

找出缺失的观测数据

现有一份 $n + m$ 次投掷单个 **六面** 骰子的观测数据，骰子的每个面从 1 到 6 编号。观测数据中缺失了 n 份，你手上只拿到剩余 m 次投掷的数据。幸好你有之前计算过的这 $n + m$ 次投掷数据的 **平均值**。

给你一个长度为 m 的整数数组 `rolls`，其中 `rolls[i]` 是第 i 次观测的值。同时给你两个整数 `mean` 和 n 。

返回一个长度为 n 的数组，包含所有缺失的观测数据，且满足这 $n + m$ 次投掷的 **平均值** 是 `mean`。如果存在多组符合要求的答案，只需要返回其中任意一组即可。如果不存在答案，返回一个空数组。

k 个数字的 **平均值** 为这些数字求和后再除以 k 。

注意 `mean` 是一个整数，所以 $n + m$ 次投掷的总和需要被 $n + m$ 整除。

示例 1：

输入：`rolls = [3,2,4,3]`, `mean = 4`, $n = 2$

输出：`[6,6]`

解释：所有 $n + m$ 次投掷的平均值是 $(3 + 2 + 4 + 3 + 6 + 6) / 6 = 4$ 。

示例 2：

输入：`rolls = [1,5,6]`, `mean = 3`, $n = 4$

输出：`[2,3,2,2]`

解释：所有 $n + m$ 次投掷的平均值是 $(1 + 5 + 6 + 2 + 3 + 2 + 2) / 7 = 3$ 。

示例 3：

输入：`rolls = [1,2,3,4]`, `mean = 6`, $n = 4$

输出：`[]`

解释：无论丢失的 4 次数据是什么，平均值都不可能是 6。

示例 4：

输入：`rolls = [1]`, `mean = 3`, $n = 1$

输出：`[5]`

解释：所有 $n + m$ 次投掷的平均值是 $(1 + 5) / 2 = 3$ 。

```
class Solution:
    def missingRolls(self, rolls: List[int], mean: int, n: int) -> List[int]:
        #计算总和
        sum_all=(n+len(rolls))*mean
        #计算丢失部分的和
        sum_loss=sum_all-sum(rolls)
        if sum_loss<0:
            return []
        #丢失部分均值
        loss_avg=sum_loss//n
        #丢失部分余数
        loss_rem=sum_loss%n
        if loss_avg<=6 and loss_rem==0:
            return [loss_avg]*n
        elif loss_avg>=6 or loss_avg==0 and loss_rem>=0:
            return []
        elif loss_avg<6 and loss_rem>0:
            return [loss_avg+1]*loss_rem+[loss_avg]*(n-loss_rem)
```