

Rakov pisar

Neki tvrde da je Leonardo bio veliki ljubitelj Johannesa Gutenberga, njemačkog kovača koji je izmislio pokretni tipkarski sustav. Leonardo je tako odao počast Gutenbergu tako što je izmislio stroj nazvan rakov pisar — *il gambero scrivano* — jednostavan pisaći stroj. Rakov pisar je sličan današnjoj pisaćoj mašini, ali prima samo dvije naredbe: jednu da doda znak na kraj niza i jednu da poništi (undo) određen broj prethodnih naredbi. Posebno je bilo to što je naredba poništenja mogla čak i poništiti drugu naredbu poništavanja, pošto su one i same naredbe, i kao takve mogu biti poništene.

Zadatak

Vaš zadatak je da ostvarite programsku verziju rakovog pisara. Pisar na početku na ulazu ima prazan niz znakova i prihvaća niz naredbi od korisnika, kao i upite za dohvat znakova sa određenih pozicija teksta, kao što slijedi:

- `Init()` — zove se jednom na početku programa, bez argumenata. Možete koristiti za inicijalizaciju struktura podataka. Ovu naredbu nikad nećete morati poništiti.
- `TypeLetter(L)` — dodaj znak `L` na kraj teksta. `L` može biti jedan od malih slova engleske abecede, tj. `a`, ..., `z`.
- `UndoCommands(U)` — poništi zadnjih `U` naredbi. `U` je pozitivan cijeli broj.
- `GetLetter(P)` — vrati znak na poziciji `P` u trenutnom tekstu, za neki ne-negativni indeks `P`. Prvi znak u tekstu ima indeks `0`. Ovaj upit nije naredba i ne utječe na naredbu poništavanja.

Nakon početnog poziva funkciji `Init()`, ostale funkcije će biti pozvane nula ili više puta bilo kojim redom. Garantirano je da `U` nikad neće biti veći od broja prethodno postavljenih naredbi, a `P` će uvijek biti manji od duljine trenutnog teksta (broj slova u tekstu).

Što se tiče naredbe `UndoCommands(U)`, ona poništava zadnjih `U` naredbi u *obrnutom* poretku: ako je naredba koju treba poništiti `TypeLetter(L)`, onda će maknuti `L` sa kraja teksta; ako je naredba koju treba poništiti `UndoCommands(X)`, onda će ponoviti tih prethodnih `X` naredbi u *originalnom* poretku.

Primjer

U nastavku prikazujemo mogući niz poziva, kao i stanje teksta nakon svakog poziva.

Poziv	Povratna vrijednost	Trenutni tekst
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

Podzadatak 1 [5 bodova]

- Ukupni broj naredbi i upita će biti između 1 i 100 (uključivo) i neće biti poziva funkciji `UndoCommands`.

Podzadatak 2 [7 bodova]

- Ukupni broj naredbi i upita će biti između 1 i 100 (uključivo), a nijedna naredba `UndoCommands` neće biti poništena.

Podzadatak 3 [22 bodova]

- Ukupni broj naredbi i upita će biti između 1 i 5 000 (uključivo).

Podzadatak 4 [26 bodova]

- Ukupan broj naredbi i upita će biti između 1 i 1 000 000 (uključivo). Svi upiti `GetLetter` će biti nakon svih naredbi `TypeLetter` i `UndoCommands`.

Podzadatak 5 [40 bodova]

- Ukupan broj naredbi i upita će biti između 1 i 1 000 000 (uključivo).

Implementacija

Morate poslati samo jednu datoteku, nazvanu `scrivener.c`, `scrivener.cpp` ili `scrivener.pas`. Ta datoteka mora implementi gore opisane funkcije koristeći sljedeće deklaracije.

C/C++ programi

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

Pascal programi

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

Ovi potprogrami se moraju ponašati prema gore opisanome. Naravno, za vaše potrebe možete implementirati druge pomoćne potprograme. Imajte na umu da vaše rješenje ne smije koristiti standardni ulaz te izlaz, kao i bilo kakve druge datoteke.

Primjer biblioteke za testiranje

Biblioteka za testiranje očekuje sljedeći ulazni format:

- linija 1: ukupan broj naredbi i upita;
- u svakoj sljedećoj liniji:
 - T popraćeno sa razmakom i malim slovom za naredbu `TypeLetter`;
 - U popraćeno sa razmakom i prirodnim brojem za naredbu `UndoCommands`;
 - P popraćeno sa razmakom i nenegativnim cijelim brojem za naredbu `GetLetter`.

Biblioteka će ispisati sve znakove koje vrate pozivi iz `GetLetter`, svaku u zasebnu liniju.