

Rak písář

Johannes Gutenberg byl německý kovář, který vynalezl knihtisk. Říká se, že Leonardo byl Gutenbergovým velkým obdivovatelem a na jeho počest navrhl předchůdce psacího stroje – il gambero scrivano. Tento stroj přijímá dva příkazy – napsat znak a vrátit zpět několik předchozích příkazů. Rafinovanost tohoto stroje spočívá v tom, že vrátit zpět lze i příkaz vrácení zpět.

Zadání

Implementujte softwarovou verzi popsaného stroje. Výpočet začíná s prázdným řetězcem. Program postupně přijímá posloupnost příkazů, které řetězec mění, a dotazy na písmena na vybraných pozicích v aktuálním řetězci.

- **Init()** – je zavolán jednou na začátku výpočtu, bez parametrů. Můžete ho využít k inicializaci datových struktur. Nebude nikdy vrácen zpět.
- **TypeLetter(L)** – připojí na konec řetězce jedno malé písmeno **L**.
- **UndoCommands(U)** – vrátí zpět posledních **U** příkazů, kde **U** je kladné celé číslo.
- **GetLetter(P)** – vrátí písmeno na pozici **P** v aktuálním řetězci, pro nezáporné celé **P**. První písmeno řetězce má pozici 0. Dotaz **GetLetter** se nepočítá mezi příkazy, **UndoCommands** ho proto nevrací zpět.

Po počátečním zavolání **Init()** mohou být ostatní funkce volány libovolně mnohokrát v libovolném pořadí. Máte zaručeno, že **U** je vždy menší nebo rovno počtu předtím provedených příkazů a že **P** je menší než délka aktuálního řetězce.

UndoCommands(U) vrací zpět předchozích **U** příkazů v opačném pořadí. Je-li vrácen zpět příkaz **TypeLetter(L)**, odstraní se písmeno **L** z konce řetězce. Je-li vrácen zpět příkaz **UndoCommands(X)**, znovu se provede **X** jemu předcházejících příkazů v původním pořadí.

Příklad

Toto je jedna možná posloupnost volání funkcí, spolu se stavem řetězce po zavolání každé z nich.

Volání	Vrácená hodnota	Řetězec
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

Podúloha 1 [5 bodů]

Celkový počet příkazů a dotazů je mezi 1 a 100 včetně a příkaz `UndoCommands` není použit.

Podúloha 2 [7 bodů]

Celkový počet příkazů a dotazů je mezi 1 a 100 včetně a žádný příkaz `UndoCommands` není vrácen zpět.

Podúloha 3 [22 bodů]

Celkový počet příkazů a dotazů je mezi 1 a 5 000 včetně.

Podúloha 4 [26 bodů]

Celkový počet příkazů a dotazů je mezi 1 a 1 000 000 včetně. Všechny dotazy `GetLetter` následují až po všech příkazech `TypeLetter` a `UndoCommands`.

Podúloha 5 [40 bodů]

Celkový počet příkazů a dotazů je mezi 1 a 1 000 000 včetně.

Implementace

Odevzdejte právě jeden soubor pojmenovaný `scrivener.c`, `scrivener.cpp` nebo `scrivener.pas`. Tento soubor implementuje výše popsané funkce s následujícími deklaracemi.

C/C++:

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

Pascal:

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

Samozřejmě také můžete používat pomocné procedury a funkce. Vaše řešení nesmí číst ze standardního vstupu, vypisovat na standardní výstup ani pracovat se soubory.

Ukázkový vyhodnocovač

Ukázkový vyhodnocovač čte vstup v následujícím formátu.

první řádka: celkový počet příkazů a dotazů

na každé další řádce:

T následované mezerou a malým písmenem pro příkaz `TypeLetter`;

U následované mezerou a celým číslem pro příkaz `UndoCommands`;

P následované mezerou a celým číslem pro dotaz `GetLetter`.

Ukázkový vyhodnocovač vypíše odpovědi na dotazy `GetLetter`, každou z nich na samostatný řádek.