

## 2028. Find Missing Observations

You have observations of  $n + m$  6-sided dice rolls with each face numbered from 1 to 6.  $n$  of the observations went missing, and you only have the observations of  $m$  rolls. Fortunately, you have also calculated the **average value** of the  $n + m$  rolls.

You are given an integer array `rolls` of length  $m$  where `rolls[i]` is the value of the  $i$ th observation. You are also given the two integers `mean` and  $n$ .

Return an array of length  $n$  containing the missing observations such that the **average value** of the  $n + m$  rolls is **exactly** `mean`. If there are multiple valid answers, return *any of them*. If no such array exists, return an empty array.

The **average value** of a set of  $k$  numbers is the sum of the numbers divided by  $k$ .

Note that `mean` is an integer, so the sum of the  $n + m$  rolls should be divisible by  $n + m$ .

### Example 1:

**Input:** `rolls = [3,2,4,3]`, `mean = 4`, `n = 2`

**Output:** `[6,6]`

**Explanation:** The mean of all  $n + m$  rolls is  $(3 + 2 + 4 + 3 + 6 + 6) / 6 = 4$ .

### Example 2:

**Input:** `rolls = [1,5,6]`, `mean = 3`, `n = 4`

**Output:** `[2,3,2,2]`

**Explanation:** The mean of all  $n + m$  rolls is  $(1 + 5 + 6 + 2 + 3 + 2 + 2) / 7 = 3$ .

### Example 3:

**Input:** `rolls = [1,2,3,4]`, `mean = 6`, `n = 4`

**Output:** `[]`

**Explanation:** It is impossible for the mean to be 6 no matter what the 4 missing rolls are.

### Constraints:

- $m == \text{rolls.length}$
- $1 \leq n, m \leq 10^5$
- $1 \leq \text{rolls}[i], \text{mean} \leq 6$

## 2028. Find Missing Observations

```
/*
    Math
    Time complexity: O(m+n)
    Space complexity: O(1)
*/

typedef std::vector<int> vi;
class Solution {
public:
    vi missingRolls(vi& rolls,int mean,int n){
        int m=rolls.size();
        int rolls_sum=std::accumulate(rolls.begin(),rolls.end(),0);
        int missing_sum=mean*(m+n)-rolls_sum;

        if(missing_sum>6*n || missing_sum<n) return {};

        int cell_val=missing_sum/n;
        int rem=missing_sum%n;

        /*
            ans={cell_val+1}*rem+{cell_val}*(n-rem);

            vi ans;
            for(int i=1;i<=n-rem;++i) ans.push_back(cell_val);
            for(int i=1;i<=rem;++i) ans.push_back(cell_val+1);
        */
        vi ans(n,cell_val);
        std::fill(ans.begin(),ans.begin()+rem,cell_val+1);

        return ans;
    }
};
```