

分割問題 (Partition)

帖木兒 (Temur) 和他的助手兀魯伯 (Ulug'bek) 正在為《烏茲別克達人秀》準備一個魔術表演秀。這個表演秀圍繞著帖木兒解決以下**分割問題**：將給定的一個正整數集劃分成 K 組，使得每組的整數總和相等。換句話說，正整數集裡的每個整數都必須被分配到 K 個組中的某一組，並且每個組內的總和必須相等。例如，如果給定的正整數集為 $[2, 1, 6, 4, 5]$ 且 $K = 3$ ，則一個有效的分割為： $[2, 4]$ 、 $[1, 5]$ 和 $[6]$ 。在這種情況下，每組的整數總和恰好為 6。

這個魔術表演秀的進行方式如下：

- 兀魯伯和帖木兒進入不同的隔間，且無法互相溝通。
- 一名評審給兀魯伯一個包含 N 個整數的陣列 A ，即 $A[0], A[1], \dots, A[N - 1]$ ，其中每個值的範圍由 1 到 M ，以及 K 的值。
- 兀魯伯最多選擇 $K - 1$ 個（不一定相異）整數作為添加到陣列 A 的新整數。其中每個新增的整數也必須介於 1 和 M 之間（含 1 和 M ）。
- 評審將兀魯伯所選的整數加入原來的陣列 A 。然後，將這個擴充的陣列**按非遞減順序排序**，並將其與 K 的值一起交給帖木兒。
- 接下來，帖木兒必須為這個已被擴充、且排序的陣列解決分割問題。

你的任務是為帖木兒和兀魯伯設計並實作一策略。事實上可以證明，在給定的條件下，評審所給的任一陣列 A ，都存在一個策略，使帖木兒和兀魯伯可以成功地完成分割。

實作細節

您應該實作兩個程序。

對**兀魯伯**您應實作下列程序：

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

- A ：長度為 N 的陣列，表示給帖木兒的原始陣列。
- K ：目標組數；請注意兀魯伯可以加入至多 $K - 1$ 個整數到陣列 A 中。
- M ：每個原始整數和新增整數所允許的最大值。
- 對於每一筆測試，此程序只會被呼叫一次。

該程序應回傳一個陣列 C ，其中包含兀魯伯想要新增到原始陣列中的整數。令 S 表示陣列 C 的長度。

- S 必須最多為 $K - 1$ 。
- C 中的每個整數都必須介於 1 和 M 之間（含 1 和 M ）。

對帖木兒，您應該實作下列程序：

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

- B ：長度為 $N + S$ 的陣列，包含來自 A 的原始整數和兀魯伯所新增的整數。陣列 B 的整數依非遞減順序排序。
- K ：目標組數。
- 對於每個測試，此程序只會被呼叫一次。

此程序應傳回一個陣列 P ，表示將 B 分成 K 個不相交的組。

- P 的長度應為 $N + S$ 。
- 對每個 j 使得 $0 \leq j < N + S$ ， $P[j]$ 表示 $B[j]$ 所屬的組。
- 各組的編號從 0 到 $K - 1$ ，這表示對於每個 j ， $0 \leq j < N + S$ ，必須滿足 $0 \leq P[j] < K$ 。
- 對於介於 0 和 $K - 1$ 之間（含 0 和 $K - 1$ ）的每個 i ，必須至少有一個整數 j ($0 \leq j < N + S$) 使得 $P[j] = i$ 。
- 分配給每個組（ K 組）的整數總和必須相同。

在實際評分過程中，有一呼叫上述程序的程式將被執行**恰好兩次**。

- 在該程式第一次被執行時：
 - `add_numbers` 程序只被呼叫恰好一次。
 - 回傳的陣列將由評分系統處理並計算陣列 B 。
- 在該程式第二次被執行時：
 - `find_partition` 程序只被呼叫恰好一次。

條件限制

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

子任務

子任務	分數	額外條件限制
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	無額外限制。

範例

請考慮以下呼叫：

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

在這個例子中， $A = [8, 2, 9, 6, 1, 5, 5]$ 。我們要將這些數字分成 $K = 3$ 組。兀魯伯可以對 1 到 $M = 9$ 之間的數字進行加法運算。

該程序可以回傳 $[5, 4]$ ，這表示兀魯伯決定增加 $S = 2$ 個新整數：5 和 4。這是有效的，因為這兩個整數都在 1 和 $M = 9$ 之間。

然後對擴展後的陣列進行排序，並透過以下呼叫將其傳遞給帖木兒：

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

我們可以將這個陣列中的整數分成以下三組： $[1, 5, 9]$ 、 $[2, 5, 8]$ 和 $[4, 5, 6]$ 。請注意，每一組的總和等於 15。要回報此分割，該程序應傳回陣列 $[0, 1, 2, 0, 1, 2, 2, 1, 0]$ 。

範例評分程式

輸入格式：

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

輸出格式：

在呼叫 `add_numbers` 完成之後，範例評分程式：

- 計算排序後的陣列 B 。
- 以下列格式列印陣列 C 和 B ，後面跟著一個空白行：

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

在呼叫 `find_partition` 完成後，範例評分程式輸出：

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

這裡， L 是 `find_partition` 回傳的陣列 P 的長度。