

## Skryba dwukierunkowy

Niektórzy twierdzą, że Leonardo da Vinci był wielkim miłośnikiem Johanna Gutenberga — niemieckiego kowala, który wynalazł prasę drukarską. Zainspirowany wynalazkiem Gutenberga, Leonardo stworzył proste urządzenie drukarskie: skrybę dwukierunkową. Przypomina ono nieco prostą maszynę do pisania, która obsługuje dwa polecenia: pierwsze polecenie służy do napisania jednej litery, zaś drugie to polecenie undo, służące do cofania ostatnio wykonanych poleceń. Co ważne, polecenie undo działa wyjątkowo sprytnie, bo traktuje wcześniejsze undo również jako polecenia, a co za tym idzie, może cofać ich wykonanie.

### Zadanie

Napisz program, który będzie symulować działanie dwukierunkowego skryby. Skryba wykonuje kolejne operacje wprowadzane przez użytkownika i odpowiada na pytania o litery znajdujące się na podanych pozycjach w tekście. Zakładamy, że początkowy tekst jest pusty.

- `Init()` — Funkcja ta zostanie wywołana dokładnie raz, na początku wykonania programu, bez żadnych argumentów. Może być ona użyta do zainicjalizowania struktur danych w Twoim programie. Nigdy nie będzie trzeba cofać jej wykonania.
- `TypeLetter(L)` — dopisz na końcu tekstu jedną literę `L`, gdzie `L` to mała litera alfabetu angielskiego.
- `UndoCommands(U)` — wycofaj `U` ostatnich poleceń, gdzie `U` to dodatnia liczba całkowita.
- `GetLetter(P)` — zwróć literę na pozycji `P` w aktualnym tekście, gdzie `P` jest nieujemną liczbą całkowitą. Pozycje w tekście numerujemy od 0. Zapytanie nie jest poleceniem i nie ma znaczenia dla wykonywania polecenia undo.

Po początkowym wywołaniu `Init()`, pozostałe funkcje mogą być wywołane zero lub więcej razy, w dowolnej kolejności. Możesz założyć, że `U` będzie nie większe niż liczba wcześniej otrzymanych poleceń oraz że `P` będzie mniejsze niż aktualna liczba liter w tekście.

Polecenie `UndoCommands(U)` cofa ostatnie `U` poleceń, począwszy od tego, które zostało wykonane jako ostatnie. Wycofanie polecenia `TypeLetter(L)` polega na usunięciu litery `L` z końca aktualnego tekstu. Z kolei wycofanie polecenia `UndoCommands(X)` polega na ponownym wykonaniu ostatnich `X` poleceń w ich pierwotnym porządku.

### Przykład

Oto przykładowy ciąg wywołań funkcji oraz tekst po każdym wywołaniu.

Wywołanie	Zwrócona wartość	Aktualny 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

### Podzadanie 1 [5 punktów]

Polecenia i zapytania zostaną łącznie wywołane od 1 do 100 razy. Polecenie UndoCommands nie będzie wykonywane.

### Podzadanie 2 [7 punktów]

Polecenia i zapytania zostaną łącznie wywołane od 1 do 100 razy. Polecenie UndoCommands nigdy nie będzie wycofywane.

### Podzadanie 3 [22 punkty]

Polecenia i zapytania zostaną łącznie wywołane od 1 do 5 000 razy.

### Podzadanie 4 [26 punktów]

Polecenia i zapytania zostaną łącznie wywołane od 1 do 1 000 000 razy. Wszystkie wywołania GetLetter będą miały miejsce po wszystkich wywołaniach TypeLetter i UndoCommands.

### Podzadanie 5 [40 punktów]

Polecenia i zapytania zostaną łącznie wywołane od 1 do 1 000 000 razy.

### Szczegóły implementacji

Należy zgłosić dokładnie jeden plik o nazwie scrivener.c, scrivener.cpp lub scrivener.pas. Powinien on zawierać implementacje funkcji opisanych powyżej.

## Programy w C/C++

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

## Programy w Pascalu

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

Funkcje powinny działać dokładnie tak, jak opisano powyżej. Twój program nie powinien korzystać ze standardowego wejścia, standardowego wyjścia lub jakichkolwiek plików.

## Przykładowy moduł oceniający

Przykładowy moduł oceniający wczytuje dane w następującym formacie:

wiersz 1: łączna liczba poleceń i zapytań opisanych w dalszej części wejścia;  
w każdym z kolejnych wierszy:

- T x — operacja TypeLetter(x);
- U y — operacja UndoCommand(y);
- P z — operacja GetLetter(z).

Przykładowy moduł oceniający wypisuje litery zwracane przez GetLetter, każdą w osobnym wierszu.