

Рак - препишувач

Некои луѓе велат дека Леонардо бил голем обожавател на Јохан Гутенберг, германски ковач кој го измислил подвижното печатење, и кој што оддал почит дизајнирајќи машина наречена рак препишувач (*il gambero scrivano*) - многу едноставен уред за типкање. На некој начин, таа е слична на едноставната модерна машина за пишување и прифаќа само две команди: една за пишување на следниот знак и една за однаправување на најскорешните команди. Значајна особина на рак препишувачот е тоа што командата за однаправување (*undo*) е екстремно моќна: самата *undo* се смета за команда, па може да биде и однаправена.

Задача

Ваша задача е да направите софтверска верзија на рак-препишувач: започнува со празен текст и прифаќа секвенца од команди внесени од страна на корисникот, и пребарува специфични позиции во тековната верзија на текстот, како што е опишано подолу.

- `Init()` — се повикува еднаш на почетокот на извршувањето, без аргументи. Може да се користи за иницијализација на податочни структури. Никогаш нема да се јави потреба за однаправување на оваа операција.
- `TypeLetter(L)` — на крајот на текстот надоврзи една мала буква `L` избрана од `a`, ..., `z`.
- `UndoCommands(U)` — однаправи ги последните `U` команди, за даден позитивен цел број `U`.
- `GetLetter(P)` — врати ја буквата на позиција `P` во тековниот текст, за даден ненегативен индекс `P`. Првата буква во текстот има индекс `0`. (Ова барање не е команда и според тоа се игнорира од страна на командата *undo*.)

По почетниот повик до `Init()`, другите рутини може да се повикаат нула или повеќе пати во произволен редослед. Се гарантира дека `U` нема да го надмине бројот на претходно примени команди, и дека `P` ќе биде помало од должината на тековниот текст (бројот на букви во тековниот текст).

Кога станува збор за `UndoCommands(U)`, таа ги однаправува претходните `U` команди во *обратен* редослед: ако командата која треба да се однаправи е `TypeLetter(L)`, тогаш истата ја отстранува `L` од крајот на тековниот текст; ако командата која треба да се однаправи е `UndoCommands(X)`, за некоја вредност `X`, истата повторно ги извршува претходните `X` команди во нивниот *оригинален* редослед.

Пример

Дадена е можна секвенца од повици, заедно со состојбата на текстот после секој од повиците.

Повик	Враќа	Тековен (моментален) текст
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

Подзадача 1 [5 поени]

- Вкупниот број на команди и барања е помеѓу 1 и 100 (вклучително) и нема да има повици до UndoCommands.

Подзадача 2 [7 поени]

- Вкупниот број на команди и барања е помеѓу 1 и 100 (вклучително) и нема да има ниту едно однаправување на UndoCommands.

Подзадача 3 [22 поени]

- Вкупниот број на команди и барања е помеѓу 1 и 5 000 (вклучително).

Подзадача 4 [26 поени]

- Вкупниот број на команди и барања е помеѓу 1 и 1 000 000 (вклучително). Сите повици до GetLetter ќе се појават после сите повици до TypeLetter и UndoCommands.

Подзадача 5 [40 поени]

- Вкупниот број на команди и барања е помеѓу 1 и 1 000 000 (вклучително).

Имплементациски детали

Треба да предадете точно еден документ, со име `scrivener.c`, `scrivener.cpp` или `scrivener.pas`. Овој документ треба да ги содржи имплементациите на потпрограмите опишани погоре користејќи ги следниве потписи.

C/C++ програми

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

Pascal програми

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

Овие потпрограми мора да се однесуваат како што е опишано погоре. Се разбира, дозволено е да имплементирате и други потпрограми за нивна внатрешна употреба. Вашите предадени решенија не смеат да имаат никаков тип на интеракција ниту со стандардниот влез/излез, ниту пак со која било друга датотека.

Пример-оценувач

Пример-оценувачот чита влез во следниот формат:

- линија 1: вкупниот број на команди и барања во влезот;
- во секоја следна линија:
 - Т по кој што следува празно место и мала буква за `TypeLetter` команда;
 - U по кој што следува празно место и цел број за `UndoCommands`;
 - P по кој што следува празно место и цел број за `GetLetter`.

Пример-оценувачот ќе ги испечати знаците вратени од функцијата `GetLetter`, секој знак во посебна линија.