

划分

Temur (帖木儿) 和他的助手 Ulug'bek (兀鲁伯) 正在为“乌兹别克斯坦达人秀”准备一个魔术表演。这个魔术的核心在于 Temur 需要解决以下**划分问题**：将一批给定的正整数划分成 K 个非空组，使得每个组中的元素之和相等。换言之，这批整数中的每个整数都必须恰好被分配到 K 个组中的一个，且每个组中的元素之和必须相等。例如，若给定的这批整数为 $[2, 1, 6, 4, 5]$ 且 $K = 3$ ，一种合法的划分可以是 $[2, 4]$ 、 $[1, 5]$ 和 $[6]$ 。此时，每个组中的元素之和均为 6。

魔术表演流程如下：

- Ulug'bek 和 Temur 分别进入不同的隔间，彼此无法交流。
- 裁判向 Ulug'bek 提供一个长度为 N 的正整数数组 A （即 $A[0], A[1], \dots, A[N-1]$ ），其中每个元素的值都在 1 到 M 之间（包括 1 和 M ）。裁判同时把 K 的值告知 Ulug'bek。
- Ulug'bek 选择至多 $K - 1$ 个整数（不需要互不相同），**作为新元素添加到数组 A 中**。每个新添加的整数也必须介于 1 到 M 之间（包括 1 和 M ）。
- 裁判将 Ulug'bek 选择的整数添加到原数组。这个扩展后的数组将按**非递减顺序排序**，并与 K 的值一并交给 Temur。
- Temur 必须对这个扩展且已排序的数组求解划分问题。

你的任务是为 Temur 和 Ulug'bek 设计并实现策略。可以证明，在给定约束条件下，无论裁判提供怎样的数组 A ，均存在一种策略使他们能够成功解决划分问题。

实现细节

你要实现两个函数。

为 Ulug'bek 实现的函数为：

```
std::vector<int> add_numbers(std::vector<int> A, int K, int M)
```

- A ：长度为 N 的数组，表示交给 Ulug'bek 的原始数组。
- K ：要求的划分组数；注意，Ulug'bek 可以至多添加 $K - 1$ 个整数到数组中。
- M ：每个原始和新增整数允许的最大值。
- 对于每个测试用例，该函数恰好被调用一次。

该函数应返回一个数组 C ，包含 Ulug'bek 希望添加到原始数组中的整数。设 S 为数组 C 的长度。

- S 不能超过 $K - 1$ 。

- C 中的每个元素的值必须介于 1 到 M 之间（包括 1 和 M ）。

为 Temur 实现的函数为：

```
std::vector<int> find_partition(std::vector<int> B, int K)
```

- B ：长度为 $N + S$ 的整数数组，包含来自 A 的原始整数以及 Ulug'bek 添加的整数。数组 B 的元素已按非递减顺序排序。
- K ：要求的划分组数。
- 对于每个测试用例，该函数恰好被调用一次。

该函数应返回一个数组 P ，给出将 B 划分成 K 个不相交组的方案。

- P 的长度应为 $N + S$ 。
- 对于满足 $0 \leq j < N + S$ 的每个 j ， $P[j]$ 表示 $B[j]$ 所属的组的编号。
- 组的编号应为从 0 到 $K - 1$ ，即对于每个 $0 \leq j < N + S$ ，必须满足 $0 \leq P[j] < K$ 。
- 对于 0 到 $K - 1$ （包含边界）之间的每个 i ，必须存在至少一个 j （ $0 \leq j < N + S$ ）使得 $P[j] = i$ 。
- K 个组中，每个组所分配的元素之和必须相同。

在实际评测中，调用上述函数的程序将运行**恰好两次**。

- 在程序的第一次运行中：
 - `add_numbers` 被恰好调用一次。
 - 评测系统将由所返回的数组计算出数组 B 。
- 在程序的第二次运行中：
 - `find_partition` 被恰好调用一次。

约束条件

- $3 \leq N \leq 100\,000$
- $2 \leq K \leq 100\,000$
- $K \leq N$
- $1 \leq M \leq 10^9$
- 对于满足 $0 \leq i < N$ 的每个 i ，均有 $1 \leq A[i] \leq M$ 。

子任务

子任务	分数	额外的约束条件
1	5	$N = 3$
2	4	$M = 1$
3	7	$M \leq 2$
4	10	$N \leq 10$
5	7	$K = 2$
6	12	$K \leq 3$
7	17	$K \leq 10$
8	20	$K \leq 100$
9	18	没有额外的约束条件。

例子

考虑以下调用：

```
add_numbers([8, 2, 9, 6, 1, 5, 5], 3, 9)
```

在这个例子中， $A = [8, 2, 9, 6, 1, 5, 5]$ 。我们要将这些数字划分成 $K = 3$ 个组。Ulug'bek 可以添加在 1 到 $M = 9$ 之间的数字。

该函数可以返回 $[5, 4]$ ，表示 Ulug'bek 决定添加 $S = 2$ 个新整数：5 和 4。两个元素均在 1 到 $M = 9$ 之间，因此这是符合要求的。

扩展后的数组经排序后，通过以下函数调用交给 Temur：

```
find_partition([1, 2, 4, 5, 5, 5, 6, 8, 9], 3)
```

我们可以将该数组中的整数划分为这三个组： $[1, 5, 9]$ 、 $[2, 5, 8]$ 和 $[4, 5, 6]$ 。可以看到，每个组中的元素之和均等于 15。该函数应返回数组 $[0, 1, 2, 0, 1, 2, 2, 1, 0]$ 来给出此划分结果。

评测程序示例

输入格式：

```
N K M  
A[0] A[1] ... A[N-1]
```

输出格式：

在 `add_numbers` 调用完成后，评测程序示例将：

- 计算排序后的数组 B 。
- 按以下格式输出数组 C 和 B ，**并在最后跟着一个空行**：

```
S  
C[0] C[1] ... C[S-1]  
B[0] B[1] ... B[N+S-1]
```

在 `find_partition` 调用完成后，评测程序示例输出：

```
L  
P[0] P[1] ... P[L-1]
```

其中， L 为 `find_partition` 返回的数组 P 的长度。