스택(Stack)

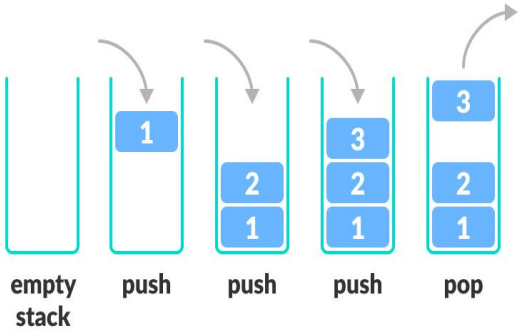
1. 스택(Stack)의 개념

한 쪽 끝에서만 자료를 넣고 뺄 수 있는 LIFO(Last In First Out) 형식의 자료 구조이다.
즉, 가장 최근에 스택에 추가한 항목이 가장 먼저 제거되는 자료구조이다.

2. 스택(Stack)의 연산(자바 기준)

1) push(item): item 하나를 스택의 가장 윗 부분에 추가한다.
2) pop(): 스택에서 가장 위에 있는 항목을 제거한다.
3) peek(): 스택의 가장 위에 있는 항목을 반환한다.
4) empty(): 스택이 비어 있을 때에 true를 반환한다.

3. 파이썬에서 스택사용하기




```
def solution():
    stack = []
    stack.append(1)
    stack.append(2)
    stack.append(3)
    stack.pop()
    print(stack[-1]) #스택의 가장 위에 있는 항목 확인
    print(len(stack) == 0) #스택이 비어있는지 확인
    stack.pop()
    stack.pop()
    print(len(stack) == 0)
    return stack

print(solution())
```

큐(Queue)

1. 큐(Queue)의 개념

한 쪽 끝에서 자료가 삽입되고, 반대쪽 끝에서 자료가 삭제되는 FIFO(First In First Out) 형식의 자료 구조이다.
즉, 먼저 추가한 항목이 가장 먼저 제거되는 자료구조이다.

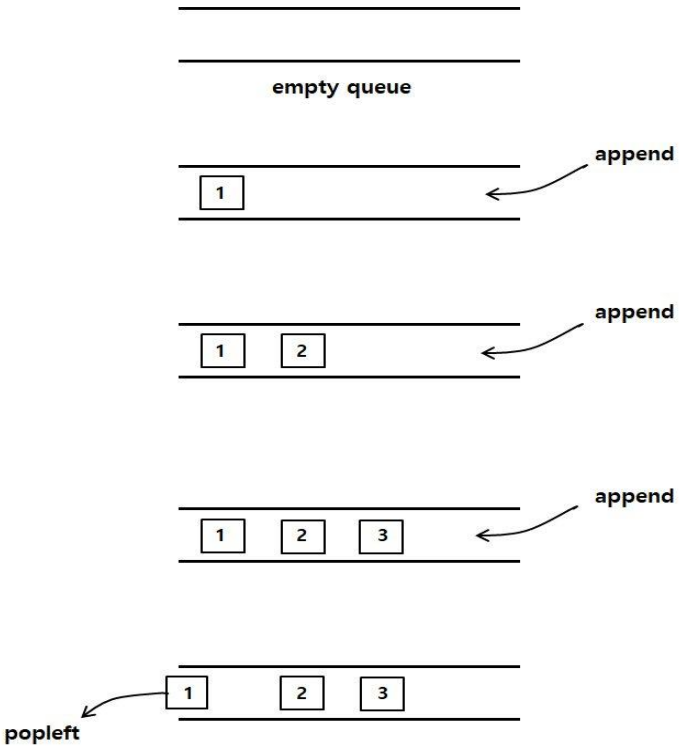
2. 큐(Queue)의 연산

1) append(item): item 하나를 큐의 뒷 부분에 추가한다.
2) popleft(): 큐에 가장 앞에 있는 항목을 제거하고 반환한다.

3. 파이썬에서 큐사용하기

```
from collections import deque
def solution():
    queue = deque()
    queue.append(1)
    queue.append(2)
    queue.append(3)
    print(queue.popleft())
    print(len(queue) == 0)
    queue.popleft()
    queue.popleft()
    print(len(queue) == 0)
    return queue

print(solution())
```



올바른 괄호

괄호 문자열이 입력되면 올바른 괄호이면 "YES", 올바르지 않으면 "NO"를 출력합니다.
(())() 이것은 괄호의 쌍이 올바르게 위치하는 거지만, (()()))은 올바른 괄호가 아니다.

입출력 예:

s	answer
"(((()))()"	"YES"
"(()(()"	"NO"
"()())"	"NO"
"())("	"NO"
"(((()))()("	"NO"

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.

Backspace

현수는 주어진 문자열의 문자 순서대로 키보드 자판의 문자를 쳐 화면에 s문자열을 작성합니다. 문자열에는 '#'문자가 있는데 이 문자는 Backspace키를 의미합니다.
매개변수 s에 현수가 키보드 자판을 쳐야할 순서인 문자열이 주어지면 현수가 s문자열을 작성했을 때 최종적으로 화면에 작성된 문자열을 반환하는 프로그램을 작성하세요.
화면에는 적어도 문자 한 개는 작성되어 있습니다.

