

스택



시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
0.5 초 (추가 시간 없음)	256 MB	252650	91618	66575	37.448%

문제

정수를 저장하는 스택을 구현한 다음, 입력으로 주어지는 명령을 처리하는 프로그램을 작성하시오.

명령은 총 다섯 가지이다.

- push X: 정수 X를 스택에 넣는 연산이다.
- pop: 스택에서 가장 위에 있는 정수를 빼고, 그 수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- size: 스택에 들어있는 정수의 개수를 출력한다.
- empty: 스택이 비어있으면 1, 아니면 0을 출력한다.
- top: 스택의 가장 위에 있는 정수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.

입력

첫째 줄에 주어지는 명령의 수 N ($1 \leq N \leq 10,000$)이 주어진다. 둘째 줄부터 N개의 줄에는 명령이 하나씩 주어진다. 주어지는 정수는 1보다 크거나 같고, 100,000보다 작거나 같다. 문제에 나와있지 않은 명령이 주어지는 경우는 없다.

출력

출력해야하는 명령이 주어질 때마다, 한 줄에 하나씩 출력한다.

예제 입력 1 복사

```
14
push 1
push 2
top
size
empty
pop
pop
pop
size
empty
pop
push 3
empty
top
```

예제 출력 1 복사

```
2
2
0
2
1
-1
0
1
-1
0
3
```

예제 입력 2 복사

```
7
pop
top
push 123
top
pop
top
pop
```

예제 출력 2 복사

```
-1
-1
123
123
-1
-1
```

1. 문제인식 : stack구조구현

입력 : N - 명령어의 수

- push X : push라 X 사이에 공백 존재, 출력 X → getline 문자열 입력 받고, substr 이용하여 문자열 분리, X는 정수로 변환
- pop : 마지막에 들어간 수 빼고, 출력 / 스택이 비어있을 경우 -1 출력
- size : 스택에 들어있는 정수의 개수 출력
- empty : 비어있으면 1, 아니면 0

2. 문제 유형 파악 : N을 입력받고 N만큼 반복문을 돌리며 명령을 입력받은 명령에 해당하는 출력을 하거나 스택을 변경

3. 다이어그램 / 알고리즘

. 알고리즘

① N 입력 받기

→ 공백 포함하여 '한 줄' 받기 위해 getline() 사용

② N만큼 반복하며 명령어 입력 받고 해당 명령어 수행

③ 각각의 케이스 판단.

→ 명령어 내에 공백 판단

있는 경우 : push X, 문자열과 정수 분리	} substr() 사용 find() 사용
없는 경우 : pop, size, empty, top	

④ 각각의 명령어 구현

i) push X

[스택의 size가 0일 때 : 빈 스택을 할당, X 정수 넣기
" 이 아닐 때 : 새로운 스택을 생성, X 값 넣어주고, 기존의 스택과 연결

ii) pop

[스택의 size가 0일 때 : -1 출력
" 이 아닐 때 : size() - 1 번째의 원소 제거 후, 원소 값 반환
새로운 스택 할당 후 기존의 스택 옮기기

iii) size

- 스택의 크기 반환

iv) empty

[스택의 size가 0일 때 : 1 출력
" 이 아닐 때 : 0 출력

v) top

[스택의 size가 0일 때 : -1 출력
" 이 아닐 때 : size() - 1 번째의 원소 값 출력

⑤ 프로그램 종료