예제 입력 3

25
700000000000000000000000

예제 출력 3

7

예제 입력 4

11
10987654321

예제 출력 4

46

배열과 리스트

01단계 문제 분석하기

1. N의 범위가 1부터 100까지이므로 값을 int형, long형과 같은 숫자형으로 담을 수 없다.
2. 문자열 형태로 입력값을 받은 후 이를 순서대로 읽으면서 숫자형으로 변환하여 더해야 한다.

예) 입력값을 “1234”와 같이 문자열로 입력 받은 후
이를 다시 ‘1’, ‘2’, ‘3’, ‘4’와 같이 문자 배열로 변환하고,
다시 문자 배열을 1, 2, 3, 4로 변환한 다음 더해 10을 구한다.

배열과 리스트

손으로 풀어보기

- 1 숫자의 개수만큼 입력받은 값을 String형으로 저장합니다.

```
String sNum = "10987654321"
```

- 2 String형으로 입력받은 값을 char []형으로 변환합니다.


char

1	0	9	8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---	---	---	---

- 3 인덱스 0부터 끝까지 배열을 탐색하며 각 값을 정수형으로 변환하고 결괏값에 더하여 누적합니다.

char

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



배열과 리스트(Java)

코드 구현하기

배열과 리스트(Python)

코드 구현하기

파이썬에서는 배열과 리스트를 구분하지 않는다.

문제) 배열 값 중 0값을 찾아 배열 뒤로 배치시키기

(출처)

<https://www.geeksforgeeks.org/>

정수 배열에서 모든 0 값을 찾아 배열의 뒤쪽에 배치시킨다. 단, 배열 값의 상대적 순서는 유지해야 한다.

Given an array of integers, move all zeros present in the array to the end. The solutions should maintain the relative order of items in the array.

알고리즘의 시간 복잡도는 $O(n)$, 추가 공간은 $O(1)$ 로 작성하시오.

Expected time complexity is $O(n)$ and extra space is $O(1)$.

Constraints:

$1 \leq N \leq 10^5$

$0 \leq arr_i \leq 10^5$

(예) { 2, 3, 4, 0, 6 }의 배열인 경우, { 2, 3, 4, 6, 0 }을 출력한다.

배열 값 중 0값을 찾아 배열 뒤로 배치시키기

입력	N개의 0 또는 자연수 숫자 배열	
출력	주어진 숫자들의 상대적 위치를 유지하면서 0 값이 배열의 마지막에 위치한 배열	
입출력 예	<u>입력</u> { 6, 0, 8, 2, 3, 0, 4, 0, 1 }	<u>출력</u> { 6, 8, 2, 3, 4, 1, 0, 0, 0 }

[기출 관련 인터뷰]

