

## 크라이피쉬 글쓰는 기계

활자 인쇄기를 발명한 독일 대장장이인 요하네스 구텐베르크는 레오나르도가 열렬히 존경하는 사람이었다. 레오나르도는 크라이피쉬 글쓰는 기계 — *il gambero scrivano* — 로 불리는 매우 단순한 타자 기계를 설계하여 구텐베르크에게 존경을 표시하였다. 이것은 단순한 현대의 타자기와 매우 비슷하며, 단지 두 개의 명령어만 받아들인다: 하나는 그 다음 문자를 타자하는 것이고, 다른 하나는 가장 최근의 명령어들을 취소(undo)하는 것이다. 이 타자 기계의 주목할만한 특징은 취소 기능이며, 이는 매우 강력한 기능이다: 취소 명령은 자기 자신, 즉 취소 명령어에 대해서도 적용될 수 있다.

### 해야 할 일

여러분이 할 일은 크라이피쉬 글쓰기 기계와 같이 동작하는 소프트웨어를 구현하는 것이다: 빈 텍스트에서 시작하고, 사용자가 입력한 명령어의 순서들을 받아들이고, 다음과 같이 텍스트의 현재 상황의 특정 위치에 대한 질의를 수행한다.

- `Init()` — 실행 초기에 인자가 없이 한번만 호출된다. 이 함수는 자료 구조를 초기화 하는데 사용될 수 있다. 이것은 결코 취소될 필요가 없을 것이다.
- `TypeLetter(L)` — `a, ..., z` 으로부터 선택된 하나의 소문자 `L` 을 텍스트의 끝에 추가한다.
- `UndoCommands(U)` — 양의 정수 `U` 에 대해, 가장 최근의 `U` 개의 명령어를 취소한다.
- `GetLetter(P)` — 음이 아닌 인덱스 `P` 에 대해, 현재 텍스트에서 위치 `P` 의 문자를 되돌려준다. 텍스트에서 첫 번째 문자의 인덱스는 `0` 이다. (이 질의는 명령어가 아니라서 취소 명령어에서는 무시된다.)

`Init()` 를 호출한 후에, 그 외의 루틴들은 임의의 순서로 `0` 번 이상 호출 될 수 있다. `U` 는 이전에 들어온 명령어들의 수를 초과하지 않음이 보장되며, `P` 는 현재의 텍스트의 길이보다 작을 것이다 (현재 텍스트에서의 문자의 수).

`UndoCommands(U)` 에서는 이전의 `U` 개의 명령어를 역순으로 취소한다: 만약 취소될 명령어가 `TypeLetter(L)` 라면, 이것은 현재 텍스트의 끝에서부터 `L` 을 삭제한다: 만약 취소될 명령어가 어떤 값 `X` 에 대해 `UndoCommands(X)` 라면 이것은 이전의 `X` 개의 명령어를 원래의 순서대로 다시 실행한다.

### 예제

하나의 가능한 호출순서에 대해서 각 호출후의 텍스트의 상태를 보인다.

Call	Returns	Current text
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

## 서브태스크 1 [5점]

명령어와 질의의 전체 수가 1이상 100이하이고, UndoCommands의 호출은 없다.

## 서브태스크 2 [7점]

명령어와 질의의 전체 수가 1이상 100이하이고, UndoCommands가 취소되는 경우는 없다.

## 서브태스크 3 [22점]

명령어와 질의의 전체 수가 1이상 5,000이하이다.

## 서브태스크 4 [26점]

명령어와 질의의 전체 수가 1이상 1,000,000이하이고, GetLetter에 대한 모든 호출은 TypeLetter 와 UndoCommands의 모든 호출뒤에 발생한다.

## 서브태스크 5 [40점]

명령어와 질의의 전체 수가 1이상 1,000,000이하이다.

## 구현 세부 사항

scrivener.c, scrivener.cpp 혹은 scrivener.pas 중 단 하나의 파일만을 제출해야 한다. 제출하는 파일에는 다음 함수들의 선언에 맞는 프로그램이 구현되어있어야 한다.

## C/C++ 프로그램

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

## Pascal 프로그램

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

이 부프로그램들은 문제에서 기술한대로 동작하여야 한다. 물론, 다른 부프로그램의 구현 내용은 자유롭게 하면 된다. 제출한 프로그램은 어떠한 방식으로든 표준 입/출력뿐만 아니라, 다른 어떠한 파일과도 상호작용을 하여서는 안된다.

## grader 예시

주어지는 grader에서는 다음의 형식에 맞는 입력을 받아들인다.

- line 1: 입력에서 명령어와 질의의 전체 수;
- 그 다음 각 라인에 대해:
  - TypeLetter 명령어에 대해서는 T와 하나의 소문자가 빈칸을 사이에 두고 주어진다.
  - UndoCommands 명령어에 대해서는 U와 하나의 정수가 빈칸을 사이에 두고 주어진다.
  - GetLetter 명령어에 대해서는 P와 하나의 정수가 빈칸을 사이에 두고 주어진다.

주어지는 grader에서는 GetLetter에 의해 리턴된 문자를 한 줄에 하나씩 출력한다.