



第三題：古文字 (Hieroglyphs)

本題為互動題。

問題敘述

Alice 與 Bob 是兩個熱愛考古的考古迷，現在兩人正在埃及研究金字塔裡的古文字以及歷史。在研究古文字的過程中他們發現封閉的金字塔環境裡時常充滿著陳年的水氣，因為濕氣造成牆壁上文字的劣化變成了解讀古文字的一個極大的難題。為了更加了解濕氣對文字的影響，在經過一番討論後，兩人決定用兩種字元白色圓圈「○」以及黑色圓圈「●」來組成一個簡易書寫系統進行實驗。

經過一些簡單的實驗後他們發現黑色圓圈「●」常常會因為濕氣而造成下一個字元劣化；劣化的字元會從白色圓圈「○」變成黑色圓圈「●」，或者從黑色圓圈「●」白色圓圈「○」。兩人歸納出了一些重要的結論：

- 一個劣化的字元前面恰好都是未劣化的黑色圓圈「●」
- 兩劣化字元之間至少距離 d
- 一個字串中最多只有 r 個劣化字元

也就是說若一串文字裡有子字串「●○」的話可能因為劣化而變成「●●」；並且子字串「●●」也有可能因為劣化變成「●○」。舉例來說若 $r = d = 2$ ，字串「●●●●○」可能會因為第 2, 4 個位置受前面黑色圓圈影響劣化變成「●○●○○」，也有可能只有位置 3 劣化變成「●●○●○」，但位置 2, 3 並不會同時劣化變成「●○○●○」。

為了能更有效率地模擬文字劣化的狀況，兩人決定創造一些金字塔模型並用圓圈表示將數字寫在牆上觀察劣化狀況。預計會進行 Q 場實驗，在實驗過程裡兩人將獨立進行以下的實驗事項：

- Alice 想一個數字，並在某個金字塔模型牆上寫上兩種圓圈字元組成的字串表示此數字
- 過一段時間後 Bob 進到金字塔模型裡將目前牆上看到的字串記錄下來並註記原本 Alice 想的數字值以便分析

由於每個金字塔模型實驗時間不盡相同，Bob 進金字塔的順序可能與 Alice 相異。為了能在文字劣化狀況下正確分別 Alice 寫下的數字，他們決定請參加資訊奧林匹亞的你來設計一個抗劣化的數字書寫系統來協助他們的實驗。並且為了增進實驗的效率，他們希望設計出來的書寫系統越短越好。



實作細節

你需要在同一個檔案裡實作以下兩個函式：

```
std::string Alice(long long x);  
long long Bob(std::string s);
```

- $Alice(x)$ 函式的參數 x 為一個 0 到 n 之間的整數，代表 Alice 將在金字塔模型裡寫下的數字。
- $Alice(x)$ 的回傳值代表 Alice 將數字 x 寫在金字塔裡表示法。這個字串必須由字元 W 與字元 B 組成，W 代表白圓圈「○」，B 代表黑圓圈「●」。且字串長度至少須為 1。
- $Bob(s)$ 函式的參數 s 代表 Bob 進金字塔後在牆上所看到可能劣化的 Alice 寫的字串。
- $Bob(s)$ 的回傳值代表 Bob 看到字串 s 後推測 Alice 原本寫的數字值。這個數字必須要和 Alice 當初想寫在牆上的數字相等。

你的程式可以引入標頭檔 `#include "hieroglyphs.h"` 並在任意時間點呼叫以下函式：

- `int getQ()`: 回傳 Q 值，也就是該筆測試資料 Alice 與 Bob 函式各自將被執行的次數。
- `long long getN()`: 回傳 n 值，也就是該筆測試資料 $Alice(x)$ 最大有可能會收到的參數 x 。
- `int getD()`: 回傳 d 值，也就是該筆測試資料兩劣化字元至少需相隔的距離。
- `int getR()`: 回傳 r 值，也就是該筆測試資料一字串劣化字元最多的數量。
- `int getK()`: 回傳該筆測資的評分參數 k ，請見評分說明一節。

執行細節

你的程式將與 `stub.cpp` 一起編譯為兩個執行檔 A 與 B 並同時執行。程式將會用以下方法測試：

1. 呼叫執行檔 A 的 $Alice(x)$ 函數 Q 次，並取得回傳值。
2. 呼叫執行檔 B 的 $Bob(s)$ 函數 Q 次，並取得回傳值。 $Bob(s)$ 函式的參數 s 是由 $Alice(x)$ 參數回傳值經題敘的劣化過程形成的。
3. 進行評分。



