



### 第三題：細菌 (Bacteria)

**本題為互動題。**

#### 問題敘述

布朗博士研究水缸細菌，在一個大水缸中，這種細菌打從出生的第二秒開始，每一秒就會分裂產生一隻新一代細菌。在時間  $T = 0$  秒時，布朗博士放進  $A$  個新生細菌，在時間  $T = 1$  秒時，放進  $B$  個新生細菌，這些細菌在整秒時才會進行分裂。有趣的是，這種細菌除非人為將水缸環境徹底消毒，否則永遠不會死亡。你也可以假設布朗博士準備的水缸足夠大，可以容納得下任意正整數隻細菌。布朗博士架設了一個能自動觀察細菌數量的設備，記錄下每過一秒水缸中的細菌總數。

為了更了解這群細菌，布朗博士做了一些實驗，除了在  $T = 0$  與  $T = 1$  時，放入  $A$  與  $B$  個細菌之外，布朗博士設定了放置細菌的機器，能夠額外在每個時間  $T = k$  整秒 ( $k \geq 2$ ) 時，另外放入若干個新生細菌。布朗博士決定進行四場不同的實驗：

- 實驗 1：在每個時間  $T = k$  秒時，額外放入 1 隻新生細菌。
- 實驗 2：在每個時間  $T = k$  秒時，額外放入  $k^2$  隻新生細菌。
- 實驗 3：在每個時間  $T = k$  秒時，額外放入  $k^2 \times 2^k$  隻新生細菌。
- 實驗 4：在每個時間  $T = k$  秒時，如果  $k$  是奇數則額外放入  $2^k$  隻新生細菌，否則不放入細菌。

某天，布朗博士在啟動架設好的實驗設備後，就出門運動去了。就在布朗運動回來以後，他卻意外地發現，自己寫的細菌紀錄程式有 bug：

- 不記得現在是第幾秒鐘了。
- 不記得現在進行的是哪一個實驗。
- 由於溢位的問題，自動觀察細菌數量的設備只記錄下目前看到的細菌數除以  $10^9 + 7$  的餘數。
- 由於記憶體限制的問題，自動觀察細菌數量的設備只記錄最新  $10^6$  秒內的資料，更久以前的資料會因為被覆蓋而消失。

為了準確產出實驗成果，並順利在截稿前將其撰寫成論文投稿至期刊，你能夠幫助焦頭爛額的布朗博士，預測出將來的  $t$  秒以後，水缸中的細菌數量嗎？從自動觀察細菌數量的設備中，查閱歷史紀錄可能會導致設備延遲，進而造成實驗結果的不準確性。因此焦急的布朗博士希望執行你的程式時，查閱歷史紀錄的次數越少越好。



## 實作細節

你需要完成以下函式：

```
int predict(long long t);
```

- `predict( $t$ )` 這個函式必須要能正確回答距離現在  $t$  秒以後水缸中的細菌數量，除以  $10^9 + 7$  的餘數。
- 每一筆測試資料，其評分程式只會呼叫你實作的 `predict( $t$ )` 函式至多 5 次。

你的程式可以呼叫以下函式：

```
int history(int k);  
int get_experiment_type();  
int get_a();  
int get_b();
```

- `history( $k$ )` 函式會回傳距離現在  $k$  秒以前水缸中的細菌數量，除以  $10^9 + 7$  的餘數。輸入的  $k$  必須滿足  $0 \leq k \leq 10^6$ 。若詢問到了實驗開始之前的時間，該函式會回傳 0。
- `get_experiment_type()` 會回傳一個整數  $M \in \{1, 2, 3, 4\}$ ，代表實驗的編號。
- `get_a()` 會回傳實驗設定之  $A$  數值。
- `get_b()` 會回傳實驗設定之  $B$  數值。
- 請注意，`get_experiment_type`、`get_a`、`get_b` 為輔助函式，能夠獲得滿分的程式不需要使用這些函式。
- `get_experiment_type`、`get_a`、`get_b` 在每次的 `predict()` 執行中只能被呼叫至多 1 次。

