

Crayfish scrivener

Beberapa orang mengatakan bahwa Leonardo adalah pengagum Johannes Gutenberg, seorang pandai besi Jerman yang menemukan “movable-type printing”, dan untuk itu dia merancang sebuah mesin yang dinamakan “crayfish scrivener” — il gambero scrivano — sebuah piranti ketik sangat sederhana. Piranti tersebut mirip mesin ketik modern dan hanya menerima dua perintah : satu perintah untuk mengetik karakter berikutnya dan satu untuk membatalkan (“undo”) perintah-perintah terbaru. Fitur penting dari crayfish scrivener adalah bahwa perintah “undo” sangat ampuh: sebuah undo juga merupakan sebuah perintah, yang juga dapat dibatalkan (di-“undo”).

Statement

Tugas Anda adalah merealisasikan versi software dari “crayfish scrivener” : yang dimulai dengan sebuah teks kosong, dan menerima sederetan perintah yang dientry oleh pengguna, dan dapat menjawab pertanyaan mengenai posisi tertentu dari versi teks pada suatu saat.

- `Init()` — dipanggil sekali pada awal eksekusi, tanpa argumen. Dapat dipakai untuk menginisialisasi struktur data. Subrutin ini tak pernah dibatalkan (di-“undo”).
- `TypeLetter(L)` — menambahkan sebuah huruf kecil L pada akhir teks, yang dipilih dari a, ..., z.
- `UndoCommands(U)` — melakukan pembatalan (undo) U perintah terakhir, dengan U adalah sebuah bilangan bulat positif.
- `GetLetter(P)` — mengembalikan huruf yang berada pada posisi P pada teks, dengan P adalah sebuah indeks bernilai tak negatif. Huruf pertama pada teks diberi nilai indeks 0. (query ini bukan perintah maka diabaikan oleh perintah undo).

Setelah pemanggilan pertama `Init()`, routines lainnya dapat dipanggil sebanyak nol atau lebih dengan urutan apapun. Dijamin bahwa U tak akan melebihi banyaknya perintah yang pernah diterima sebelumnya, dan bahwa P akan lebih kecil dari panjang teks saat itu (banyaknya huruf pada teks).

Sedangkan perintah `UndoCommands (U)`, melakukan Undo terhadap sebanyak U buah perintah dengan urutan terbalik: jika perintah yang di-“undo” adalah `TypeLetter(L)`, maka akan menghapus L huruf terakhir dari teks; jika perintah yang akan dibatalkan (undo) adalah `UndoCommands(X)` untuk beberapa nilai X, akan dilakukan re-does terhadap X perintah sebelumnya dengan urutan semula.

Contoh

Kami tunjukkan urutan call yang mungkin, dilengkapi dengan state dari teks setelah sebuah call

Pemanggilan	Pengembalian	Teks
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

Subtask 1 [5 points]

- Banyaknya total perintah dan queries adalah antara 1 dan 100 (inklusif) dan tak ada pemanggilan `UndoCommands`.

Subtask 2 [7 points]

- Banyaknya total perintah dan queries adalah antara 1 dan 100 (inklusif) dan tak ada `UndoCommands` yang dibatalkan.

Subtask 3 [22 points]

- Banyaknya total perintah dan queries adalah antara 1 dan 5 000 (inklusif).

Subtask 4 [26 points]

- Banyaknya total perintah and queries adalah antara 1 dan 1 000 000 (inklusif). Semua pemanggilan `GetLetter` akan terjadi setelah semua pemanggilan `TypeLetter` dan `UndoCommands`.

Subtask 5 [40 points]

- Banyaknya total perintah dan queries adalah antara 1 dan 1 000 000 (inklusif).

Detail implementasi

Anda harus mengumpulkan sebuah file, dinamakan `scrivener.c`, `scrivener.cpp` atau `scrivener.pas`. File yang dikumpulkan harus mengimplementasi subprogram-subprogram yang dijelaskan di atas dengan signatures sebagai berikut.

C/C++ programs

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

Pascal programs

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

Subprogram harus berperilaku seperti yang dijelaskan di atas. Selain itu, anda bebas untuk mengimplementasikan subprogram lain untuk kepentingan internal. Program yang Anda kumpulkan dilarang berinteraksi dengan input/output standar, atau dengan file lain.

Contoh grader

Contoh grader membaca input dengan format sebagai berikut

- baris 1: banyaknya total perintah dan queries pada input;
- pada setiap baris sebagai berikut:
 - T diikuti oleh sebuah spasi dan sebuah huruf kecil untuk sebuah `TypeLetter` command;
 - U diikuti oleh sebuah spasi dan sebuah bilangan bulat untuk `UndoCommands`;
 - P diikuti oleh sebuah spasi dan sebuah bilangan bulat untuk `GetLetter`.

Contoh grader akan mencetak karakter yng dikembalikan dengan `GetLetter`, setiap karakter pada baris terpisah.