

## <자료구조및실습 과제> - 큐로 스택 만들기

**문제:** 주교재의 심층문제 7-4(두 개의 큐로 스택 만들기)를 확장한 문제

### 프로그램 요구사항

- 1) 먼저, 교재의 심층문제 7-4에서 요구하는대로 스택 **S**의 메소드들을 **C** 함수로 구현하라.
- 2) 여기에 추가하여 주함수를 작성하라. 주함수는 사용자가 표준입력을 통해 전달한 아래 명령들에 대해 **S**의 메소드를 호출하여 실행하고 결과를 인쇄한다.  
**S** : **S**에 대해 **isEmpty()** 호출 (**isEmpty**)  
**t** : **S**에 대해 **top()** 호출 (**top**)  
**p n1 n2 n3 ...** : **S**에 대해 **push(n1), push(n2), push(n3) ...** 를 차례로 호출 (**push**)  
**P** : **S**에 대해 **push(n)**을 100만번 호출 - 여기서 **n**은 10~99 사이의 정수 난수 (**pushMillion**)  
**o** : **S**에 대해 **pop()** 호출 (**pop**)  
**q** : 수행 종료 (**quit**)
- 3) 프로그램 작성이 완료되면 아래 주어진 순서대로 명령을 실행하여 정확성을 테스트하라.

### 입출력 예시

입력	출력
<b>S</b>	Empty (0), cputime = XX.XXXXXXXXXX
<b>o</b>	Empty Stack Exception!! (0), cputime = XX.XXXXXXXXXX
<b>p 5</b>	OK (1), cputime = XX.XXXXXXXXXX
<b>S</b>	Nonempty (1), cputime = XX.XXXXXXXXXX
<b>o</b>	5 (0), cputime = XX.XXXXXXXXXX
<b>p 8 1 8 3</b>	OK (4), cputime = XX.XXXXXXXXXX
<b>t</b>	3 (4), cputime = XX.XXXXXXXXXX
<b>o</b>	3 (3), cputime = XX.XXXXXXXXXX
<b>o</b>	8 (2), cputime = XX.XXXXXXXXXX
<b>P</b>	OK (1000002), cputime = XX.XXXXXXXXXX
<b>o</b>	?? (1000001), cputime = XX.XXXXXXXXXX
<b>p 8</b>	OK (1000002), cputime = XX.XXXXXXXXXX
<b>o</b>	8 (1000001), cputime = XX.XXXXXXXXXX
<b>o</b>	?? (1000000), cputime = XX.XXXXXXXXXX
<b>p 1</b>	OK (1000001), cputime = XX.XXXXXXXXXX
<b>o</b>	1 (1000000), cputime = XX.XXXXXXXXXX
<b>q</b>	

### ※ 참고:

1. 위 실행예에 보인 **출력형식을 준수**해야 한다.

2. 실행예에서 괄호 속의 수는 해당 명령 수행직후의 스택 **S** 사이즈를 나타낸다.
3. `cputime X.XXXXXXXX`는 측정된 `cputime` in milliseconds이다. 이 실행시간은 각 부함수에 대한 호출과 반환 사이에 소요된 `cputime`을 표시해야 한다. 이 과제의 경우, 상수 시간을 나타내는 `cputime`은 거의 0초 혹은 아예 0초로 표시되더라도 무방하다.
4. 위 실행예에서 ??는 임의의 난수 정수를 나타낸다. 즉, 물음표 두 개가 아닌, **S**가 실제 반환한 10~99 사이의 두 자릿수 정수 난수를 말한다.
5. 채점에 방해가 되므로, 위에 보인 것과 같이, 수많은 스택원소들을 출력하는 일은 절대 없어야 한다.
6. 위 실행예 수행에서, 자신의 PC 성능 문제(너무 느림)로 인해 불가피하다면 100만 대신 10만을 사용해서 작성해도 좋다.
7. 위 실행예는 예시일 뿐, 다른 실행예를 스스로 만들어 추가로 수행해보는 것도 좋다.

### 주의

- 1) **S**를 만들기 위한 부품으로 사용되는 큐들은 배열 또는 연결리스트 가운데 어느 것을 사용하여 구현하더라도 무방하며, 만약 배열로 할 경우 각 큐의 할당 크기는 위 실행예를 구현하기에 충분한 크기로 설정하여 크기 부족으로 인한 예외상황이 발생하지 않도록 하라.
- 2) **S**의 각 메소드가 응용문제 7.5.2 해결에서 설명한 최선의 접근시간 성능에 미치지 않으면 감점.
- 3) 원래 응용문제에서 부품 큐들은 **size** 메소드를 지원하지 않는다고 했으므로, 이 메소드(또는 이름만 다를 뿐 큐의 원소 수를 반환하는 유사 메소드)를 구현하여 사용하는 것은 허용되지 않는다. 참고로 스택 **S**가 **size** 메소드를 지원하는 것은 금지하지 않는다.
- 4) 상수시간 작업의 `cputime`이 0초 또는 거의 0초로 나타나는 것은 허용함.
- 5) 모듈화 설계할 것. 특히, 명령 에코, 괄호 속의 현재 스택 원소 수, `cputime` 등에 대한 계산 및 출력은 모두 (각 부함수가 아닌) 주함수에서 수행해야 함. 이러한 계산 또는 출력 명령이 부함수에 포함되면 모듈화 불량으로 감점.
- 6) 주어진 큐 두 개만 이용해서 **S**를 구현해야 한다. 다시 말해, **S** 내에 주어진 두 개의 큐 이외의 다른 데이터구조를 포함할 수 없으며, 각각의 큐 내에도 큐 이외의 다른 데이터구조를 당연히 포함할 수 없다.
- 7) 속임수에 의한 프로그램을 작성한 경우 부정행위로 처리함.
- 8) 출력 화면을 캡처하여 제출하는 과제가 아님. 주어진 실행예 및 스스로 만든 추가 실행예를 각자 수행해보고 잘 수행하는지 확인한 후 소스프로그램만을 OJ에 제출할 것.