排列 (Permutation)

法老們利用相對速度以及星球重力來加速他們的太空船。 假設太空船會經過軌道速度依序為 $p[0], p[1], \ldots, p[n-1]$ 的 n 個星球。 對每個星球，這些法老科學家們可選擇是否要利用該星球來加速太空船。為了節省能源，在利用軌道速度為 $p[i]$ 的星球加速後，太空船就不能以任何軌道速度 $p[j] < p[i]$ 的星球來加速了。換言之，被選擇的星球之軌道速度會形成 $p[0], p[1], \ldots, p[n-1]$ 的一個「遞增子序列」。一個 p 的子序列是由 p 中刪除零個以上的元素所得到的序列。舉例來說，$[0]$、$[]$、$[0, 2]$ 以及 $[0, 1, 2]$ 都是 $[0, 1, 2]$ 的子序列，但 $[2, 1]$ 不是。

這些科學家們已得知共有 k 種不同的星球集合可被用來加速太空船，但他們遺失了所有軌道速度的紀錄 (甚至連 n 的值也遺失了)。然而，他們記得 $(p[0], p[1], \ldots, p[n-1])$ 是一個 $0, 1, \ldots, n-1$ 的排列 (permutation)。一個排列是包含 0 到 $n-1$ 每個整數恰好一次的序列。 你的任務是找到一個長度足夠小的排列 $p[0], p[1], \ldots, p[n-1]$。

你需要為 q 艘不同的太空船解決這個問題。對太空船 i，你會拿到一個整數 k_i，表示有幾種不同的星球集合可以被用來加速太空船。你的任務是找到一個長度 n_i 夠小的軌道速度序列，使得從中恰可選出 k_i 個遞增的軌道速度子序列。

實作細節 (Implementation details)

你應實作下列函式:

```
int[] construct_permutation(int64 k)
```

- k: 預期的遞增子序列個數。
- 此函式應回傳一 n 個元素的陣列，每個元素至少為 0，至多為 $n-1$。
- 此回傳的陣列必須是一合法的排列，且包含恰好 k 個遞增子序列。
- 此函式會被呼叫恰好 q 次。每次呼叫應被視為一個個別的情境。

限制 (Constraints)

- $1 \leq q \leq 100$
- $2 \leq k_i \leq 10^{18}$ (對所有 $0 \leq i \leq q-1$)

子任務 (Subtasks)

1. (10 points) $2 \le k_i \le 90$ (對所有 $0 \le i \le q-1$)。如果你找到的所有排列長度不超過 90 且符合要求，你會得到 10 分；否則會得到 0 分。
2. (90 points) 無額外限制。 對於此子任務，令 m 為所有情境下你所使用的最大排列長度。則你的分數會依下表計算：

Condition	Score
$m \le 90$	90
$90 < m \le 120$	$90 - \frac{(m-90)}{3}$
$120 < m \le 5000$	$80 - \frac{(m-120)}{65}$
$m > 5000$	0

範例 (Example)

Example 1

考慮以下呼叫：

```
construct_permutation(3)
```

此函式應回傳一包含恰好 3 個遞增子序列之排列。一個可能的答案為 $[1, 0]$，包含 $[]$ (空的子序列)、$[0]$ 以及 $[1]$ 為其中的遞增子序列。

Example 2

考慮以下呼叫：

```
construct_permutation(8)
```

此函式應回傳一包含恰好 8 個遞增子序列之排列。一個可能的答案為 $[0, 1, 2]$。

範例評分程式 (Sample grader)

此範例評分程式用以下格式讀取輸入：

- line 1: q
- line $2 + i$ $(0 \le i \le q-1)$: k_i

對每個 k_i，此範例評分程式輸出單一一行，包含`construct_permutation` 的回傳值；若有錯誤發生則會輸出錯誤訊息。