

D. 等比數列和

Description

廢話不多說，本題希望你求出所謂的「等比數列和」，當然，大家都知道等比數列和有公式解，但這個「公式解」可是有使用條件的，而在本題，我們把使用公式解的條件去除了，請問你是否依然有方法快速的求出等比數列和呢？(本題定義 $0^0 = 1$)

也就是說，給定公比 r 、項數 n 和首項 1，請你求出：

$$\sum_{i=0}^{n-1} r^i = 1 + r + r^2 + \dots + r^{n-1}$$

Instructions

為了實現「去除條件」，**本題為互動題**，你要透過評分系統提供的函數和物件，來模擬運算的過程。請引用標頭檔 `series.h`，並且完成 `Element get_sum(Element, long long)` 這個函數。請使用以下的函數們與評分系統溝通，並且 **請勿進行任何輸入 (stdin)、輸出 (stdout)** (輸出到 `stderr` 將被忽略)。

1. `class Element`：這是經過特殊處理的整數，你可以將其當成一個特殊的整數型態使用（意即，宣告 `Element e` 可以獲得一個特殊整數 `e`）。在附件當中提供了簡單的實作版本，但與 **Judge 端上的版本有非常大的不同**，請勿利用接下來提到的函式以外的方法來接觸這類型的整數。
2. `Element get_sum(Element r, long long n)`：這是你要寫出的函數。評分系統會進行 T 次等比數列求和，每次會呼叫 `get_sum(r, n)` 來嘗試獲得答案。你必須在這個函數裡面與評分系統互動，來完成等比數列的求和，並將結果回傳。注意到你回傳的物件也必須是一個 `Element` 型態的整數。
3. `Element Element::operator+(const Element &e)`：我們提供了 `Element` 整數的**加法**運算子做為運算使用。你可以很自由的把他當成一般整數加法一樣對 `Element` 型態的整數們使用。
4. `Element Element::operator*(const Element &e)`：我們提供了 `Element` 整數的**乘法**運算子做為運算使用。你可以很自由的把他當成一般整數乘一樣對 `Element` 型態的整數們使用。注意到**先乘後加的順序並沒有改變**。
5. `Element& Element::operator=(const Element &e)`：你可以很自由的使用 `=` 運算子來複製得到一個已知的 `Element` 整數。當然，`=` 運算子實際上是預設就有的，我們並沒有真的將其實作出來，在這邊只是做個提醒你可以使用他。
6. `Element get_one()`：你可以呼叫此函數來獲得 `Element` 整數型態包裝過後的 1。用來提供你一個運算的基底。

對於特殊整數 `Element`，請勿試圖呼叫或幫忙實作上述函數以外的其他功能，因為你不可能也無法知道 Judge 端上的實作內容，包含 CMS 附件中的所有實作長相**都是假的**，只是為了讓你本地測試用。

當然，我們保證上述函數絕對可以正常呼叫，你也同樣不需要擔心任何與溢位相關的問題，你可以想像我們已經在加法和乘法的運算過程中幫你取模了。基本上你完全可以把 `Element` 當成正常的整數（例如 `int`）使用，只要注意不能使用減法、除法、取模等等操作就行。

不過因為技術因素，`Element` 的運算非常緩慢，你可能要做點速度測試來確認時限內可以進行的運算次數，也可以等 TLE 之後再來思考是否是演算法過於緩慢。

Sample Code

以下程式碼是一個簡單的範例，可以在呼叫 `get_sum(r, n)` 後回傳 $(r + 1) * r$ ，在 CMS 的附件中的 `series.cpp` 內同樣可以看到類似的實作。

```
1 Element get_sum(Element r, long long n) {  
2     return (r + get_one()) * r;  
3 }
```

Local testing tool

範例評分程式會按照以下的格式讀取輸入：

- 第一列：一個正整數 T ，代表該筆測試資料的測試數量。
- 第 $2 \sim T + 1$ 列：第 i 列兩個非負整數 r_i, n_i 。代表第 i 次測試要求出公比為 r_i ，項數為 n_i ，首項為 1 的等比數列和。

輸入完後，若你的 `get_sum` 函數實作正確，範例評分程式會輸出 T 行，第 i 行代表第 i 次測試的等比數列和除以 998244353 後的餘數。

在 CMS 的附件中，我們有附上 `series.cpp` 這個檔案，內部已經有寫好一些東西的檔案。直接上傳該檔案並不會導致 Compile Error，並且往後你要寫的程式都可以參考這份檔案，並在這份檔案內進行修改、編譯以及執行。

請注意，檔案內有兩行註解分別為「do not modify above」和「do not modify below」，這兩行意味著希望你盡量只更動被這兩行夾住的區域，若你對互動題並沒有很熟悉，請不要更動外面的區域！

Constraint

以下是一些輸入限制：

- $1 \leq T \leq 2 \times 10^4$
- $1 \leq n \leq 10^{18}$
- r 可以正常的被 Element 存取

Sample 1

Input	Output
5	1
100 1	13
3 3	63
2 6	10
1 10	917335668
10 4284285	

配分

在一個子任務的「測試資料範圍」的敘述中，如果存在沒有提到範圍的變數，則此變數的範圍為 Constraint 所描述的範圍。

子任務編號	子任務配分	測試資料範圍
1	20%	$T, n \leq 100$
2	30%	$T \leq 500, n = 2^k$ 對於某個非負整數 k
3	30%	$T \leq 500$
4	20%	無特殊限制

Hint

請真的不要嘗試去偷資料什麼的，我們會保留人工檢查的權利，如果發現你有一些不太正規的操作可能會導致你該題零分。