

## 龍蝦抄寫器

Johannes Gutenberg是一個德國的鐵匠，他發明了移動式印刷術。人們都說Leonardo是他最偉大的追隨者，因為Leonardo為了紀念Johannes而設計出一部非常精巧的打字機器，名叫龍蝦抄寫器(il gambero scrivano)。它有點像現今一部有簡單功能的打字機，它只接受二種指令：一個指令是打出下一個字元，而另一個指令則是復原上一個最近的指令。這部龍蝦抄寫器最大特點就是有這個功能強大的復原指令(undo command)：因為一個復原的指令本身也是一個命令，因此它也可以重做(undone)。

## 說明

你的任務就是制作這個龍蝦抄寫器的軟體，它開始是一個空白的字符串，然後它能接受由用戶輸入的一連串指令，並可查詢當前字符串中的指定位置。詳細說明如下：

- `Init()` — 在程序開始運行時被呼叫一次，而且不需要參數。它用來初始化資料結構，而且也絕不會被復原。
- `TypeLetter(L)` — 附加一個小寫字母L在字符串的後面，其中參數L是a, ..., z中的其中一個。
- `UndoCommands(U)` — 復原(undo)最後的U個指令，其中參數U是一個正整數。
- `GetLetter(P)` — 返回在當前字符串中位置P的字母，其中參數P是一個非負整數序號(index)。文字中的第一個字母的序號(index)為0。（由於這個查詢並不是一個指令，因此它會被復原指令所忽略。）

在初始化函數`Init()`被呼叫之後，其他的函數都可以被呼叫0次或多次，並且不限順序。我們保證U一定不會超過前面已經收到的指令數目，而且P一定小於目前文字的長度（當前字符串中的字母數）。

對於指令`UndoCommands(U)`，它會以相反的順序復原(undo)前面的U個指令：例如，如果要復原的指令是`TypeLetter(L)`，那麼它就會從文字最後面移除字母L。如果要復原的指令是`UndoCommands(X)`，那麼它將會重做(undone)這些原本被復原的X個指令，並且要依照原本執行的順序執行（it re-does the previous X commands in their *original* order）。

## 範例

我們列出一連串可能的呼叫，以及每次呼叫後字符串文字的狀態。

呼叫(call)	返回>Returns)	當前字符串(Current text)
Init()		
TypeLetter(a)		a
TypeLetter(b)		ab
GetLetter(1)	b	ab
TypeLetter(d)		abd
UndoCommands(2)		a
UndoCommands(1)		abd
GetLetter(2)	d	abd
TypeLetter(e)		abde
UndoCommands(1)		abd
UndoCommands(5)		ab
TypeLetter(c)		abc
GetLetter(2)	c	abc
UndoCommands(2)		abd
GetLetter(2)	d	abd

## 子任務 1 [5 分]

指令和查詢的總數在1到100之間(包括100)，而且不會呼叫到UndoCommands指令。

## 子任務 2 [7 分]

- 指令和查詢的總數在1到100之間(包括100)，而且UndoCommands指令不會被重做。

## 子任務 3 [22 分]

- 指令和查詢的總數在1到5 000之間(包括5 000)

## 子任務 4 [26 分]

指令和查詢的數目在1到1 000 000之間(包括1 000 000)。所有GetLetter呼叫都一定會在所有的TypeLetter 和 UndoCommands指令後出現。

## 子任務 5 [40 分]

- 指令和查詢的數目在 1 到 1 000 000 之間(包括1 000 000)

## 編程細節

你必須上傳一程式檔案scrivener.c、scrivener.cpp 或scrivener.pas。這檔案必須實現上述的描述並符合下面的說明。

## C/C++ 程式

```
void Init();  
void TypeLetter(char L);  
void UndoCommands(int U);  
char GetLetter(int P);
```

## Pascal 程式

```
procedure Init;  
procedure TypeLetter(L : Char);  
procedure UndoCommands(U : LongInt);  
function GetLetter(P : LongInt) : Char;
```

這些子程式一定要根據上述的特點來編寫。當然你亦可自由地編寫其他在計算過程中需要的子程式。你所編寫的程式不要與標準輸入輸出有任何的直接互動。亦不能使用任何其他的檔案。

### 評分系統範例 (Sample grader)

範例評分系統從符合下面的格式的資料進行輸入：

- 第一行: 在輸入中所有指令和查詢的總數;
- 接著下面的每一行:
  - T 後面接著一個空白和一個小寫字母，表示一個 `TypeLetter` 指令;
  - U 後面接著一個空白和一個整數，表示一個 `UndoCommands` 指令;
  - P 後面接著一個空白和一個整數，表示一個 `GetLetter` 指令。

範例評分系統會打印出呼叫 `GetLetter` 後返回的字符，每一行印一個字符。