입출력 예:

s	answer
"abc##ec#ab"	"aeab"
"kefd#ef##s##"	"ke"
"teac#cher##er"	"teacher"
"englitk##shabcde##ff##ef##ashe####"	"englishabc"
"itistime####gold"	"itisgold"

제한사항:
• 문자열 s의 길이는 1,000을 넘지 않습니다.

연속된 문자 지우기

매개변수 s에 문자열이 주어지면 이웃한 두 개의 문자가 같으면 두 문자를 제거합니다. 이 과정을 반복해서 최종적으로 남는 문자만으로 이루어진 문자열을 반환하는 프로그램을 작성하세요.
만약 "acbbcaa"라는 문자열이 주어진다면 최초 bb가 연속되어 있어 제거하고 나면 "accaa"가 되고, 다시 cc가 연속되어 제거하면 "aaa"가 되고 "aa"연속되어 제거하면 "a"가 최종적으로 남습니다.

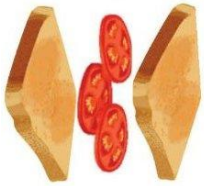
입출력 예:

s	answer
"acbbcaa"	"a"
"bacccaba"	"bacaba"
"aabaababbaa"	"a"
"bcaacccbaabccabbaa"	"ba"
"cacaabbc"	"ca"

제한사항:
- 문자열 s의 길이는 100,000을 넘지 않습니다.
- 문자열 s는 소문자로만 이루어져 있습니다.

샌드위치

현수는 샌드위치를 만들려고 합니다.
식탁에 샌드위치 재료인 식빵과 토마토가 일렬로 놓여 있습니다.
현수가 만드는 샌드위치는 식빵-토마토-식빵 순으로 합쳐서 포장을 합니다. 현수는 식탁에 놓여 있는 재료의 순서를 유지하면서 샌드위치를 만들어야 합니다.



만약 식탁에 [식빵, 식빵, 식빵, 토마토, 식빵, 식빵, 토마토, 식빵, 토마토, 식빵] 순으로 놓여 있다면 현수는 3번째, 4번째, 5번째를 합쳐서 샌드위치를 만들고, 6번째, 7번째, 8번째를 합쳐서 샌드위치를 만들고, 2번째, 9번째, 10번째를 합쳐서 샌드위치를 만들어 총 3개의 샌드위치를 만들 수 있습니다. 매개변수 nums에 식탁에 놓여 있는 샌드위치 재료의 정보가 주어지면 현수가 만들 수 있는 샌드위치의 총 개수를 반환하는 프로그래을 작성하세요.

입출력 예:

nums	answer
[1, 1, 1, 2, 1, 1, 2, 1, 2, 1]	3
[2, 1, 1, 2, 1, 1, 2, 1, 1, 1, 2, 1, 2, 1, 1, 2, 1, 2, 1, 2, 1]	6
[1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1]	2
[2, 1, 1, 1, 2, 1, 2]	1
[1, 1, 1, 1, 1, 1, 1]	0

제한사항:
- nums의 길이는 1,000을 넘지 않습니다.
- nums[i]값은 1 또는 2입니다. 1은 식빵을 의미하고, 2는 토마토를 의미합니다.

고장난 프린터

현수가 다니는 회사의 프린터가 고장이 나서 프린트를 요청한 순서대로 프린트를 하는게 아니라 약간 이상은 규칙에 의해서 프린트를 합니다.
이상한 규칙은 다음과 같습니다.
요청한 순서에서 먼저 2개의 작업을 프린트하고, 3번째 작업은 순서상 맨 뒤로 보내버립니다.
이 규칙을 반복하면서 프린트를 합니다.
만약 프린트를 요청한 작업번호 순서가 아래와 같다면
[3, 1, 4, 5, 2, 6, 7]
3번, 1번 작업을 프린트 하고, 4번 작업은 맨뒤로 보냅니다.
[5, 2, 6, 7, 4]
5번, 2번 작업을 프린트 하고, 6번 작업을 맨뒤로 보냅니다.
[7, 4, 6]
7번, 4번 작업을 프린트 하고 6번 작업을 맨뒤로 보냅니다.
[6]
마지막으로 6번 작업을 프린트합니다.
매개변수 nums에 프린트를 요청한 작업번호 순서가 주어지면 맨 마지막에 출력되는 작업의 번호를 반환하는 프로그램을 작성하세요.

입출력 예:

nums	answer
[3, 1, 4, 5, 2, 6, 7]	6
[10, 8, 3, 1, 4, 5, 2, 6, 7, 9]	5
[10, 8, 3, 11, 12, 1, 4, 5, 2, 6, 7, 9]	2
[10, 8, 3, 1, 4, 5, 2, 11, 13, 6, 7, 12, 9, 14]	12
[1, 8, 6, 10, 4, 7, 2, 5, 3, 9]	7

제한사항:
• nums의 길이는 1,000을 넘지 않습니다.
• 1 <= nums[i] <= 1,000