

Crayfish scrivener

Hay gente que dice que Leonardo fue un gran admirador de Johannes Gutenberg, el herrero germano que inventó la impresión de tipos movibles, y que pagó homage by diseñando una máquina llamada el crayfish scrivener — *il gambero scrivano* — un dispositivo muy simple de escritura. Es de algún modo similar a una simple y moderna máquina de escribir y acepta sólo dos comandos: uno para escribir el siguiente carácter y uno para deshacer los comandos más recientes. El notable feature del crayfish scrivener es que el comando deshacer es extremadamente poderoso: un deshacer es también considerado un comando en sí mismo, y puede ser deshecho.

Enunciado

Tu tarea es realizar una versión software del crayfish scrivener: comienza con un texto vacío y acepta una secuencia de comandos ingresados por el usuario, y consultas por posiciones específicas de la versión corriente del texto, tal como sigue.

- `Init()` — llamado una vez al comienzo de la ejecución, sin argumentos. Puede ser usado para inicializar estructuras de datos. Nunca será necesario deshacerlo.
- `TypeLetter(L)` — agregue al final del texto una sola letra `L` en minúscula elegida de entre `a`, ..., `z`.
- `UndoCommands(U)` — deshace los últimos `U` comandos, para un entero positivo `U`.
- `GetLetter(P)` — retorna la letra en la posición `P` en el texto corriente, para un índice no-negativo `P`. La primera letra en el texto tiene índice 0. (Esta consulta no es un comando y por lo tanto es ignorado por el comando deshacer.)

Después de la llamada inicial a `Init()`, las otras rutinas pueden ser llamadas cero o más veces en cualquier orden. Está garantizado que `U` no excederá el número de comandos previamente recibidos, y que `P` será menor que el largo del texto corriente (el número de letras en el texto corriente).

Como para `UndoCommands(U)`, él deshace los `U` comandos previos en orden *reverso*: si el comando a ser deshecho es `TypeLetter(L)`, entonces remueve `L` del fin del texto corriente; si el comando a ser deshecho es `UndoCommands(X)` para algún valor `X`, re-hace los `X` comandos previos en su orden *original*.

Example

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

Llamada	Retorna	Texto corriente
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 total de comandos y consultas estará entre 1 y 100 (inclusive) y no habrá llamadas a `UndoCommands`.

Subtarea 2 [7 puntos]

- El número total de comandos y consultas estará entre 1 y 100 (inclusive) y no `UndoCommands` será deshecho.

Subtarea 3 [22 puntos]

- El número total de comandos y consultas estará entre 1 y 5 000 (inclusive).

Subtarea 4 [26 puntos]

- El número total de comandos y consultas estará entre 1 y 1 000 000 (inclusive). Todos los llamados a `GetLetter` después de todas las llamadas a `TypeLetter` y `UndoCommands`.

Subtarea 5 [40 puntos]

- El número total de comandos y consultas estará entre 1 y 1 000 000 (inclusive).

Detalles de implementación

Debes enviar exactamente un archivo, llamado `scrivener.c`, `scrivener.cpp` or `scrivener.pas`. Este archivo debe implementar los subprogramas descritos arriba usando los siguientes encabezamientos.

programas C/C++

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

programas Pascal

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

Estos subprogramas se deben comportar tal como está descrito arriba. Por supuesto eres libre de implementar otros subprogramas para su uso interno. Sus envíos no deben interactuar de ningún modo con “standard input/output”, ni con ningún otro archivo.

Calificador simple

- El calificador simple lee su entrada en el siguiente formato:
- Línea 1: el número total de comandos y consultas en la entrada;
- En cada línea siguiente:
 - T seguido de un espacio y un letra minúscula para un comando `TypeLetter`;
 - U seguido de un espacio y un entero para `UndoCommands`;
 - P seguido de un espacio y un entero para `GetLetter`.

El calificador simple imprimirá los caracteres retornados por `GetLetter`, cada uno en una línea separada.