

Ch 10-A: Recursion Practice

이 chapter에 있는 문제들을 coding할때에는 아래 나열된 Built-In Function들 이외의 Built-In Function 혹은 External Module에 있는 Function들을 쓰면 안됨

`print(), range(), len(), append()`

1. List 에있는 element 들의 합을 리턴하는 함수 f1 (list)를 작성하시오.

```
>>> f1([1,2,3,4])
```

```
10
```

```
>>> f1([])
```

```
0
```

2. 아래에 있는 함수 $f(n)$ 을 가지고 $f(n)$ 의 스텝 수를 1에 도달 할 때까지 리턴하는 함수 $f2(n)$ 를 작성하시오 (이는 Collatz Conjecture라고도 함).

$$f(n) = \begin{cases} n//2 & \text{if } n \text{ is even} \\ 3n+1 & \text{if } n \text{ is odd} \end{cases}$$

예를 들면, $f(6):6 \rightarrow 3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$, 1에 도달할때까지 9단계이므로, $f2(6)$ 는 9가 된다

```
>>> f2(1)
1
```

```
>>> f2(6)
9
```

```
>>> f2(11)
15
```

```
>>> f2(637228127)
276
```

3. 입력된 list 의 element들을 역순으로 print() 하는 함수 f3 (list)를 작성하시오.

```
>>> f3([1,2,3])
```

```
3
```

```
2
```

```
1
```

```
>>> f3([ ])
```

```
>>> f3([3,2,1 ])
```

```
1
```

```
2
```

```
3
```

4. 입력된 List 의 모든 홀수 element 에 3을 곱한 값을 print() 하는 함수 f4(list)를 작성하시오.

```
>>> f4([1,2,3,4])
```

```
3
```

```
9
```

```
>>> f4([2,4])
```

```
>>> f4([11,42,63,15])
```

```
33
```

```
189
```

```
45
```

5. 입력된 list의 모든 홀수 element에만 3을 곱하고 list를 수정하고, 수정된 list의 element들을 역순으로 print() 하는 함수 f5 (list)를 작성하시오

```
>>> f5([1,2,3,4])
```

```
4
```

```
9
```

```
2
```

```
3
```

```
>>> f5([2,4])
```

```
4
```

```
2
```

```
>>> f5([11,42,64,15])
```

```
45
```

```
64
```

```
42
```

```
33
```

6. 다차원 list를 같은 값을 가진 1차원 list로 반환하는 함수 f6(lst)를 작성하시오. (이것을 flattening a list 라고 한다) (Hint: 어떤 data가 list인것을 check하려고 한다면 `type([1,2,3]) == list` 를 이용)

```
>>> f6(['baa'])
['baa']
```

```
>>> f6(['baa', [4, True, [10, 5], [1, 2, ['moo']]]], ["chirp"])
['baa', 4, True, 10, 5, 1, 2, 'moo', "chirp"]
```

```
>>> f6([ ])
[ ]
```

```
>>> f6([[[[[[[[[[[[[[23]]]]]]]]]]]))
[23]
```

7. L_n 의 정의가 아래와 같을때 L_n 을 계산하는 함수 $f7(n)$ 을 구현하시오:

$$\begin{aligned} L_n &= 2 && \text{if } n=0; \\ &1 && \text{if } n=1; \\ &L_{n-1}+L_{n-2} && \text{if } n>1; \end{aligned}$$

```
>>> f7(3)
```

```
4
```

```
>>> f7(14)
```

```
843
```

```
>>> f7(0)
```

```
2
```

```
>>> f7(22)
```

```
39603
```


8. 입력된 string 이 palindrome이면 true를 return 하고, 아니면 false를 return하는 함수 f8(s) 를 구현하시오 (입력된 string은 모두 알파벳 소문자로 구성되어 있다고 가정)

```
>>> f8( " " )
```

```
True
```

```
>>> f8( " kayak " )
```

```
True
```

```
>>> f8( " penguin " )
```

```
False
```

```
>>> f8("a")
```

```
True
```

