



第三題：密碼提示系統 (guesspass)

本題為互動題。

問題描述

有一個未知的密碼字串，其長度為 n 。已知一個正整數 m 而且該字串僅包含前 m 個大寫英文字母；而且每一種大寫英文字母皆至少出現一次。

由於當初設定密碼的人擔心自己忘記密碼，他建置了一套密碼提示系統，你可以詢問關於整串密碼子序列的一些問題。這個系統為了防止健忘，它僅支援子序列輸入與兩種功能： $\text{GetCount}(S)$ 與 $\text{GetRank}(S)$ 。

用法說明如下： S 為一數值嚴格遞增的陣列，其中每個元素表示一個註標。密碼字串中，位於註標 $S[0] < S[1] < \dots < S[k-1]$ 的字元依序取出所形成的子序列稱為 X 。

- $\text{GetCount}(S)$: 若對系統下達此詢問，則該系統會告訴你子序列 X 裡面有多少種不同的字元。
- $\text{GetRank}(S)$: 若對系統下達此詢問，則可以得知 X 的順位為何。此處的順位指的是在 X 中所有出現的字元所能形成的相異排列中，依照字典順序由小到大排序後， X 為當中的第幾個。比方說，當子字串 $X = \text{"TOI"}$ 時， GetRank 功能會回傳 6。不過由於建置密碼提示系統的人過於健忘，他完全忘記實作大數運算了。一旦順位超過 10^{15} ，此功能就會完全失效，會回傳 -1 。

你的目標是要透過一系列上述的子序列詢問，精準還原出未知的密碼字串。



實作細節

你需要完成以下函式：

```
std::string Recover(int n, int m);
```

- `Recover(n, m)` 函式的參數 n 代表未知字串的長度、參數 m 代表未知字串使用的不同字母數量。
- `Recover(n, m)` 的回傳值為還原後的密碼字串。
- 對於每一筆測試資料，評分程式會呼叫你實作的 `Recover()` 函式至多 5 次。

你的程式可以呼叫以下輔助函式：

```
int GetCount(const std::vector<int> &S);  
long long GetRank(const std::vector<int> &S);
```

- `GetCount(S)` 函式的參數 S 描述密碼裡任何一個子序列的註標編號。傳入之註標編號必須遞增，且每一個數字均介於 0 與 $n - 1$ 之間。
- `GetCount(S)` 回傳值代表了密碼裡該子序列包含了多少種不同的字元。
- `GetRank(S)` 函數的參數 S 描述密碼裡任何一個子序列的註標編號。傳入之註標編號必須遞增，且每一個數字均介於 0 與 $n - 1$ 之間。
- `GetRank(S)` 回傳值代表了該子序列在所有可能排列中的順位，若該順位超過 10^{15} 會回傳 -1。
- 每一次呼叫 `GetCount(S)` 或 `GetRank(S)` 都會伴隨著代價，這個代價被定義為 $5 + |S|$ 。
- 在一次 `Recover()` 的呼叫中，你的程式呼叫 `GetCount()` 與 `GetRank()` 函式的代價總和不能超過 200000。

如果你的程式回傳了錯誤的密碼字串、或者傳入的註標陣列 S 不符合上述條件限制、或者呼叫的代價總和超過 200000，那麼你的程式會被判為 **Wrong Answer**；否則你的程式會被判為 **Accepted**，並以評分說明中的方式來決定分數。



測資限制

- $2 \leq n \leq 1000$ 。
- $2 \leq m \leq 26$ 。
- $m \leq n$ 。
- 未知的密碼字串在評分程式呼叫 `Recover()` 之前就已經確定了，不會因為你的詢問而改變。

互動範例

一個可能被評分程式判斷為 **Accepted** 的互動例子顯示如下：

評分程式端	參賽者端
呼叫 <code>Recover(6, 2)</code> 。	呼叫 <code>GetCount([0, 1, 2])</code> 。
回傳 2。	呼叫 <code>GetRank([0, 1, 2])</code> 。
回傳 3。	呼叫 <code>GetCount([1, 2, 3, 4, 5])</code> 。
回傳 1。	回傳 "BAAAAA"。

評分說明

對於每一筆測試資料，你將會得到的分數比重 S 值如下：

- 如果任何一次 `Recover()` 回傳的密碼不正確，則 $S = 0$ 。
- 現在考慮回傳的字串皆為正確密碼字串的情形。假設你的程式執行時，在所有 `Recover()` 的呼叫中，你的程式呼叫 `GetCount()` 與 `GetRank()` 函式的代價總和的最大值為 Q ，那麼

$$S = \begin{cases} 0 & \text{if } Q > 200000, \\ \left\lfloor \frac{200000 - Q}{3200} \right\rfloor \times 0.01 & \text{if } 40000 < Q \leq 200000, \\ 0.5 + \left\lfloor \frac{40000 - Q}{200} \right\rfloor \times 0.01 & \text{if } 30000 < Q \leq 40000, \\ 1 & \text{if } Q \leq 30000. \end{cases}$$



本題共有 4 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	密碼字串中的每一個字元皆相異。
2	10	$m = 2$ 。
3	20	$m \leq 10$ 。
4	67	無額外限制。

範例評分格式

範例評分程式以下列格式讀取輸入：

```
T
password1
password2
⋮
passwordT
```

- T 代表評分程式呼叫 `Recover()` 函式的次數。
- $password_i$ 是一個由大寫英文字母組成的字串。

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**，範例評分程式會輸出你的程式在每一次 `Recover()` 執行時的代價總和 Q_1, Q_2, \dots, Q_T 。

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：

- `incorrect result`：函式執行完畢以後，回傳的字串與密碼字串不同。
- `Q limit exceeded`： $Q > 200000$ 。
- `invalid parameters`：傳給 `GetCount` 或 `GetRank` 的參數不符合規定。
- `corrupted stub`：可能改到 `stub` 了，請重新下載 `stub.cpp`。