排列（Permutation）

法老们利用行星的引力来加速飞船。假设飞船将依次以 $p[0], p[1], \ldots, p[n-1]$ 的轨道速度飞掠 n 颗行星。飞掠每颗行星时，法老科学家可以选择是否利用它来加速飞船。为了节省能量，当飞船以轨道速度 $p[i]$ 飞掠一颗行星并完成加速后，它将不能再在以轨道速度 $p[j] < p[i]$ 飞掠行星时进行加速。也就是说，选择用来加速的行星构成 $p[0], p[1], \ldots, p[n-1]$ 的一个**递增子序列**。p 的子序列是从 p 中删除零个或多个元素得到的序列。例如，$[0]$、$[]$、$[0, 2]$ 和 $[0, 1, 2]$ 是 $[0, 1, 2]$ 的子序列，但 $[2, 1]$ 不是。

科学家已经确认，总共有 k 种方案来选择行星对飞船进行加速，但是他们弄丢了轨道速度的记录信息（甚至包括 n 的大小）。不过他们记得 $(p[0], p[1], \ldots, p[n-1])$ 是 $0, 1, \ldots, n-1$ 的一个排列。这里的排列是包含从 0 到 $n-1$ 每个整数恰好一次的序列。你的任务是找出一个长度尽量小且符合要求的排列 $(p[0], p[1], \ldots, p[n-1])$。

你要对 q 艘不同的飞船来解决该问题。对每艘飞船 i，你会得到一个整数 k_i，表示选择行星加速飞船的不同方案数。你的任务是找出长度 n_i 足够小的轨道速度序列，使得从中恰好可以选出 k_i 个轨道速度递增的行星子序列。

实现细节

你要实现以下函数：

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

- k 是应有的递增子序列的数量。
- 该函数要返回有 n 个元素的数组，每个元素是 0 到 $n-1$ 之间（包括 0 和 $n-1$）的数。
- 返回的数组必须是恰好有 k 个递增子序列的合法排列。
- 该函数总共被调用 q 次。每次调用被视为一个独立的场景。

约束条件

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

子任务

1. （10 分）$2 \le k_i \le 90$（对所有 $0 \le i \le q-1$）。如果你给出的所有排列长度至多为 90 且结果正确，你将获得 10 分，否则获得 0 分。

2. （90 分）没有额外的约束条件。对该子任务，令 m 为你在所有场景中给出的排列的最大长度，则你的得分按下表来计算：

条件	得分
$m \leq 90$	90
$90 < m \leq 120$	$90 - \dfrac{(m - 90)}{3}$
$120 < m \leq 5000$	$80 - \dfrac{(m - 120)}{65}$
$m > 5000$	0

例子

例 1

考虑以下调用：

```
construct_permutation(3)
```

该函数应该返回一个恰好有 3 个递增子序列的排列。一种可能的答案是 $[1, 0]$，它的递增子序列有 $[]$（空的子序列）、$[0]$ 和 $[1]$。

例 2

考虑以下调用：

```
construct_permutation(8)
```

该函数应该返回一个恰好有 8 个递增子序列的排列。一种可能的答案是 $[0, 1, 2]$。

评测程序示例

评测程序示例按以下格式读取输入：

- 第 1 行：q
- 第 $2 + i$ 行（$0 \leq i \leq q - 1$）：k_i

评测示例程序对每个 k_i 打印一行，包含对应 `construct_permutation` 调用的返回值。如果出错则打印错误信息。