

Kreeftschrijver

Sommige mensen zeggen dat Leonardo een groot bewonderaar was van Johannes Gutenberg, de Duitse smid die het moderne drukwerk uitvond, en dat hij te zijner ere een machine uitvond die de kreeftschrijver werd genoemd - *il gambero scrivano* - een heel eenvoudig soort typemachine. Het lijkt wat op onze moderne typemachine en kent maar twee instructies: Eén waarmee het volgende karakter wordt gedrukt en één om de laatste opdrachten ongedaan te maken (undo). Een opmerkelijke eigenschap van de kreeftschrijver is dat het ongedaan maken een buitengewoon krachtige instructie is: het ongedaan maken wordt ook beschouwd als een opdracht, en kan daarom zelf ongedaan worden gemaakt.

Opdracht

Je moet een software-versie van de kreeftschrijver maken: Die begint met een lege tekst en ontvangt een reeks opdrachten van de gebruiker, alsook vragen naar specifieke posities in de huidige versie van de tekst (queries). Dat gaat als volgt.

- `Init()` — wordt slechts éénmaal 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` toe, gekozen uit de letters `a, ..., z`.
- `UndoCommands(U)` — maak de laatste `U` opdrachten ongedaan. `U` is een positief geheel getal.
- `GetLetter(P)` — geeft de letter weer die op positie `P` staat in de huidige tekst, waarbij `P` een niet-negatieve index is. De eerste letter in de tekst heeft index 0. (Dit is een query, geen opdracht, en wordt daarom genegeerd door de undo-opdracht.)

Na de eerste aanroep van `Init()`, kunnen de andere functies nul of meer keren worden aangeroepen in willekeurige volgorde. Je hebt de garantie dat `U` nooit groter zal zijn dan het aantal eerder ontvangen opdrachten, en dat `P` altijd kleiner is dan de lengte (het aantal letters) van de huidige tekst.

`UndoCommands(U)` maakt de vorige `U` opdrachten ongedaan *in omgekeerde volgorde* : als de ongedaan gemaakte opdracht `TypeLetter(L)` is, dan verwijdert het `L` van het einde van de huidige tekst; als de ongedaan gemaakte opdracht `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 daarbij de toestand van de tekst na iedere functie-aanroep.

Instructie	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

Subtaak 1 [5 punten]

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

Subtaak 2 [7 punten]

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

Subtaak 3 [22 punten]

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

Subtaak 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`.

Subtaak 5 [40 punten]

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

Implementatiedetails

Je moet precies één bestand indienen, genaamd `scrivener.c`, `scrivener.cpp` of `scrivener.pas`. Dit bestand moet de gevraagde functies implementeren volgens deze declaraties:

C/C++ programma's

```
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 functies moeten werken zoals hierboven werd beschreven. Natuurlijk staat het je vrij bijkomende functies te implementeren voor intern gebruik. Je submissions mogen op geen enkele manier interageren met standaard input/output, of met andere bestanden.

Voorbeeld-grader

De voorbeeld-grader leest de invoer in het volgende formaat:

- 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` opdracht;
 - P gevolgd door een spatie en een integer voor een `GetLetter` opdracht.

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