Escribano Cangrejo

Alguna gente piensa que Leonardo era un gran admirador de Johannes Gutenberg, el herrero alemán que inventó la imprenta de tipos movibles, y al cual le dio un homenaje diseñando una máquina llamada el escribano cangrejo – *il gambero scrivano* – un dispositivo muy simple de digitación. De alguna manera es semejante a una máquina de escribir moderna que acepta solamente dos comandos: uno para digitar el siguiente carácter y uno para deshacer el comando más reciente. Una característica notable del escribano cangrejo es que el comando deshacer es tremendamente poderoso: un deshacer se considera también un comando, y puede ser desecho.

Enunciado

Su tarea es implementar un programa que simule el escribano cangrejo: comienza con un texto vacío y acepta una secuencia de comandos introducidos por el usuario, y pregunta por posiciones especificas de la versión actual del texto, como sigue.

- `Init`() — Se llama una vez al comienzo de la ejecución, sin argumentos. Puede ser usado para inicializar estructuras de datos. Nunca necesitará ser desecho.
- `TypeLetter`(L) — Añadir al final del texto una sola letra minúscula L elegida `a`, …, `z`.
- `UndoCommands`(U) — Deshacer los últimos U comandos para un entero positivo U.
- `GetLetter`(P) — Devolver la letra en la posición P en el texto actual, para un indice no negativo P. La primera letra en el texto tiene índice 0. (Esta pregunta no es un comando y por lo tanto es ignorada por el comando deshacer).

Después de la primera llamada a `Init`(), las otras rutinas pueden ser llamadas cero o más veces en cualquier orden. Se garantiza que U no excederá el número de comandos recibidos previamente, y que P nunca será menor que la longitud del texto actual (el número de letras en el texto actual).

Con respecto al comando `UndoCommands`(U), deshace los U comandos previos en orden inverso: si el comando a ser desecho es `TypeLetter`(L), borra L del final del texto actual; si el comando antes de undone es `UndoCommands`(X) para algún valor X, vuelve a ejecutar los X comandos previos en su orden original.

Ejemplo

Mostramos una secuencia posible de llamadas, junto con el estado del texto después de cada llamada.

Llamado	Devuelve	Texto Actual
`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`

Subtarea 1 [5 puntos]

- El número de comandos y de preguntas está entre 1 y 100 (inclusive) y no habrán llamadas a `UndoCommands`.

Subtarea 2 [7 puntos]

- El número de comandos y preguntas está entre 1 y 100 (inclusive) y ningún comando `UndoCommands` será desecho.

Subtarea 3 [22 puntos]

El número de comandos y preguntas está entre 1 y 5 000 (inclusive).

Subtarea 4 [26 puntos]

- El número de comandos y preguntas está entre 1 y 1 000 000 (inclusive). Todas las llamadas a `GetLetter` ocurren después de todos los comandos `TypeLetter` y `UndoCommands`.

Subtarea 5 [40 puntos]

- El número de comandos y de preguntas está entre 1 y 1 000 000 (inclusive).

Detalles de implementación

Usted debe enviar exactamente un archivo, llamado `scrivener.c`, `scrivener.cpp` o `scrivener.pas`. Este archivo debe implementar los subrutinas descritas anteriormente usando los siguientes encabezados.

Programas en C/C++

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

Programas en Pascal

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

Estas subrutinas se deben comportar como describe el enunciado. Por supuesto, usted es libre de implementar cualquier otra subrutina para uso interno. Sus envíos no deben interactuar de ninguna forma con la entrada o salida estándar, o con algún otro archivo.

Calificador ejemplo

El calificador ejemplo lee la entrada y la salida con el siguiente formato:

- Línea 1: El número total de consultas y comandos en la entrada;
- En cada una de las siguientes líneas:
  - `T` seguida de un espacio y una letra en minúscula para un comando tipo `TypeLetter`;
  - `U` seguida de un espacio y un entero para un comando tipo `UndoCommands`;
  - `P` seguida de un espacio y un entero para un comando tipo `GetLetter`.

El calificador ejemplo imprimirá los carácteres que retorna `GetLetter`, cada una en una línea separada.