

Computing Foundation for Data Science

Project 2

Due: 10/19/2021 10:59AM

주의사항

- 코드를 Jupyter Notebook에서 작성하였더라도 **python**파일 (.py)로 제출할 것.
- 함수가 의도한 값을 return 하는지를 확인할 것. (print 와 혼동하지 말 것)
- 파일명은 **P1~P4.py**를 유지하고, 해당 해당 파일들을 PROJ2_학번_이름.zip 으로 압축하여 제출할 것. 예를 들면 학번이 2020-12345 이고, 이름이 Keondo Park 이라면 **PROJ2_2020_12345_KeondoPark.zip** 으로 압축하여 제출.
- 예시로 제시한 입력 값 외에도 조교가 임의로 생성한 입력 값으로도 코드가 잘 실행되는지 테스트할 예정.
- 뼈대 코드의 함수 이름 및 매개변수(parameter)는 변경하지 말 것.
- 채점은 프로그램에 의해 기계적으로 처리되므로 위 사항을 지키지 않은 경우 누락되거나 불이익을 받을 수 있음.
- 늦은 제출은 받지 않음.
- 표절검사를 수행하여 발각될 경우 성적 F 부여함.

Problem 1: 괄호 열고 닫기

'(', ')', '{', '}', '[', ']'으로 구성된 **String** 을 **Input** 으로 받고, 이 **String** 이 아래 규칙에 의해 유효한지를 판단하는 함수를 만들어라.

- 괄호가 열렸다면 같은 타입의 괄호에 의해 닫혀야 한다.
- 적절한 순서로 괄호가 배열되어야 한다. 안쪽에서 열린 괄호가 먼저 닫히고, 바깥쪽 괄호가 닫혀야 한다.

예시

1. 유효한 입력 예제:

```
>>> P1('()')
```

```
True
```

2. 유효한 입력 예제:

```
>>> P1('()[]{}')
```

```
True
```

3. 안쪽에서 열린 '['가 닫히기 전에 ')'가 먼저 닫혔으므로 유효하지 않다.

```
>>> P1('([)]')
```

```
False
```

Problem 2: 주식 매매

예지는 HS컴퓨팅이라는 주식에 단기투자를 통하여 수익을 내려고 한다. 아래의 표는 HS컴퓨팅의 날짜 별 주가 이다. 예지는 수익을 내기 위하여 걸리는 최소 일수(日數)를 알고 싶어 한다.

날짜	10/7	10/8	10/9	10/10	10/11	10/12	10/13	10/14
주가	500	600	700	300	100	400	800	500
일수	1	1	4	2	1	1	0	0

위의 표를 보면,

- 예지가 10/7일에 주식을 산다면 1일 뒤인인 10/8일에 팔면 이득을 본다.
- 예지가 10/9일에 주식을 산다면 4일 뒤인 10/13일에 팔면 이득을 본다.

해당 예시를 참고하여 input으로 일별 주가(list)를 받고 output으로 이득을 보기 위해 기다려야 하는 최소 일수(list)를 반환하는 함수 P2를 작성하여라.

예시

```
>>> P2([500, 600, 700, 300, 100, 400, 800, 500])  
[1, 1, 4, 2, 1, 1, 0, 0]
```

Stack을 활용하여 구현해볼 것!

Problem 3: Linked List 순서 뒤집기

Singly linked list 를 입력받아 순서를 뒤집은 Singly linked list 로 만드는 함수를 작성하여라.
Linked list 의 각 Node 는 linked_list_helper.py 파일의 ListNode 정의를 활용하고,
함수의 return 값은 뒤집은 Linked list 의 첫번째 Node 로 한다.

P3은 LinkedList의 head인 ListNode를 input으로 받고,
뒤집은 Linked List의 head인 ListNode를 리턴 하면 된다.

예시

1.

```
>>> l1 = create_linked_list([4,2,1,3])  
>>> print_linked_list(P3(l1),[])  
[3,1,2,4]
```
2.

```
>>> l2 = create_linked_list([-1,5,0,2,3])  
>>> print_linked_list(P3(l2),[])  
[3,2,0,5,-1]
```
3.

```
>>> l3 = create_linked_list([])  
>>> print_linked_list(P3(l3),[])  
[]
```

Problem 4: 숫자 덧셈

십진수 숫자를 나타내는 LinkedList 두 개가 주어질 때, 두 숫자의 합을 다시 LinkedList로 반환하는 함수 P4를 작성하여라.

예를 들어, (5)->(3)->(4)->(8)이라는 LinkedList와 (2)->(1)->(3)이라는 LinkedList가 input으로 주어지면, 이 둘의 합인 (5)->(5)->(6)->(1)이라는 LinkedList를 리턴하면 된다.

주의사항: LinkedList를 list로 변환 후 덧셈을 하면 0점 처리

(참고) 아래 그림에서 오른쪽과 같이 잘못 계산하지 않도록 주의할 것.

	5	3	4	8		5	3	4	8
+		2	1	3		+	2	1	3
<hr/>					<hr/>				
	5	5	6	1		7	4	7	8

참고: **Linked list**를 위한 헬퍼 함수 소개

1. ListNode 클래스와 class variables

```
1 # Definition for singly-linked list.
2 class ListNode:
3     def __init__(self, val=0, next=None):
4         self.val = val
5         self.next = next
```

2. list로 LinkedList만들기 위한 함수.

```
1 #Create a linked list from a list, then return head node
2 def create_linked_list(l: list) -> ListNode:
3     if len(l) == 0:
4         return None
5
6     start = ListNode(l[0])
7     node = start
8
9     for i in range(1, len(l)):
10         node_new = ListNode(l[i])
11         node.next = node_new
12         node = node_new
13
14     return start
```

3. LinkedList를 list로 변환 후 print하는 함수

```
1 #Print all values in the linked list
2 def print_linked_list(n: ListNode, l: list) -> None:
3     if n is not None:
4         l.append(n.val)
5         print_linked_list(n.next, l)
6     else:
7         print(l)
```