9. $n!$ 를 return하는 함수 $f9(n)$ 을 구현하시오

```
>>> f9(0)
```

```
1
```

```
>>> f9(1)
```

```
1
```

```
>>> f9(2)
```

```
2
```

```
>>> f9(3)
```

```
6
```

10. 주어진 list에서 len(list)를 return하는 함수 f10(list)을 구현하시오

```
>>> f10([1,2,3])  
3
```

```
>>> f10([ ])  
0
```

```
>>> f10([2])  
1
```

11. 주어진 list에서 last element 를 return하는 함수 f11(list)을 구현하시오

```
>>> f11([1,2,3])
```

```
3
```

```
>>> f11([])
```

```
>>> f11([1])
```

```
1
```

12. n 이 입력되면 n 에서 1까지 내림차순으로 print() 하는 함수 f12(n)을 구현하시오

```
>>> f12(3)
```

```
3
```

```
2
```

```
1
```

```
>>> f12(0)
```

```
>>> f12(1)
```

```
1
```

13. 양의 정수 n 이 입력되면 n 의 자릿수를 return 하는 함수 $f13(n)$ 을 구현하시오

```
>>> f13(9175)
```

```
4
```

```
>>> f13(34)
```

```
2
```

```
>>> f13(268)
```

```
3
```

```
>>> f13(0)
```

```
1
```

14. 주어진 list 에서 첫 번째 홀수를 리턴하는 함수 f14 (list)를 구현하시오.
(주어진 list 에 홀수가 없으면 None 을 return 함)

```
>>> f14([1,2,3])  
1
```

```
>>> f14([2,4])
```

```
>>> f14([2,4,6,8,10,3])  
3
```

15. 입력된 list 에있는 모든 홀수의 합을 리턴하는 함수 f15 (list)를 작성하시오

```
>>> f15([1,2,3])
```

```
4
```

```
>>> f15([2,4])
```

```
0
```

```
>>> f15([1,3,6,9])
```

```
13
```


16. 입력된 list 에있는 모든 홀수들로 구성된 list 를 리턴하는 함수 f16(list)를 작성하시오

```
>>> f16([1,3,5,7])  
[1, 3, 5, 7]
```

```
>>> f16([2,4])  
[]
```

```
>>> f16([1,2,3,4,5])  
[1, 3, 5]
```

17. 입력된 **list** 에서 마지막에서 두번째 **element** 를 리턴하는 함수 **f17(list)**를 작성하시오. **len(list)> 1**이라고 가정하시오.

```
>>> f17([1,2])
```

```
1
```

```
>>> f17([1,2,3,4])
```

```
3
```

```
>>> f17([1,2,3])
```

```
2
```

18. 입력된 두값 a, b 의 최대공약수를 return 하는 함수 f18(a,b) 를 구현하시오

```
>>> f18(5,4)
```

```
1
```

```
>>> f18(40,60)
```

```
20
```

```
>>> f18(9,3)
```

```
3
```

19. 주어진 list1과 list2를 오름차순으로 병합하는 함수 f19 (list1, list2)를 작성하시오. list1과 list2가 이미 정렬되어 있다고 가정하시오.

```
>>> f19([1,2,3],[4,5])  
[1, 2, 3, 4, 5]
```

```
>>> f19([4,5],[1,2,3])  
[1, 2, 3, 4, 5]
```

```
>>> f19([], [1,2,3])  
[1, 2, 3]
```

```
>>> f19([1,2,3], [])  
[1, 2, 3]
```

```
>>> f19([], [])  
[]
```

20. 수업시간에 배운 MergeSort를 하는 함수 f20(list) 를 구현하시오.
앞장에 있는 f19(list1, list2) 를 merge step에서 이용하시오.

```
>>> f20([3,2,1])  
[1, 2, 3]
```

```
>>> f20([])  
[]
```

```
>>> f20([5,3,1,2,4,6])  
[1, 2, 3, 4, 5, 6]
```