- 2018 삼성 인턴쉽 인터뷰
- 2018 삼성 rd Bangalore 인턴쉽 인터뷰

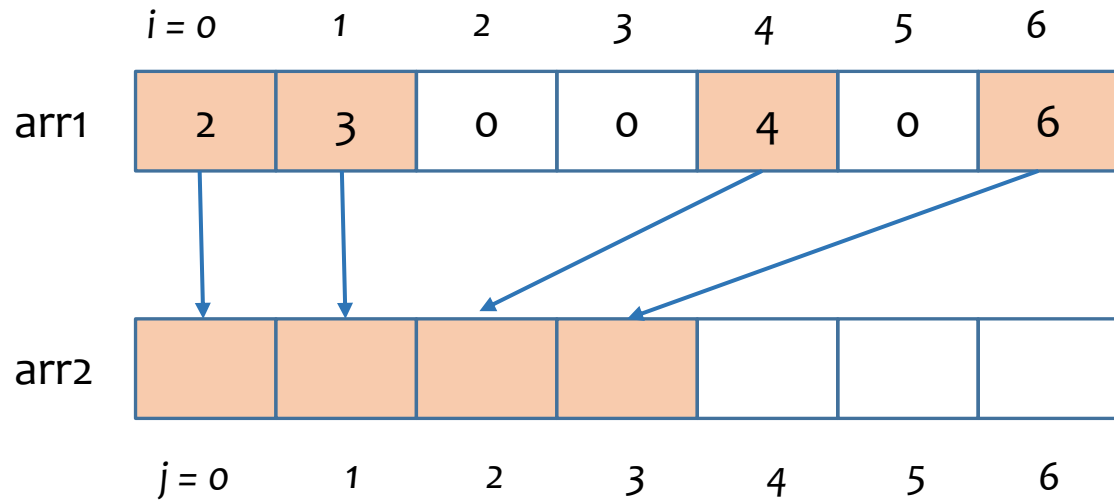
배열 값 중 0값을 찾아 배열 뒤로 배치시키기

Think-1

- 정수 배열에서 모든 0 값을 찾아 배열의 뒤쪽에 배치시킨다.
- {2, 0, 4, 0, 6}의 배열인 경우 {2, 4, 6, 0, 0}을 출력한다.
- 배열의 원소의 순서는 변경없이 유지되어야 한다.
- 시간 복잡도는 $O(n)$, 추가 공간은 $O(1)$
- **Constraints:**
 - $1 \leq N \leq 10^5$
 - $0 \leq arr_i \leq 10^5$

배열 값 중 0값을 찾아 배열 뒤로 배치시키기

Think-1



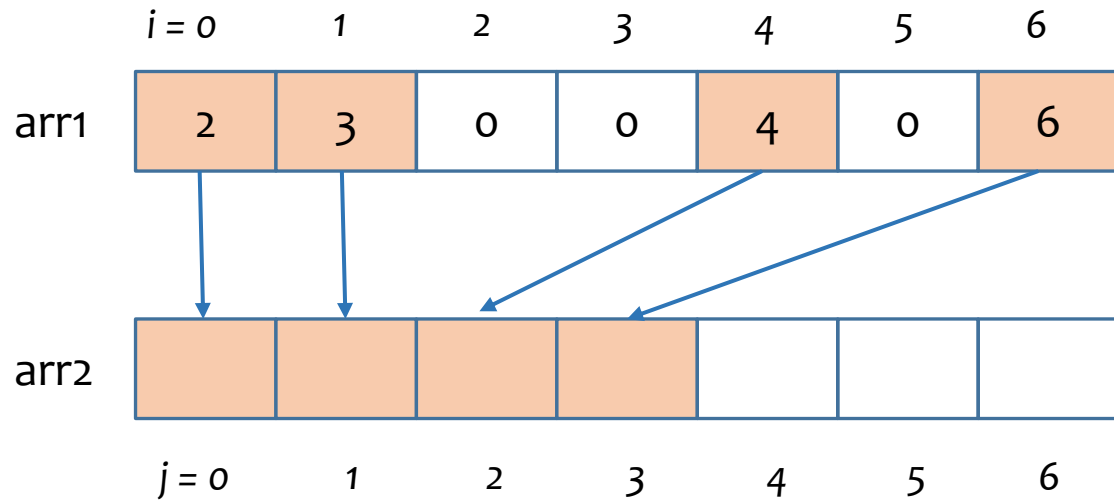
- 1 `arr1[i]`의 왼쪽에서 오른쪽으로 읽으면서 0이 아닌 값을 찾는다.
- 2 `arr[i]`의 0이 아닌 값을 `arr2[j]`에 차례로 넣는다.
- 3 `arr[i]`의 모든 값에 대해 처리가 끝나면 `arr2[j]`의 나머지에 0을 삽입



시간 복잡도 $O(n)$

추가 공간의 사용 $O(n)$???

배열 값 중 0값을 찾아 배열 뒤로 배치시키기



시간 복잡도 $O(n)$

추가 공간의 사용 $O(n)$???

