



ザリガニの代書 (Crayfish scrivener)

レオナルド (Leonardo) はヨハネス・グーテンベルク (Johannes Gutenberg, 活版印刷術を発明したドイツの鍛冶屋) を賞賛しており, 「ザリガニの代書」 — *il gambero scrivano* — と呼ばれる単純なタイピング装置を設計することによって敬意を払ったと言われている. これは現代の単純なタイプライターに似ていて, 2 個のコマンドのみを受け付ける. 1 つは次の文字を入力するコマンドであり, もう 1 つは `undo` と呼ばれる最近のコマンドを取り消すコマンドである. ザリガニの代書の特徴は `undo` がとても強力なことにある: `undo` も 1 つのコマンドとみなされ, `undo` を `undo` することができる.

問題文 (Statement)

あなたの課題はザリガニの代書のソフトウェア版を実現することである. 空のテキストからスタートして, ユーザが入力したコマンドの列を受け付け, 現在のテキストのある位置の文字を尋ねるクエリに以下のように答える.

- `Init()` — 実行の最初に 1 回だけ引数なしで呼ばれる. これはデータ構造の初期化に使うことができる. これは決して `undo` されることはない.
- `TypeLetter(L)` — テキストの末尾に小文字 `a`, ..., `z` のうちの 1 文字 `L` を 1 個加える.
- `UndoCommands(U)` — 正整数 `U` に対し, 最近の `U` 個のコマンドを `undo` する.
- `GetLetter(P)` — 非負整数 `P` に対し, 現在のテキストの `P` 番目の文字を返す. テキストの最初の文字は 0 番目と数える. このクエリはコマンドではないので, `undo` コマンドには無視される.

最初に `Init()` が呼ばれた後は, 他のルーチンは任意の順番で 0 回以上呼ばれる可能性がある. `U` が過去に受け付けたコマンドの個数を超えないことと, `P` が現在のテキストの長さ (文字の個数) より小さいことが保証されている.

`UndoCommands(U)` は, 最近の `U` 個のコマンドの `undo` を逆順に行う: `undo` するコマンドが `TypeLetter(L)` であれば, `L` を現在のテキストの末尾から取り除く. `undo` するコマンドが `UndoCommands(X)` であれば, 直前の `X` 個のコマンドを元の順番でやり直す.

例 (Example)

可能性のある呼び出しの列の例を、各呼び出し後のテキストの状態とともに示す。

呼び出し	返り値	現在のテキスト
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

小課題 1 [5 点]

- コマンドとクエリの回数の合計は 1 以上 100 以下であり、UndoCommands は呼び出されない。

小課題 2 [7 点]

- コマンドとクエリの回数の合計は 1 以上 100 以下であり、UndoCommands が undo されることはない。

小課題 3 [22 点]

- コマンドとクエリの回数の合計は 1 以上 5 000 以下である。

小課題 4 [26 点]

- コマンドとクエリの回数の合計は 1 以上 1 000 000 以下である。すべての GetLetter は、すべての TypeLetter と UndoCommands が呼び出された後に呼び出される。

小課題 5 [40 点]

- コマンドとクエリの実行回数の合計は 1 以上 1 000 000 以下である。

実装の詳細 (Implementation details)

あなたは `scrivener.c`, `scrivener.cpp` または `scrivener.pas` という名前のファイルをちょうど 1 つ提出しなければならない。そのファイルは上記で説明されたサブプログラムを以下のシグネチャを用いて実装しなければならない。

C/C++ プログラム

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

Pascal プログラム

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

これらのサブプログラムは上記で説明された通りに動作しなければならない。もちろん、内部での使用のために他のサブプログラムを実装することは自由である。あなたの提出は標準入力・標準出力、あるいは他のファイルといかなる方法でもやりとりしてはならない。

採点プログラムのサンプル (Sample grader)

採点プログラムのサンプルは以下の書式の入力を読み込む：

- 1 行目： 入力中のコマンドとクエリの個数の合計。
- 続く各行は、以下のいずれかである：
 - T と 1 個の空白と `TypeLetter` コマンドのための 1 個の小文字、または
 - U と 1 個の空白と `UndoCommands` コマンドのための 1 個の整数、または
 - P と 1 個の空白と `GetLetter` のための 1 個の整数。

採点プログラムのサンプルは、`GetLetter` の返した文字を、1 行に 1 個ずつ出力する。