Crayfish scrivener

Se spune că Leonardo a fost un mare admirator al lui Johannes Gutenberg, fierarul german care a inventat tiparul și în onoarea sa a proiectat un dispozitiv numit editorul rac - crayfish scrivener - un dispozitiv simplu pentru scris. El este similar cu mașina de scris modernă și acceptă doar două comenzi: una pentru a scrie următorul caracter și una de anulare (undo) a ultimelor comenzi. Cea mai importantă proprietate a editorului rac este că undo este o comandă foarte puternică: ea este considerată o comandă în sine și poate fi la rândul ei anulată.

Enunț

Sarcina ta este să realizezi o versiune software a editorului rac: se începe cu un text vid și acceptă o succesiune de comenzi introduse de utilizator, și întrebări pentru anumite poziții din versiunea curentă a textului, după cum urmează.

- `Init`() - se apelează o singură dată la începutul execuției, fără argumente. Se utilizează pentru inițializarea structurilor de date. Aceasta nu va trebui anulată niciodată.
- `TypeLetter`(L) — adaugă la sfârșitul textului o singură literă mică L din intervalul `a`, …, `z`.
- `UndoCommands`(U) — anulează ultimele U comenzi, unde U este un număr intreg pozitiv.
- `GetLetter`(P) — returnează litera de la poziția P din textul curent, unde P este un indice nenegativ. Prima literă din text are indicele 0. (Această întrebare nu este o comandă și este ignorată de comanda undo).

După apelul inițial al `Init`, celelalte rutine pot fi apelate de zero sau mai multe ori în orice ordine. Se garantează că U nu va depăși numărul de comenzi primite anterior, și că P va fi mai mic decât lungimea textului curent (numărul de litere al textului curent).

În ceea ce privește `UndoCommands`(U), ea anulează ultimele U comenzi în ordine *inversă*: dacă comanda ce trebuie anulată este `TypeLetter`(L), atunci ea elimină litera L de la sfârșitul textului curent; dacă comanda ce trebuie anulată este `UndoCommands`(X) pentru o valoare X, aceasta reface ultimele X comenzi în ordinea lor "originală".

Exemplu

Vă vom arăta o secvență posibilă de apeluri, împreună cu configurația textului după fiecare apel.

Apel	Returnează	Textul curent
`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 puncte]

- Numărul de comenzi și întrebări este între 1 și 100 (inclusiv) și nu va exista niciun apel `UndoCommands`.

Subtask 2 [7 puncte]

- Numărul de comenzi și întrebări este între 1 și 100 (inclusiv) și niciun apel `UndoCommands` nu va fi anulat.

Subtask 3 [22 puncte]

- Numărul de comenzi și întrebări este între 1 și 5 000 (inclusiv).

Subtask 4 [26 de puncte]

- Numărul de comenzi și întrebări este între 1 și 1 000 000 (inclusiv). Toate apelurile `GetLetter` vor apărea după toate apelurile `TypeLetter` și `UndoCommands`.

Subtask 5 [40 de puncte]

- Numărul de comenzi și întrebări este între 1 și 1 000 000 (inclusiv).

Detalii de implementare

Trebuie să trimiți exact un fișier, numit `scrivener.c`, `scrivener.cpp` sau `scrivener.pas`. Acest fișier trebuie să implementeze subprogramele descrise mai sus folosind următoarele semnături.

programele în limbajul C/C++

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

programele în limbajul Pascal

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

Aceste subprograme trebuie să se comporte aşa cum este descris mai sus. Desigur, eşti liber să implementezi pentru uzul intern al acestora şi alte subprograme. Submit-urile tale nu trebuie să interacţioneze în nici într-un fel cu intrarea / ieşirea standard, şi nici cu oricare alt fişier.

Modelul de evaluator

Modelul de evaluator citește intrarea în formatul ce urmează:

- linia 1: numărul total de comenzi și întrebări din input;
- pe fiecare dintre următoarele linii:
  - `T` urmat de un spațiu și o literă mică a pentru comanda `TypeLetter`;
  - `U` urmat de un spațiu și un număr întreg pentru comanda `UndoCommands`;
  - `P` urmat de un spațiu și un număr întreg pentru comanda `GetLetter`.

Modelul de evaluator va scrie caracterele returnate de `GetLetter`, fiecare pe câte o linie.