# 1018. Binary Prefix Divisible By 5

Solved

Easy | Topics | Companies | Hint

You are given a binary array `nums` (**0-indexed**).

We define $x_i$ as the number whose binary representation is the subarray `nums[0..i]` (from most-significant-bit to least-significant-bit).

- For example, if `nums = [1,0,1]`, then $x_0 = 1$, $x_1 = 2$, and $x_2 = 5$.

Return *an array of booleans* `answer` *where* `answer[i]` *is* `true` *if* $x_i$ *is divisible by* `5`.

**Example 1:**

```
Input: nums = [0,1,1]
Output: [true,false,false]
Explanation: The input numbers in binary are 0, 01, 011; which are
0, 1, and 3 in base-10.
Only the first number is divisible by 5, so answer[0] is true.
```

**Example 2:**

```
Input: nums = [1,1,1]
Output: [false,false,false]
```

**Constraints:**

- `1 <= nums.length <= 10^5`
- `nums[i]` is either `0` or `1`.

# Python:

```
class Solution:
    def prefixesDivBy5(self, nums: List[int]) -> List[bool]:
        val = 0
        for i in range(len(nums)):
            val = ((val << 1) + nums[i]) % 5
            nums[i] = val == 0
        return nums
```

## JavaScript:

```
const prefixesDivBy5 = (nums) => {
    let val = 0;

    for (let i = 0; i < nums.length; i++) {
        val = ((val << 1) + nums[i]) % 5;
        nums[i] = val === 0;
    }

    return nums;
};
```

## Java:

```
class Solution {
    public List<Boolean> prefixesDivBy5(int[] nums) {
        List<Boolean> res = new ArrayList<>();
        int val = 0;

        for (int n : nums) {
            val = ((val << 1) + n) % 5;
            res.add(val == 0);
        }

        return res;
    }
}
```