Problem 2634: Filter Elements from Array

Problem Information

Difficulty: Easy
Acceptance Rate: 85.50%
Paid Only: No

Problem Description

Given an integer array `arr` and a filtering function `fn`, return a filtered array `filteredArr`.

The `fn` function takes one or two arguments:

* `arr[i]` \- number from the `arr` * `i` \- index of `arr[i]`

`filteredArr` should only contain the elements from the `arr` for which the expression `fn(arr[i], i)` evaluates to a **truthy** value. A **truthy** value is a value where `Boolean(value)` returns `true`.

Please solve it without the built-in `Array.filter` method.

Example 1:

Input: arr = [0,10,20,30], fn = function greaterThan10(n) { return n > 10; } **Output:** [20,30] **Explanation:** const newArray = filter(arr, fn); // [20, 30] The function filters out values that are not greater than 10

Example 2:

Input: arr = [1,2,3], fn = function firstIndex(n, i) { return i === 0; } **Output:** [1] **Explanation:** fn can also accept the index of each element In this case, the function removes elements not at index 0

Example 3:

Input: arr = [-2,-1,0,1,2], fn = function plusOne(n) { return n + 1 } **Output:** [-2,0,1,2] **Explanation:** Falsey values such as 0 should be filtered out

Constraints:

* `0 <= arr.length <= 1000` * `-109 <= arr[i] <= 109`

Code Snippets

JavaScript:

```
/**
* @param {number[]} arr
* @param {Function} fn
* @return {number[]}
*/
var filter = function(arr, fn) {

};
```

TypeScript:

```
type Fn = (n: number, i: number) => any

function filter(arr: number[], fn: Fn): number[] {

};
```