시간 복잡도

```
int solve(int n)
{
    for (int i = 1; i <= n; i++) {
        //
        // 0이 아닌 arr[i] 값을 찾아 arr2[j++] 저장
        //
    }
}
```

$O(n)$

n 값이 1, 10, 100 인 경우,

// 0이 아닌 arr[i] 값을 찾아 arr2[j++] 저장 연산은
각각 1회, 10회, 100회 수행
(for 문 loop 수)

공간 복잡도

// arr2[n]이 추가로 필요

$O(n)$

배열 값 중 0값을 찾아 배열 뒤로 배치시키기

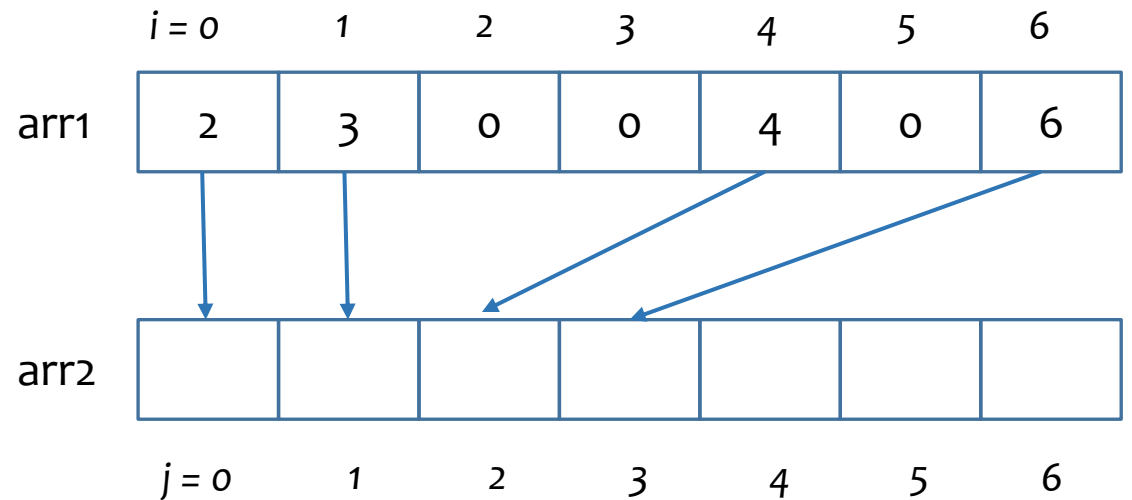
입출력	입력	출력
예	{ 6, 0, 8, 2, 3, 0, 4, 0, 1 }	{ 6, 8, 2, 3, 4, 1, 0, 0, 0 }

```
#include <stdio.h>
int main(void)
{
    int arr1[] = { 2, 3, 0, 0, 4, 0, 6 };
    int n = sizeof(arr1) / sizeof(arr1[0]);

    int arr2[n];
    int j = 0;
    for (int i = 0; i < n; i++)
    {
        if (arr1[i] != 0) {
            arr2[j] = arr1[i];
            j = j + 1;
        }
    }
    for (int i = j; i < n; i++)
        arr2[i] = 0;
```

```
for (int i = 0; i < n; i++)
    printf("%d ", arr2[i]);

return 0;
}
```



