

# Problem 2777: Date Range Generator

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 0.00%

**Paid Only:** No

## Problem Description

Given a start date

start

, an end date

end

, and a positive integer

step

, return a generator object that yields dates in the range from

start

to

end

inclusive.

The value of

step

indicates the number of days between consecutive yielded values.

All yielded dates must be in the string format

YYYY-MM-DD

.

Example 1:

Input:

```
start = "2023-04-01", end = "2023-04-04", step = 1
```

Output:

```
["2023-04-01", "2023-04-02", "2023-04-03", "2023-04-04"]
```

Explanation:

```
const g = dateRangeGenerator(start, end, step); g.next().value // '2023-04-01' g.next().value //  
'2023-04-02' g.next().value // '2023-04-03' g.next().value // '2023-04-04'
```

Example 2:

Input:

```
start = "2023-04-10", end = "2023-04-20", step = 3
```

Output:

```
["2023-04-10", "2023-04-13", "2023-04-16", "2023-04-19"]
```

Explanation:

```
const g = dateRangeGenerator(start, end, step); g.next().value // '2023-04-10' g.next().value //  
'2023-04-13' g.next().value // '2023-04-16' g.next().value // '2023-04-19'
```

Example 3:

Input:

```
start = "2023-04-10", end = "2023-04-10", step = 1
```

Output:

```
["2023-04-10"]
```

Explanation:

```
const g = dateRangeGenerator(start, end, step); g.next().value // '2023-04-10'
```

Constraints:

```
new Date(start) <= new Date(end)
```

start

and

end

dates are in the string format

YYYY-MM-DD

0 <= The difference in days between the start date and the end date <= 1500

1 <= step <= 1000

## Code Snippets

JavaScript:

```
/**
 * @param {string} start
 * @param {string} end
 * @param {number} step
```

```

* @yields {string}
*/
var dateRangeGenerator = function* (start, end, step) {

};

/**
* const g = dateRangeGenerator('2023-04-01', '2023-04-04', 1);
* g.next().value; // '2023-04-01'
* g.next().value; // '2023-04-02'
* g.next().value; // '2023-04-03'
* g.next().value; // '2023-04-04'
* g.next().done; // true
*/

```

## TypeScript:

```

function* dateRangeGenerator(start: string, end: string, step: number) :
Generator<string> {

};

/**
* const g = dateRangeGenerator('2023-04-01', '2023-04-04', 1);
* g.next().value; // '2023-04-01'
* g.next().value; // '2023-04-02'
* g.next().value; // '2023-04-03'
* g.next().value; // '2023-04-04'
* g.next().done; // true
*/

```

## Solutions

### JavaScript Solution:

```

/**
* Problem: Date Range Generator
* Difficulty: Medium
* Tags: string
*

```

```

* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
 * @param {string} start
 * @param {string} end
 * @param {number} step
 * @yields {string}
 */
var dateRangeGenerator = function* (start, end, step) {

};

/**
 * const g = dateRangeGenerator('2023-04-01', '2023-04-04', 1);
 * g.next().value; // '2023-04-01'
 * g.next().value; // '2023-04-02'
 * g.next().value; // '2023-04-03'
 * g.next().value; // '2023-04-04'
 * g.next().done; // true
 */

```

## TypeScript Solution:

```

/**
 * Problem: Date Range Generator
 * Difficulty: Medium
 * Tags: string
 *
 * Approach: String manipulation with hash map or two pointers
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(1) to O(n) depending on approach
 */

function* dateRangeGenerator(start: string, end: string, step: number) :
Generator<string> {

};

```

```
/**
 * const g = dateRangeGenerator('2023-04-01', '2023-04-04', 1);
 * g.next().value; // '2023-04-01'
 * g.next().value; // '2023-04-02'
 * g.next().value; // '2023-04-03'
 * g.next().value; // '2023-04-04'
 * g.next().done; // true
 */
```