

Crayfish scrivener

Разправят, че Леонардо бил голям почитател на германския ковач Йоханес Гутенберг, който изобретил печатарската машина, и в негова чест проектирал пишеща машина наречена *il gambero scrivano* — много просто печатащо устройство, приличащо на съвременна пишеща машина и приемащо две команди – отпечатване на поредната буква и отмяна на последната изпълнена команда. Особеност на устройството е, че командата за отмяна е много мощна и може да отмени дори току що изпълнени себеподобни.

Задача

Напишете програма, която да симулира работата на пишещата машина на Леонардо: започва с празен буфер, приема и изпълнява поредица от команди и заявки, въведени от потребителя, за поддържане на текущата версия на текста както следва:

- `Init()` - заявка, която се извиква веднъж в началото на изпълнението на програмата, без аргументи. Може да бъде използвана за инициализация на структурите от данни на програмата. Няма да се наложи да се отменя никога.
- `TypeLetter(L)` — добавя в края на текста малката латинска буква `L`, зададена като аргумент и принадлежаща на `a, ..., z`.
- `UndoCommands(U)` — отменя последните `U` команди, където `U` е положително цяло число.
- `GetLetter(P)` — връща буквата, която се намира на позиция `P` в текущия текст, където `P` е неотрицателен индекс. Първата буква на текста е с индекс `0`. (Тази заявка не е команда и следователно не се отменя от последваща команда `UndoCommands(U)`)

Гореспоменатите три функции могат да бъдат извиквани по време на изпълнението нула или повече пъти в произволен ред. Гарантирано е, че `U` няма никога да надхвърля броя на изпълнените и неотменени до момента команди, както и че `P` е по-малко от текущата дължина на текста (т.е. от броя на буквите в буфера на програмата).

Командата, `UndoCommands(U)` отменя последните неотменени `U` команди в обратен ред: ако командата която трябва да се отмени е `TypeLetter(L)`, тогава програмата премахва намиращата се в края на текста буква `L`; ако командата която трябва да се отмени е `UndoCommands(X)` за някаква стойност на `X`, програмата изпълнява отново последните отменени `X` команди в реда по който са били изпълнени.

Пример

По-долу е показана една възможна последователност от заявки, заедно със състоянието на текста след изпълнение на всяка от тях.

Заявка	Връща	Текущ 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 точки]

- Общият брой на командите и заявките е между 1 и 100 (включително) и няма заявки от вида UndoCommands.

Подзадача 2 [7 точки]

- Общият брой на командите и заявките е между 1 и 100 (включително) и няма отмяна на заявки от вида UndoCommands.

Подзадача 3 [22 точки]

- Общият брой на командите и заявките е между 1 и 5000 (включително).

Подзадача 4 [26 точки]

- Общият брой на командите и заявките е между 1 и 1 000 000 (включително). Всички извиквания на GetLetter ще бъдат след всички извиквания на TypeLetter и UndoCommands.

подзадачи 5 [40 точки]

- Общият брой на командите и заявките е между 1 and 1 000 000 (включително).

Детайли на имплементацията

Трябва да изпратите точно един файл, наречен `scrivener.c`, `scrivener.cpp` или `scrivener.pas`, който имплементира описаните по-горе функции, специфицирани както следва:

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;
```

Тези функции трябва да работят както е описано по-горе. Не е забранено да напишете свои подпрограми, според нуждите на Вашата имплементация. Изпратеното решение не бива да има нищо общо със стандартния вход/изход, или с който и да било друг файл.

Прост грейдър

Простият грейдър чете входа в следния формат:

- ред 1: общият брой на командите и заявките във входа;
- на всеки от следващите редове:
 - T последвана от интервал и малка латинска буква за командата `TypeLetter`;
 - U последвана от интервал и положително число за командата `UndoCommands`;
 - P последвана от интервал положително число за командата `GetLetter`.

Простият грейдър ще отпечата знаците връщани от `GetLetter`, по един знак на ред.