시간 복잡도 $O(n)$

추가 공간의 사용 $O(n)$

배열 값 중 0값을 찾아 배열 뒤로 배치시키기

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int arr1[] = { 2, 3, 0, 0, 4, 0, 6 };
6     int n = sizeof(arr1) / sizeof(arr1[0]);
7
8     int arr2[n];
9     int j = 0;
10
11     for (int i = 0; i < n; i++)
12     {
13         if (arr1[i] != 0) {
14             arr2[j] = arr1[i];
15             j = j + 1;
16         }
17     }
18
19     for (int i = j; i < n; i++)
20         arr2[i] = 0;
21
22     for (int i = 0; i < n; i++)
23         printf("%d ", arr2[i]);
24
25     return 0;
26 }
```

Input Goes Here.

Copy

Run

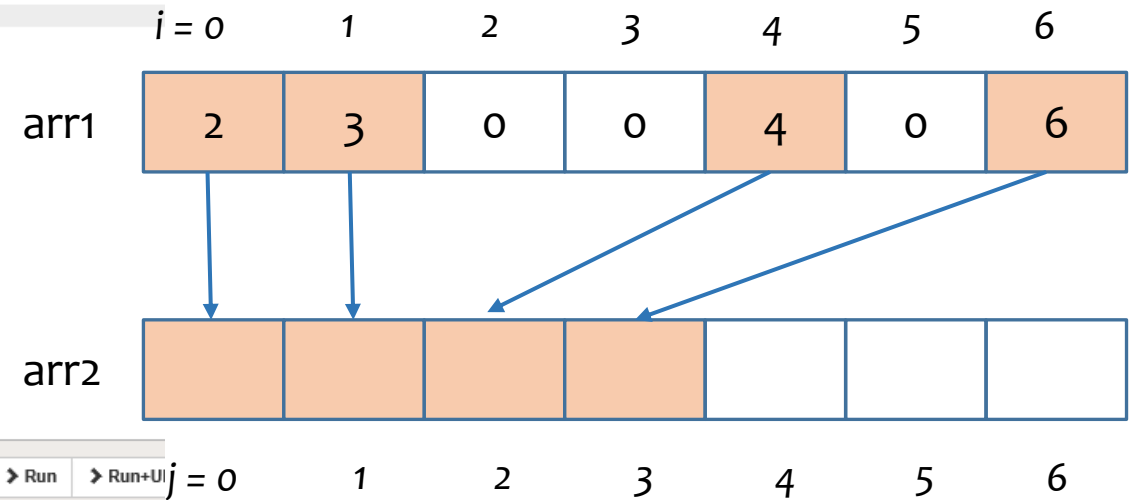
Run+UI

Time(sec) : 0

Memory(MB) : 1.5450804812622

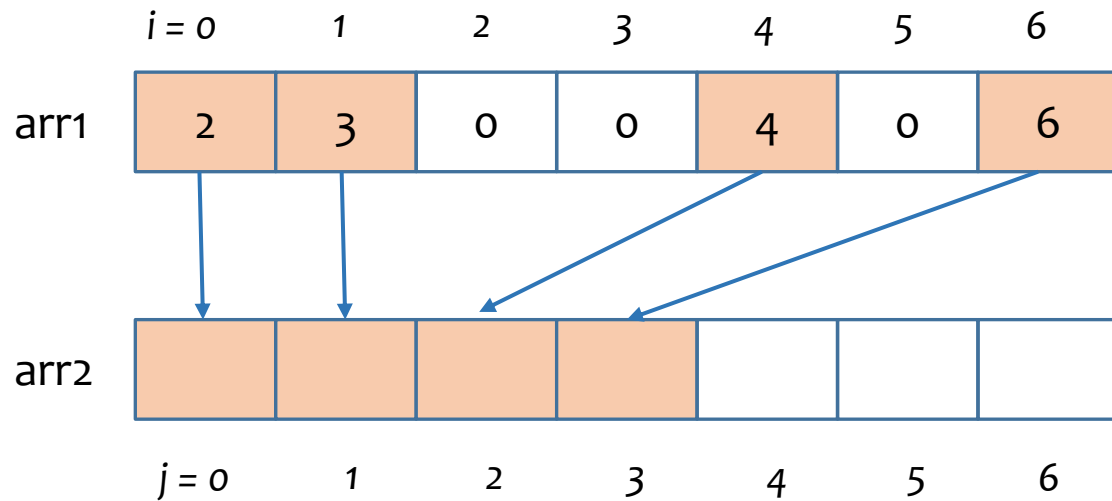
Output:

2 3 4 6 0 0 0



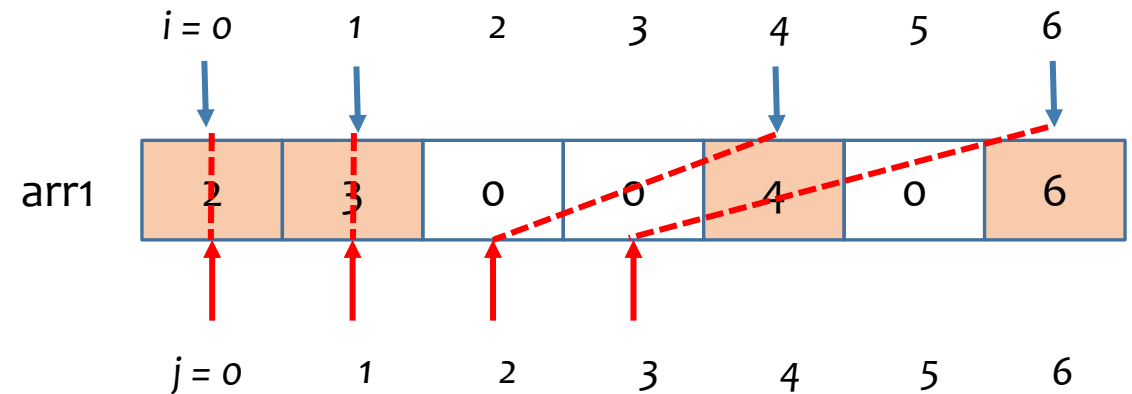
배열 값 중 0값을 찾아 배열 뒤로 배치시키기

Think-2



시간 복잡도 $O(n)$

추가 공간의 사용 $O(n)$



시간 복잡도 $O(n)$

추가 공간의 사용 $O(1)$

배열 값 중 0값을 찾아 배열 뒤로 배치시키기 (C)

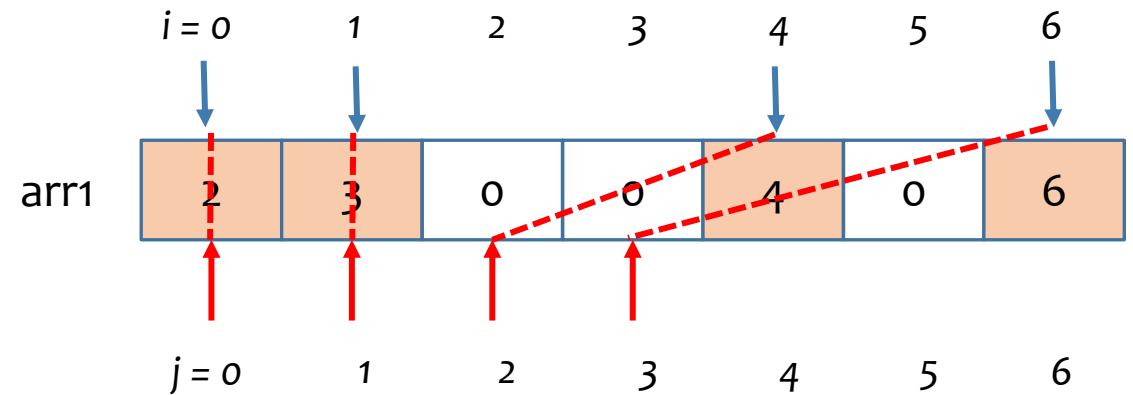
입출력 예	입력 { 6, 0, 8, 2, 3, 0, 4, 0, 1 }	출력 { 6, 8, 2, 3, 4, 1, 0, 0, 0 }
----------	-------------------------------------	-------------------------------------

