駭入！ (hack)

在 Codeforces 比賽進行一小時後，你注意到同房間的另一位參賽者使用 `unordered_set` 解決了一個問題。是時候發動駭客攻擊了！

你知道 `unordered_set` 使用了一個有 n 個桶的哈希表，這些桶編號從 0 到 $n-1$。不幸的是，你不知道 n 的值，並希望找出它。

當你將一個整數 x 插入哈希表時，它會被插入到第 $(x \bmod n)$ 個桶中。如果在插入前該桶已有 b 個元素，則會發生 b 次哈希碰撞。

通過向交互器提供 k 個不同的整數 $x[0], x[1], \ldots, x[k-1]$，你可以得知在創建包含這些數字的 `unordered_set` 時發生的總哈希碰撞次數。然而，每次查詢中提供 k 個整數將花費 k 的成本。

例如，若 $n = 5$，向交互器提供 $x = [2, 15, 7, 27, 8, 30]$ 將總共導致 4 次碰撞：

操作	新增碰撞	桶的狀態
初始狀態	$-$	$[], [], [], [], []$
插入 $x[0] = 2$	0	$[], [], [2], [], []$
插入 $x[1] = 15$	0	$[15], [], [2], [], []$
插入 $x[2] = 7$	1	$[15], [], [2, 7], [], []$
插入 $x[3] = 27$	2	$[15], [], [2, 7, 27], [], []$
插入 $x[4] = 8$	0	$[15], [], [2, 7, 27], [8], []$
插入 $x[5] = 30$	1	$[15, 30], [], [2, 7, 27], [8], []$

注意，交互器通過按順序將元素插入初始為空的 `unordered_set` 來創建哈希表，且每次查詢都會創建一個新的空 `unordered_set`。換言之，所有查詢都是獨立的。

你的任務是在總成本不超過 $1\ 000\ 000$ 的情況下，找出桶的數量 n。

實作細節

你應實作以下函式：

```
int hack()
```

- 此函式應回傳一個整數——隱藏的 n 值。
- 對於每個測試案例，評分程式可能會多次呼叫此函式。每次呼叫應視為獨立的新情境。

在此函式中，你可以呼叫以下函式：

```
long long collisions(std::vector<long long> x)
```

- x: 一個由不同數字組成的陣列，其中每個 $x[i]$ 滿足 $1 \le x[i] \le 10^{18}$。
- 此函式回傳將 x 的元素插入 `unordered_set` 時產生的碰撞次數。
- 此函式可被多次呼叫。在單次 `hack()` 呼叫中，所有 `collisions()` 呼叫的 x 長度總和不得超過 $1\ 000\ 000$。

注意：**由於 `hack()` 函式會被多次呼叫，參賽者需注意前次呼叫的殘留數據對當前呼叫的影響，尤其是全域變數中儲存的狀態。**

成本限制 $1\ 000\ 000$ 適用於每個測試案例。一般情況下，若有 t 次 `hack()` 呼叫，你可使用總成本不超過 $t \times 1\ 000\ 000$，且每次 `hack()` 呼叫的成本不超過 $1\ 000\ 000$。

交互器不具有適應性，即 n 的值在互動開始前已固定。

範例

假設有 2 個多重測試。評分程式將進行以下呼叫：

```
hack()
```

假設在函式中，你進行了以下呼叫：

呼叫	回傳值
`collisions([2, 15, 7, 27, 8, 30])`	4
`collisions([1, 2, 3])`	0
`collisions([10, 20, 30, 40, 50])`	10

之後，若你確定 n 的值為 5，則 `hack()` 函式應回傳 5。

接著評分程式將進行另一次呼叫：

```
hack()
```

假設在函式中，你進行了以下呼叫：

呼叫	回傳值
`collisions([1, 3])`	1
`collisions([2, 4])`	1

唯一滿足這些查詢的 n 是 2。因此，`hack()` 函式應回傳 2。

限制條件

- $1 \le t \le 10$，其中 t 為多重測試的數量。
- $2 \le n \le 10^9$
- 每次 `collisions()` 呼叫中，$1 \le x[i] \le 10^{18}$。

子任務

1. (8 分) $n \le 500\ 000$
2. (17 分) $n \le 1\ 000\ 000$
3. (75 分) 無額外限制。

在最後一個子任務中，你可以獲得部分分數。設 q 為該子任務所有測試案例中每次 `hack()` 呼叫的最大總成本。你的分數將根據以下表格計算：

條件	分數
$1\ 000\ 000 < q$	0
$110\ 000 < q \le 1\ 000\ 000$	$75 \cdot \log_{50}\left(\frac{10^6}{x-90000}\right)$
$q \le 110\ 000$	75

若在任何測試案例中，`collisions()` 的呼叫不符合實作細節中的限制條件，或 `hack()` 回傳的數字不正確，該子任務的分數將為 0。

範例評分程式

範例評分程式的輸入格式如下：

- 第 1 行：t

接著是 t 行，每行包含一個 n 的值：

- 第 1 行：n

對於每個測試案例，設 m 為 `hack()` 的回傳值，c 為所有查詢的總成本。範例評分程式將以以下格式輸出你的答案：

- 第１行：$m\ c$