

## 可被 5 整除的二进制前缀

给定由若干 0 和 1 组成的数组 A。我们定义 N<sub>i</sub>：从 A[0] 到 A[i] 的第 i 个子数组被解释为一个二进制数（从最高有效位到最低有效位）。

返回布尔值列表 answer，只有当 N<sub>i</sub> 可以被 5 整除时，答案 answer[i] 为 true，否则为 false。

**示例 1：**

输入：[0,1,1]

输出：[true,false,false]

**解释：**

输入数字为 0, 01, 011；也就是十进制中的 0, 1, 3 。只有第一个数可以被 5 整除，因此 answer[0] 为真。

**示例 2：**

输入：[1,1,1]

输出：[false,false,false]

**示例 3：**

输入：[0,1,1,1,1,1]

输出：[true,false,false,false,true,false]

**示例 4：**

输入：[1,1,1,0,1]

输出：[false,false,false,false,false]

```
class Solution {
public:
    vector<bool> prefixesDivBy5(vector<int>& A) {
        vector<bool> ans(A.size());
        int num = 0;
        for(int i = 0 ; i < A.size(); i++){
            // 左移一位，依次左移然后添加新位至末尾  0  01  011
            num <<= 1;
            num += A[i];
        }
    }
};
```

```
        // 只保留最后一位能被 5 整除的情况
        num %= 5;
        ans[i] = (num % 5 == 0);
    }
    return ans;
}
};
```