

# Rak prepisivač

Neki ljudi kažu da je Leonardo bio veliki poštovalac Johannes Gutenberg-a, nemačkog kovača koji je izmislio pokretni-tip štampanja, i da mu treba odati počast za dizajniranje mašine koja se zove rak prepisivač — *il gambero scrivano* —, vrlo jednostavan uređaj za kucanje. Sličan je jednostavnoj modernoj mašini za kucanje i prihvata samo dvije komande: jedna da se unese sledeći karakter i jedna da poništi najnovije komande. Značajna karakteristika date mašine je da je 'undo' komanda izuzetno moćna: 'undo' se smatra samom komandom i može se opozvati.

## Podstavka

Tvoj zadatak je da napraviš jednu softversku verziju crayfish scrivener-a: počinje sa praznim tekstom i prihvata sekvencu komandi unešene od user-a, i upite za specifične pozicije trenutne verzije teksta, kao prateće.

`Init()` — se poziva na početku izvršenja, bez argumenata. Može se koristiti za inicijalizaciju strukture podataka. Nikada neće biti potrebno da se poništi.

- `TypeLetter(L)` — dodaj na kraju teksta malo slovo `L` izabrano iz `a, ..., z`.
- `UndoCommands(U)` — 'undo' poslednje komande, za pozitivni integer `U`.
- `GetLetter(P)` — vrati slovo na poziciji `P` u samom tekstu, za ne-negativni indeks `P`. Prvo slovo u tekstu ima indeks 0. (Ovaj upit nije komanda i time se ignoriše 'undo' komanda.)

Nakon inicijalnog poziva na `Init()`, druge rutine mogu pozvati 0 ili više puta, bilo kojim redosledom. Garantuje se da `U` neće premašiti broj prethodno primljenih komandi, i da će `P` biti manje od trenutne dužine teksta (broj slova u tekućem tekstu).

Što se tiče `UndoCommands(U)`, ona poništava prethodne `U` komande u obrnutom redosledu: ako je komanda opozvana sa `TypeLetter(L)`, onda uklanja `L` sa kraja tekućeg teksta; ako je komanda opozvana sa `UndoCommands(X)` za neku vrijednost `X`, ona ponovo radi prethodne `X` komande u svom originalnom poretku.

Primjer

Prikazati mogući redosled poziva, zajedno sa stanjem teksta posle svakog poziva.

Poziv	Vraća	Tekući 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 poena]

Ukupan broj komandi i upita je između 1 i 100 (inkluzivno) neće biti nikakvih poziva na `UndoCommands`.

## Podzadatak 2 [7 poena]

Ukupan broj komandi i upita je između 1 i 100 (inkluzivno) i ne `UndoCommands` će biti poništen.

## Podzadatak 3 [22 poena]

- Ukupan broj komandi i upita je između 1 i 5000 (inkluzivno).

## Podzadatak 4 [26 poena]

- Ukupan broj komandi i upita je između 1 i 1 000 000 (inkluzivno). Svaki poziv ka `GetLetter` će se pojaviti nakon svih poziva na `TypeLetter` i `UndoCommands`.

## Podzadatak 5 [40 poena]

- Ukupan broj komandi i upita je između 1 i 1 000 000 (inkluzivno).

## Implementacijski detalji

Morate potvrditi tačno jedan fajl pod nazivom `scrivener.c`, `scrivener.cpp` ili `scrivener.pas`. Ovaj fajl mora da implementira podprograme koristeći sledeću signaturu.

## 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 trebaju ponašati kao što je opisano gore. Naravno da ste slobodni da koristite i druge pomoćne podprograme. Vaše rješenje ne smije ni na koji način koristiti standardni ulaz i izlaz niti bilo koju drugu datoteku.

## Probni tester

Ulazni podaci za probni test su u sledećem obliku:

- linija 1: ukupan broj komandi i upita u input-u;
- na svakoj sledećoj liniji:
  - T praćen razmakom i malim slovom za TypeLetter komandu;
  - U praćen razmakom i jednim integerom za UndoCommands;
  - P praćen razmakom i jednim integerom za GetLetter.

Probni test će štampati karaktere koje vraća GetLetter, svaki u posebnoj liniji.