

## 〈자료구조 실습〉 - 스택 (1)

### ※ 입출력에 대한 안내

- 특별한 언급이 없으면 문제의 조건에 맞지 않는 입력은 입력되지 않는다고 가정하라.
- 특별한 언급이 없으면, 각 줄의 맨 앞과 맨 뒤에는 공백을 출력하지 않는다.
- 출력 예시에서 □는 각 줄의 맨 앞과 맨 뒤에 출력되는 공백을 의미한다.
- 입출력 예시에서  $\mapsto$  이 후는 각 입력과 출력에 대한 설명이다.

[ 문제 1 ] 스택 ADT를 배열로 구현하고 테스트하는 프로그램을 작성하시오.

- 원소 : 영문자
- 다음 연산을 지원해야 함.
  - **push**(stack, 'c') : stack의 top에 원소를 추가한다. stack이 이미 꽉 차있으면 해당 데이터는 스택에 저장하지 않고 "Stack FULL"을 출력한다.
  - **pop**(stack) : stack의 top에 있는 원소를 반환하고 stack에서 제거한다. stack이 비어 있으면 "Stack Empty"를 출력한다.
  - **peek**(stack) : stack의 top에 있는 원소를 화면에 출력한다. stack은 변화하지 않는다. stack이 비어 있으면 "Stack Empty"를 출력한다.
  - **duplicate**(stack) : stack의 top에 있는 원소를 pop해서 두 번 push한다. stack이 이미 꽉 차 있으면 "Stack FULL"을 출력한다.
  - **upRotate**(stack, n) : stack의 맨 위 n개의 원소를 회전시킨다. 예를 들어 n이 3이고 stack의 top에서부터 elem1, elem2, elem3, .... 이 저장되어 있으면 원소를 하나씩 위쪽으로 이동시킨다. 맨 위쪽(top)의 elem1은 n-1번 아래쪽으로 이동해서 스택의 내용은 elem2, elem3, elem1, ... 이 된다.
  - **downRotate**(stack, n) : stack의 맨 위 n개의 원소를 회전시킨다. 예를 들어 n이 3이고 stack의 top에서부터 elem1, elem2, elem3, .... 이 저장되어 있으면 원소를 하나씩 아래쪽으로 이동시킨다. top에서부터 n번째의 원소를 top으로 이동해서 스택의 내용은 elem3, elem1, elem2, ... 이 된다.
  - **print**(stack) : stack의 모든 원소를 top에서부터 순서대로 공간없이 출력한다.
- 입력에 대한 설명 (아래 입출력 예시 참조)
  - 한 라인에 하나의 연산이 입력되고, 라인에는 연산의 종류와 그에 필요한 인자가 주어진다.
  - 연산의 종류 : 각 연산 이름은 POP, PUSH, PEEK, DUP, UpR, DownR, PRINT로 주어진다.

입력 예시 1

4	↳ stack의 크기 N
10	↳ 연산의 개수
POP	↳ pop(stack)
PUSH s	↳ push(stack, 's')
PUSH t	↳ push(stack, 't')
PUSH a	↳ push(stack, 'a')
PUSH r	↳ push(stack, 'r')
PRINT	↳ print(stack)
UpR 3	↳ upRotate(stack, 3)
PRINT	↳ print(stack)
PUSH s	↳ push(stack, 's')
PEEK	↳ push(stack)

출력 예시 1

Stack Empty	↳ 1번 POP 연산의 결과
rats	↳ 6번 PRINT 연산의 결과
atrs	↳ 8번 PRINT 연산의 결과
Stack FULL	↳ 9번 PUSH 연산의 결과
a	↳ 10번 PEEK 연산의 결과

입력 예시 2

5	↳ stack의 크기 N
11	↳ 연산의 개수
PUSH s	↳ push(stack, 's')
PUSH r	↳ push(stack, 'r')
PUSH a	↳ push(stack, 'a')
PUSH t	↳ push(stack, 't')
PUSH s	↳ push(stack, 's')
PRINT	↳ print(stack)
DownR 4	↳ downRotate(stack, 4)
PRINT	↳ print(stack)
POP	↳ pop(stack)
POP	↳ pop(stack)
PRINT	↳ print(stack)

출력 예시 2

stars	↳ 6번 PRINT 연산의 결과
rstas	↳ 8번 PRINT 연산의 결과
tas	↳ 11번 PRINT 연산의 결과

입력 예시 3

3	↳ stack의 크기 N
5	↳ 연산의 개수
PUSH d	↳ push(stack, 'd')
DUP	↳ duplicate(stack)
PUSH a	↳ push(stack, 'a')
PRINT	↳ print(stack)
PUSH s	↳ push(stack, 's')

출력 예시 3

add	↳ 4번 PRINT 연산의 결과
Stack FULL	↳ 5번 PRINT 연산의 결과

[ 문제 2 ] 스택 응용으로, 키보드로부터 입력된 한 라인의 수식 문장 내 괄호 쌍의 대칭성을 검사하는 프로그램을 작성하세요. 괄호 쌍은 { }, [ ] , ( ), 세 종류가 있다.

※ 주의: 한 개의 수식 문장은 1000개를 넘지 않는 문자로 이루어진다. 수식 문장은 공백문자를 포함할 수 있다. 검사 결과 대칭이 아니면 'Wrong\_N'을, 대칭이면 'OK\_N'를 출력한다. 여기서 N은 문장 내 괄호의 개수다.

입력 예시 1

출력 예시 1

(3+40\*(2+(30-7)\*2133)

Wrong\_5

입력 예시 2

출력 예시 2

3\*{4+(2-792)/1} + [3\*{4-2\* (100 -7)}]

OK\_10

입력 예시 3

출력 예시 3

301\*{4+(2101-7)/1} + 9\*{4-2\* (10108-7)}]

Wrong\_9

입력 예시 4

출력 예시 4

(3\*{4001+(2-7)/1} + [3\*{4-2\* (1-7)}])

OK\_12