互動範例

一個可以得到分數比重 $S = 1.0$ 的互動例子顯示如下：(輸入檔格式請參考末頁「本地測試輸入格式」)

[執行檔 A]:

輸入檔	呼叫	回傳	呼叫	回傳
Alice 2 5 10 5 2 3 5	Alice(3)			
			getN()	
				5
			getQ()	
				2
			getK()	
				10
			getR()	
				5
			getD()	
				2
		"BBB"		
	Alice(5)			
		"BB"		

[執行檔 B]:

輸入檔	呼叫	回傳	呼叫	回傳
Bob 2 5 10 5 2 BB BWB	Bob("BB")			
		5		
	Bob("BWB")			
		3		

註：BWB 為 BBB 經劣化變成的字串，請注意 Bob 的呼叫順序不一定和 Alice 一樣。



測資限制

- $1 \leq Q \leq 100$ 。
- $2 \leq n < 2^{63}$ 。
- $0 \leq x \leq n$ 。
- $2 \leq d \leq 10$ 。
- $1 \leq r \leq k + 20$ 。
- s 為一個僅由 W 與 B 字元組成的字串，並且 s 必定是由某 $\text{Alice}(x)$ 回傳值經劣化形成的。 s 中至多有 r 個字元劣化，並且任意兩劣化字元都至少距離 d 。
- k 的範圍請見各子題設定。

評分說明

每筆測試資料皆有一個評分參數 k ，程式將計算以下的值：

- L = 該筆測資中 Alice 回傳最長的字串長度
- $l_1 = \min\{5, \max\{0, L - k\}\}$
- $l_2 = \max\{0, L - k - 5\}$

並用以下方式計算每筆輸入資料**分數比重** S ：

- 若 Alice 所回傳的字串不符合規定，程式將回傳 `Alice(): Invalid Return`，該筆輸入 $S = 0$ 。
- 若 Bob 解讀出來的數字與當初 Alice 想的數字不符，程式將回傳 `Bob(): Incorrect Number`，該筆輸入 $S = 0$ 。
- 若 $L > k + 20$ ，程式將回傳 `Alice(): Too Long`，該筆輸入 $S = 0$ 。
- 若 $k < L \leq k + 20$ 且 Bob 解讀的數字正確。程式將回傳 `Score: {S}`，該筆輸入 $S = 0.75^{l_1} \times 0.9^{l_2}$ 。
- 若 $L \leq k$ 且 Bob 解讀的數字正確，程式將回傳 `Score: 1.0`，該輸入 $S = 1.0$ 。



本題共有 7 組子任務，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	3	$n = 99, k = 100$ 。
2	13	$k = 89, r = 1$ 。
3	9	$k = 89, r = 2$ 。
4	15	$k = 89, d = 10$ 。
5	10	$k = 189$ 。
6	9	$k = 126$ 。
7	41	$k = 89$ 。

評分訊息

請注意: 只有正式評分才会有以下訊息，並且本題請不要線上測試

- `Alice(): Invalid Return`: 代表 Alice 回傳的字串不符合格式規定。
- `Bob(): Incorrect Number`: 代表 Bob 解讀的數字與當初 Alice 想寫的数字不符。
- `Alice(): Too Long`: Alice 所寫下的字串長度大於 $k + 20$ 。
- `Score {S}`: 代表 Bob 成功解讀數字， S 為該輸入檔所得分數比重。



本地測試輸入格式

你可以在本地執行程式測試，附件將提供 `stub.cpp` 作為與 judge 溝通的手段參考。你可以用以下的方法將自己的程式與 `stub.cpp` 一起編譯：

```
g++ stub.cpp my_program.cpp -o execute # 編譯成 execute 執行檔
./execute # 執行 execute 執行檔
```

與 `stub.cpp` 編譯的程式將依照輸入格式決定該程式的角色為 A 程式或 B 程式。以下為與 `stub.cpp` 一起編譯的 A 程式接受的輸入格式：

```
Alice
Q n k r d
x1
x2
⋮
xQ
```

其中 x_1 到 x_Q 分別為 Alice 將依序收到的函式參數 x 值。以下為與 `stub.cpp` 一起編譯的 B 程式接受的輸入格式：

```
Bob
Q n k r d
s1
s2
⋮
sQ
```

其中 s_1 到 s_Q 分別為 Bob 將依序收到的函式參數 s 值。