

Kreeftlettertang

Sommige mensen zeggen dat Leonardo een grote bewonderaar was van Johannes Gutenberg, de Duitse smid die het moderne drukwerk uitvond, en dat hij hem eerde door een machine uit te vinden die de kreeftlettertang werd genoemd - *il gambero scrivano*- een uiterst simpel apparaat. Het apparaat lijkt wat op een eenvoudige moderne typemachine en kent maar twee opdrachten: Eén waarmee het volgende teken wordt gedrukt en één om de laatste opdrachten ongedaan te maken. Een opmerkelijke eigenschap van de kreeftlettertang is dat het ongedaan maken een buitengewoon krachtige opdracht is; een ongedaan maken wordt zelf ook beschouwd als een opdracht, en kan daarom ongedaan worden gemaakt.

Opdracht

Jij moet een software versie van de kreeftlettertang maken: Die begint met een lege tekst en ontvangt een reeks opdrachten van de gebruiker, en vragen naar specifieke posities in de huidige versie van de tekst. Dat gaat als volgt.

- `Init()` — Dit wordt slechts eenmaal aan het begin van de uitvoering van het programma aangeroepen, zonder argumenten. Het kan worden gebruikt om datastructuren te initialiseren. Deze opdracht wordt nooit ongedaan gemaakt.
- `TypeLetter(L)` — voeg aan het einde van de tekst een enkele kleine letter `L`, gekozen uit de letters van `a, ..., z`.
- `UndoCommands(U)` — maak de laatste `U` opdrachten ongedaan, voor een positie geheel getal `U`.
- `GetLetter(P)` — geeft de letter weer die op positie `P` staat in de huidige tekst, voor een niet-negatief getal `P`. De eerste letter in de tekst heeft nummer 0. (Deze query is geen opdracht en wordt daarom genegeerd door de undo-opdracht.)

Na de eerste aanroep van `Init` kunnen de andere routines kunnen nul of meer keren worden aangeroepen, in een willekeurige volgorde. `U` zal nooit groter zijn dan het aantal eerder ontvangen opdrachten; `P` is altijd kleiner dan de lengte van de tekst op dat moment.

Voor `UndoCommands(U)`, geldt dat het de vorige `U` opdrachten achterstevoren ongedaan maakt: als de opdracht die ongedaan moet worden gemaakt `TypeLetter(L)` is, dan wordt `L` aan het einde de huidige tekst weggehaald; als de opdracht die ongedaan moet worden gemaakt `UndoCommands(X)` is voor één of andere waarde van `X`, dan worden de vorige `X` opdrachten in hun oorspronkelijke volgorde weer uitgevoerd.

Voorbeeld

We geven een mogelijke reeks opdrachten, met de toestand van de tekst na iedere aanroep.

Aanroep	Resultaat	Huidige tekst
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 punten]

- Het totaal aantal opdrachten en queries is niet groter dan 100 en er zijn geen aanroepen van `UndoCommands`.

Subtask 2 [7 punten]

- Het totaal aantal opdrachten is niet groter dan 100 en er wordt geen `UndoCommands` ongedaan gemaakt.

Subtask 3 [22 punten]

- Het totaal aantal opdrachten en queries is niet groter dan 5 000.

Subtask 4 [26 punten]

- Het totaal aantal opdrachten en queries is niet groter dan 1 000 000. Alle aanroepen van `GetLetter` komen pas na alle aanroepen van `TypeLetter` en `UndoCommands`.

Subtask 5 [40 punten]

- Het totaal aantal opdrachten en queries is niet groter dan 1 000 000.

Implementatie details

Je moet precies één programmabestand inzenden, genaamd `scrivener.c`, `scrivener.cpp` of `scrivener.pas`. Dit bestand moet de deelprogramma's hierboven implementeren met de volgende kenmerken.

C/C++ programmas

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

Pascal programma's

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

Deze deelprogramma's moeten werken zoals hierboven beschreven. Natuurlijk staat het je vrij andere deelprogramma's te maken voor intern gebruik. Je inzendingen mogen op geen enkele manier ingrijpen op standard input en standard output, of met andere bestanden.

Voorbeeld grader

De voorbeeld grader leest de invoer in het volgende format:

- regel 1: het totaal aantal opdrachten en queries in de invoer; * op iedere volgende regel: ** T gevolgd door een spatie en een kleine letter voor een `TypeLetter` opdracht; ** U gevolgd door een spatie en een integer voor een `UndoCommands`; ** P gevolgd door een spatie en een integer voor een `GetLetter`.

De voorbeeld grader zal de tekens die door `GetLetter` worden geretourneerd elk op een aparte regel afdrukken.