

Flusskrebsschreiber

Einige Leute sagen dass Leonardo ein grosser Bewunderer von Johannes Gutenberg war, dem deutschen Schmied, der den Druck mittels beweglicher Drucktypen erfand. Er würdigte ihn, indem er ein Schreibgeraet namens Flusskrebsschreiber - "il gambero scrivano" - entwarf. Das Geraet aehnt einer einfachen modernen Schreibmaschine und akzeptiert nur zwei Befehle: der eine schreibt den naechsten Buchstaben und der andere macht die letzten Befehle rueckgaengig (eng.: Undo). Das besondere Merkmal des Flusskrebsschreibers ist, dass der Undo Befehl sehr maechtig ist: ein Undo gilt selber als Befehl und kann deshalb auch rueckgaengig gemacht werden.

Problemstellung

Deine Aufgabe ist es, den Flusskrebsschreiber als Programm realisieren: es startet mit einem leeren Text und akzeptiert eine vom Benutzer eingegebene Befehlsfolge. Ausserdem kann man abfragen, welcher Buchstabe an einer bestimmten Textposition steht.

- `Init()` — wird einmal beim Programmstart aufgerufen, immer ohne Argument. Dieser Befehl kann genutzt werden, um Datenstrukturen anzulegen. Dieser Befehl wird nie rueckgaengig gemacht.
- `TypeLetter(L)` — haengt einen einzelnen kleingeschriebenen Buchstaben `L` (`a`, ..., `z`) ans Ende des Textes an.
- `UndoCommands(U)` — macht die letzten `U` Befehle rueckgaengig, `U` ist eine positive ganze Zahl.
- `GetLetter(P)` — gibt den Buchstaben an Position `P` im aktuellen Text zurueck, `P` ist ein nicht-negativer Index. Der erste Buchstabe im Text hat den Index 0. (Diese Abfrage gilt nicht als Befehl und wird deshalb vom Undo Befehl komplett ignoriert.)

Nach dem initialen Aufruf von `Init()` koennen die anderen Befehle gar nicht oder mehrmals in beliebiger Reihenfolge aufgerufen werden. Es ist sichergestellt, dass `U` nie die Anzahl der vorherigen Befehle ueberschreitet und dass `P` kleiner als die aktuelle Textlaenge (die Anzahl der Buchstaben im aktuellen Text) ist.

Der `UndoCommands(U)` Befehl macht die vorherigen `U` Befehle in "umgekehrter" Reihenfolge rueckgaengig: wird der `TypeLetter(L)` Befehl rueckgaengig gemacht dann wird der Buchstabe `L` vom Ende des aktuellen Textes entfernt; wird der `UndoCommands(X)` Befehl mit Wert `X` rueckgaengig gemacht, dann werden die `X` vorherigen Befehle in ihrer "originalen" Reihenfolge erneut ausgefuehrt.

Beispiel

Das folgende Beispiel zeigt eine moegliche Reihenfolge an Befehlen zusammen mit dem entsprechenden Text nach jedem Befehlsaufruf.

Aufruf	Rueckgabewert	Aktueller 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

Subtask 1 [5 Punkte]

- Die Gesamtanzahl an Befehlen und Abfragen liegt zwischen 1 und 100 (inklusiv) und es werden keine `UndoCommands` Befehle aufgerufen.

Subtask 2 [7 Punkte]

- Die Gesamtanzahl an Befehlen und Abfragen liegt zwischen 1 und 100 (inklusiv) und es werden keine `UndoCommands` Befehle rueckgaengig gemacht.

Subtask 3 [22 Punkte]

- Die Gesamtanzahl an Befehlen und Abfragen liegt zwischen 1 und 5 000 (inklusiv).

Subtask 4 [26 Punkte]

- Die Gesamtanzahl an Befehlen und Abfragen liegt zwischen 1 und 1 000 000 (inklusiv). Alle Aufrufe von `GetLetter` Befehlen finden nach allen Aufrufen von `TypeLetter` und `UndoCommands` Befehlen statt.

Subtask 5 [40 Punkte]

- Die Gesamtanzahl an Befehlen und Abfragen liegt zwischen 1 und 1 000 000 (inklusiv).

Implementierungsdetails

Du musst genau eine Datei einreichen. Diese Datei muss `scrivener.c`, `scrivener.cpp` oder `scrivener.pas` benannt sein. Diese Datei muss die oben beschriebenen Unterprogramme implementieren und die folgenden Signaturen respektieren:

C/C++ Programme

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

Pascal Programme

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

Diese Unterprogramme muessen sich wie oben beschrieben verhalten. Es ist dir natuerlich freigestellt, andere Unterprogramme zur internen Anwendung anzulegen. Dein eingeschicktes Programm darf in keinsten Weise mit der Standard Ein- und Ausgabe oder mit irgendeiner Datei interagieren.

Beispielgrader

Der Beispielgrader liest die Eingabe im folgenden Format:

- Zeile 1: die Anzahl an Befehlen in der Eingabe;
- in jeder der folgenden Zeilen befindet sich entweder:
 - T gefolgt von einem Leerzeichen und einem kleingeschriebenen Buchstaben fuer einen `TypeLetter` Befehl;
 - U gefolgt von einem Leerzeichen und einer Ganzzahl fuer einen `UndoCommands` Befehl;
 - P gefolgt von einem Leerzeichen und einer Ganzzahl fuer einen `GetLetter` Befehl.

Der Beispielgrader wird die, vom `GetLetter` Befehl zurueckgegebenen Zeichen ausgeben, ein Zeichen pro Zeile.