如果不符合上述條件限制，你的程式會被判為 **Wrong Answer**；否則你的程式會被判斷為 **Accepted**。

## 測資限制

- $M \in \{1, 2, 3, 4\}$ 。
- $0 \leq A \leq 10^9$ 。
- $0 \leq B \leq 10^9$ 。
- 所有呼叫 `predict` 函式的實驗中，保證該次實驗已進行了至少  $T = 5$  秒。



## 互動範例

考慮以下的測試資料：實驗編號  $M = 1$ ,  $A = 1$ ,  $B = 1$ 。

一個被評分程式判斷為 **Accepted** 的互動例子顯示如下：

評分程式端	參賽者端
呼叫 <code>predict(1)</code> 。	呼叫 <code>get_experiment_type()</code> 。
回傳 1。	呼叫 <code>history(5)</code> 。
回傳 1。	呼叫 <code>history(4)</code> 。
回傳 2。	回傳 33。

## 評分說明

對於每一次的 `predict` 函式執行中，假設你呼叫 `history` 的次數為  $C_h$ 。此外，我們同時定義  $C_M, C_A, C_B$  這三個數值，其中  $C_M$  的值為 1 或 0，表示有無呼叫 `get_experiment_type`。 $C_A$  與  $C_B$  也相似，分別表示有無呼叫 `get_a` 與 `get_b`。得到的**分數比重**  $S$  值定義如下：

$$S = \min \left( 1.0, \sqrt{\frac{\log 6}{\log C_h}} \right) \times (1.0 - 0.5C_M - 0.1C_A - 0.1C_B)$$

本題共有 6 組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，你的得分是該組所有測試資料之得分比重  $S$  中最低者，乘以該子任務的分數。

子任務	分數	額外輸入限制
1	4	總是進行實驗 1，並且傳入之參數 $t$ 滿足 $1 \leq t \leq 10^6$ 。
2	5	總是進行實驗 2，並且傳入之參數 $t$ 滿足 $1 \leq t \leq 10^6$ 。
3	6	總是進行實驗 3，並且傳入之參數 $t$ 滿足 $1 \leq t \leq 10^6$ 。
4	7	總是進行實驗 4，並且傳入之參數 $t$ 滿足 $1 \leq t \leq 10^6$ 。
5	23	保證實驗開始迄今不超過 1000 秒，參數 $t$ 滿足 $1 \leq t \leq 10^{18}$ 。
6	55	實驗開始迄今秒數無上限，參數 $t$ 滿足 $1 \leq t \leq 10^{18}$ 。



## 範例評分程式

範例評分程式以下列格式讀取輸入：

- 第 1 列： $K$  ( $K \leq 5$ )，代表有幾場實驗。
- 第 2 至  $K + 1$  列：每一列有五個以空白隔開的整數： $M, A, B, now, t$ 。其中  $now$  代表了詢問當下實驗已進行的秒數、而  $t$  值則是用來傳入 `predict` 函式的引數。

**請注意：使用自己上傳的測試資料進行測試時，沒有下列 MSG 描述的情形時你總會得到 Accepted。** 如果你的程式被評為 **Accepted**，範例評分程式輸出 `Accepted: <output1> <output2> ...`，其中  $output_i$  對應到各場實驗的 `predict()` 回傳值。如果你的程式被評為 **Wrong Answer**，範例評分程式輸出 `Wrong Answer: MSG`，其中 MSG 格式與意義如下：

- `invalid K`: 實驗場數不合法。
- `invalid experiment type`: 實驗編號不合法。
- `invalid A`: 輸入之  $A$  值不合法。
- `invalid B`: 輸入之  $B$  值不合法。
- `invalid now`: 輸入之當前秒數不合法。
- `invalid t`: 呼叫 `predict` 傳入之引數不合法。
- `too many history calls`: 呼叫 `history` 超過  $10^6$  次。
- `invalid k for history`: 呼叫 `history` 傳入之引數不合法。
- `too many get_experiment_type calls`: 呼叫 `get_experiment_type` 超過 1 次。
- `too many get_a calls`: 呼叫 `get_a` 超過 1 次。
- `too many get_b calls`: 呼叫 `get_b` 超過 1 次。
- `corrupted stub`: 可能改到 `stub` 了，請重新下載 `stub.cpp`。