

Crayfish scrivener

Algunas personas dicen que Leonardo fue un gran admirador de Johannes Gutenberg, el herrero alemán que invento la impresora movil, y lo homenajeó diseñando una máquina llamada crayfish scrivener — *il gambero scrivano* — un dispositivo para imprimir muy simple. Este es de un modo similar a la simple impresora moderna y acepta solamente dos comandos: uno para el tipo del próximo caracter y uno para deshacer el comando mas reciente. La notable característica de la crayfish scrivener es que el comando deshacer es extremadamente fuerte: un deshacer puede también considerarse un comando mismo y puede deshacerse.

Problema

Su tarea es realizar una versión del software de la crayfish scrivener: este inicia con un texto vacío y acepta una secuencia de comandos entrados por el usuario, y pregunta por posiciones específicas de la versión actual del texto, como sigue.

- `Init()` — llamado una vez al comenzar la ejecución, sin argumentos. Este puede usarse para inicializar las estructuras de datos. Este nunca necesitará un deshacer.
- `TypeLetter(L)` — agrega al final del texto una simple letra minúscula `L` seleccionada de `a, ..., z`.
- `UndoCommands(U)` — deshace los últimos `U` comandos, para un entero positivo `U`.
- `GetLetter(P)` — retorna la letra de la posición `P` en el texto actual, para un índice entero no negativo `P`. La primera letra en el texto tiene índice 0. (Esta pregunta no es un comando y esta es ignorada por el comando deshacer)

Las rutinas de arriba pueden ser llamadas cero o mas veces en cualquier orden. Se garantiza que `U` no excederá el número de comandos previamente recibidos, y que `P` será menor que la longitud del texto actual (el número de letras en el texto actual).

En cuanto a `UndoCommands(U)`, este deshace los `U` comandos previos en orden 'inverso': *si el comando a ser deshecho es `TypeLetter(L)`, entonces este elimina la letra `L` de el final del texto actual; si el comando a ser deshecho es `UndoCommands(X)` para algún valor `X`, este rehace los `X` comandos previos en su orden original*.

Ejemplo

Mostramos una posible secuencia de llamadas, junto con el estado de los textos después de cada llamada.

Llamada	Retorna	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 total de comandos y preguntas está entre 1 y 100 (inclusive) y no habrá llamadas a `UndoCommands`.

Subtarea 2 [7 puntos]

- El número total de comandos y preguntas está entre 1 y 100 (inclusive) y ningún `UndoCommands` hace un deshacer.

Subtarea 3 [22 puntos]

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

Subtarea 4 [26 puntos]

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

Subtarea 5 [40 puntos]

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

Detalles de la implementación

Usted tiene que enviar exactamente un fichero, llamado `scrivener.c`, `scrivener.cpp` o `scrivener.pas`. Este fichero tiene implementado los subprogramas descritos arriba usando las siguientes firmas.

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;
```

Estos subprogramas tienen que comportarse como se describe arriba. Por supuesto que usted es libre para implementar otros subprogramas para su uso interno. Sus envíos no tienen que interactuar de cualquier manera con standard input/output, ni con cualquier otro fichero.

Ejemplo de evaluador (grader)

Este ejemplo de evaluador lee la entrada en el siguiente formato:

- línea 1: el número total de comandos y preguntas en la entrada;
- en cada una de las siguientes líneas:
 - T seguido por un espacio y una letra minúscula para un comando `TypeLetter`;
 - U seguido por un espacio y un entero para `UndoCommands`;
 - P seguido por un espacio y un entero para `GetLetter`.

El ejemplo de evaluador imprimirá los caracteres retornados por `GetLetter`, cada uno en una línea separada.