

分組問題

帖木兒和他的助手兀魯伯正在為《烏茲別克斯坦達人秀》節目準備一場魔術表演。這個魔術的核心在於帖木兒要解決以下這個**分組問題**：將一組給定的正整數分成 K 個非空組，且每個組內元素的和都必須相等。換言之，集合中的每個整數都必須被分配到 K 個非空組別內的其中一個，且每個組別內的元素的和必須相等。例如，若給定的集合為 $[2, 1, 6, 4, 5]$ 和 $K = 3$ ，則一個有效的劃分方式為 $[2, 4]$ 、 $[1, 5]$ 以及 $[6]$ 。在這種情況下，每個組別的中元素的和均恰好為6。

此魔術的表演方式如下：

- 兀魯伯與帖木兒分別進入獨立的隔間，彼此無法溝通。
- 一位評審向兀魯伯提供一個陣列 A ，其中包含 N 個正整數 $A[0], A[1], \dots, A[N - 1]$ ，每個數值介於 1 至 M 之間（含兩端）。該評審同時也向兀魯伯提供數值 K 。
- 兀魯伯選擇至多 $K - 1$ 個整數（不一定互不相同），將其 **作為新元素加入** 陣列 A 。每個新增的整數也必須介於 1 至 M 之間（包含兩端）。
- 評審將兀魯伯選出的整數新增至原始陣列中。接著，此擴充後的陣列將被**按非遞減順序排序**，並連同 K 的值一併提供給帖木兒。
- 帖木兒必須針對此擴充且已排序的陣列，解決其分區問題。

你的任務是為帖木兒和兀魯伯設計並實踐設定策略。可以證明，在給定的限制下，無論評審提供的陣列 A 是甚麼樣的，都存在著一套能夠讓他們成功解決分組問題的策略。

實作細節

你應該要編寫兩個子程序。

你應該要為**兀魯伯**編寫的子程序如下：

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

- A ：長度為 N 的陣列，代表提供給兀魯伯的原始陣列。
- K ：目標分組數量；請注意，兀魯伯可能會向陣列中加入最多 $K - 1$ 個整數。
- M ：允許每個原始整數及新增整數的最大值。
- 此程式針對每個測試案例只會被呼叫一次。

程式傳回一個陣列 C ，其中包含兀魯伯欲新增至原始陣列中的整數。設 S 表示陣列 C 的長度。

- S 必須不大於 $K - 1$ 。
- C 每個元素必須在1與 M 之間（含兩端）。

您應該為 **帖木兒** 編寫的子程序如下：

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

- B ：一個長度為 $N + S$ 的陣列，其中包含來自 A 的原始整數以及兀魯伯新增的整數。陣列 B 的元素按照非遞減順序排序。
- K ：目標分組數。
- 此程序針對每個測試案例僅被呼叫一次。

子程序回傳一個陣列 P ，表示將 B 分為 K 個互不相交的分組。

- P 的長度應為 $N + S$ 。
- 對於每個 j 且滿足 $0 \leq j < N + S$ 的情況， $P[j]$ 表示 $B[j]$ 屬於的分組。
- 分組應從 0 編號至 $K - 1$ ，這意味著對於每個 $0 \leq j < N + S$ ， $0 \leq P[j] < K$ 必須成立。
- 對於每個位於 0 至 $K - 1$ 之間（含兩端）的 i ，必須至少有一個元素 j ($0 \leq j < N + S$)，且須滿足 $P[j] = i$ 。
- 分配給 K 個群組的元素總和必須完全相同。

在實際評分過程中，調用上述程序的方案將被執行**兩次**。

- 在首次執行期間：
 - `add_numbers` 會被呼叫恰好一次。
 - 評分系統會處理該回傳的陣列，以計算陣列 B 。
- 在第二次執行期間：
 - `find_partition` 會被呼叫恰好一次。

約束條件

- $3 \leq N \leq 100\,000$
- $2 \leq K \leq 100\,000$
- $K \leq N$
- $1 \leq M \leq 10^9$
- 對於每個 i ，皆有 $1 \leq A[i] \leq M$ ，且滿足 $0 \leq i < N$ 。

子任務

子任務	分數	其他約束條件
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	沒有其他約束條件

範例

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

這裡範例中， $A = [8, 2, 9, 6, 1, 5, 5]$ 。我們希望將這些數字分成 $K = 3$ 組。兀魯伯可以添加1與 $M = 9$ 之間的數字。

此程式可傳回 $[5, 4]$ ，表示烏魯格貝克決定增加 $S = 2$ 個新增的整數：5及4。由於這兩個元素均在1與 $M = 9$ 之間，因此是有效的。

接著對擴展後的陣列進行排序，並透過以下調用將其傳遞給 帖木兒：

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

我們可以將此陣列中的整數分為以下三組： $[1, 5, 9]$ 、 $[2, 5, 8]$ 以及 $[4, 5, 6]$ 。請注意，這三組的每組總和皆等於15。為了提交此分組結果，該子程序應傳回陣列 $[0, 1, 2, 0, 1, 2, 2, 1, 0]$ 。

樣例評分程式

輸入格式：

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

輸出格式：

在呼叫`add_numbers`完成後，樣例評分器會：

- 計算出已排序的陣列 B 。
- 用以下格式輸出陣列 C 和 B ，**後面接一行空行**：

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

在調用`find_partition`完成後，評分程式會輸出：

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

此處， L 是`find_partition`所回傳的陣列 P 的長度。