```
#include <stdio.h>
int main(void)
{
    int arr1[] = { 2, 3, 0, 0, 4, 0, 6 };
    int n = sizeof(arr1) / sizeof(arr1[0]);

    int j = 0;
    for (int i = 0; i < n; i++)
    {
        if (arr1[i] != 0) {
            arr1[j] = arr1[i];
            j = j + 1;
        }
    }
    for (int i = j; i < n; i++)
        arr1[i] = 0;
```

```
for (int i = 0; i < n; i++)
    printf("%d ", arr1[i]);

return 0;
}
```

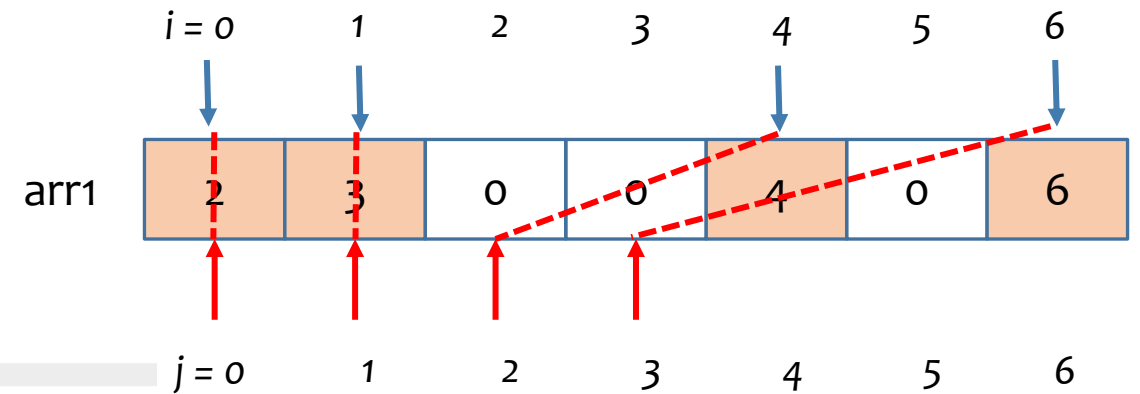


시간 복잡도 $O(n)$

추가 공간의 사용 $O(1)$

배열 값 중 0값을 찾아 배열 뒤로 배치시키기

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int arr1[] = { 2, 3, 0, 0, 4, 0, 6 };
5     int n = sizeof(arr1) / sizeof(arr1[0]);
6
7     int j = 0;
8     for (int i = 0; i < n; i++)
9     {
10         if (arr1[i] != 0) {
11             arr1[j] = arr1[i];
12             j = j + 1;
13         }
14     }
15
16     for (int i = j; i < n; i++)
17         arr1[i] = 0;
18
19     for (int i = 0; i < n; i++)
20         printf("%d ", arr1[i]);
21
22     return 0;
23 }
24
```



Input Goes Here..

Copy

Run

Run

Time(sec) : 0

Memory(MB) : 1.4394807815552

Output:

2 3 4 6 0 0 0

배열 값 중 0값을 찾아 배열 뒤로 배치시키기 (Java)

입출력 예	입력 { 6, 0, 8, 2, 3, 0, 4, 0, 1 }	출력 { 6, 8, 2, 3, 4, 1, 0, 0, 0 }
----------	-------------------------------------	-------------------------------------

```
import java.util.Arrays;
class Solution {
    public static void reorder(int[] arr) {
        int j = 0;
        for (int item : arr) {
            if (item != 0) {
                arr[j] = item;
                j++;
            }
        }
        for (int i = j; i < arr.length; i++) {
            arr[i] = 0;
        }
    }
}
```

```
public static void main(String[] args)
{
    int[] arr = { 6, 0, 8, 2, 3, 0, 4, 0, 1 };
    reorder(arr);
    System.out.println(Arrays.toString(arr));
}
```

배열 값 중 0값을 찾아 배열 뒤로 배치시키기 (Python)

입출력 예	입력 { 6, 0, 8, 2, 3, 0, 4, 0, 1 }	출력 { 6, 8, 2, 3, 4, 1, 0, 0, 0 }
----------	-------------------------------------	-------------------------------------

```
def reorder(arr, n):  
    j = 0 # Count of non-zero elements  
    for i in range(n):  
        if arr[i] != 0:  
            # here count is incremented  
            arr[j] = arr[i]  
            j+=1  
    for i in range(k, n):  
        arr[i] = 0  
# code  
arr = [6, 0, 8, 2, 3, 0, 4, 0, 1]  
n = len(arr)  
reorder(arr, n)  
print(arr)
```

소스코드 관리

- 배열 값 중 0값을 찾아 배열 뒤로 배치시키기
- 위 문제를 풀고 소스코드를 JAVA, C, C++, Python, 기타언어 중 2가지를 택하여 작성하여
- 과제(LMS(eclass.dongyang.ac.kr))에 제출해 주세요.