

# Split numbers

Given an array of ints, for example [6, 4, -3, 0, 5, -2, -1, 0, 1, -9], implement in one of the following languages

to move all positive integers to the left, all negative integers to the right, and all zeros to the middle.

You are not allowed to consume extra  $O(n)$  storage space, aka. your algorithm space complexity should be  $O(1)$ .

Your answer should be compilable and runnable, in one of the following function signatures:

- Golang:

```
func CloudmallInterview1(numbers []int) []int
```

- Python3:

```
from typing import Sequence
def cloudmall_interview1(numbers: Sequence[int]) -> Sequence[int]:
```

- Javascript:

```
/**
 * @param {number[]} numbers
 * @returns {number[]}
 */
const cloudmallInterview1 = numbers => {
  // To implement
}
```

- Java:

```
public static int[] CloudmallInterview1(int[] Numbers) {}
```

# Serialize reversed list

Given a “reversed list”, whose first tuple value is the id of string type, and second tuple value is the path to locate the id in a structured document.

For example, {"1": "bar", "2": "foo.bar", "3": "foo.foo", "4": "baz.cloudmall.com", "5": "baz.cloudmall.ai"}

Your mission is to transform the list back to a document in JSON format.

For example, a legit JSON for the above list would be:

```
{
  "bar": "1",
  "foo": {
    "bar": "2",
    "foo": "3"
  },
  "baz": {
    "cloudmall": {
      "com": "4",
      "ai": "5"
    }
  }
}
```

Your answer should be compilable and runnable, in one of the following function signatures:

- Golang:

```
func CloudmallInterview2(revList map[string]string) (jsonStr string, err error)
```

- Python3:

```
from typing import Dict
def cloudmall_interview2(revList: Dict[str, str]) -> str:
```

- Javascript:

```
/**
 * @param {Object.<string, string>} revList
 * @returns {string}
 */
const cloudmallInterview2 = revList => {
  // To implement
}
```

- Java:

```
public static String CloudmallInterview2(Map<String, String> RevList) {}
```