

Computing Foundations for Data Science

HW 3

제출기한: 2021/03/22 14:00 PM

주의사항

- 코드를 Jupyter Notebook 에서 작성하였더라도 python 파일(.py)로 제출할 것.
- 함수가 의도한 값을 return 하는지를 확인할 것. (print 와 혼동하지 말 것)
- 파일명은 P1.py ~ P10.py 를 유지하고, 해당 파일들을 HW3_학번_이름.zip 으로 압축하여 제출할 것. 예를 들면 학번이 2020-12345이고, 이름이 Keondo Park 이라면 **HW3_2020_12345_KeondoPark.zip** 으로 압축하여 제출.
 - 압축 시 **반드시 zip** 으로 할 것. (egg, tar, gz, rar, 등은 0점. 채점 대상 제외)
- 각 파일들은 문제를 해결하기 위한 함수만 있어야 하며 불필요한 출력이 있을 시 불이익 받을 수 있음
 - 예시) P1.py 에는 def P1() 함수만 있어야함.
 - 테스트를 위한 P1() 함수 호출이 있을 시 불이익 이 있을 수 있음
 - 제출시에 print 문은 지워서 제출할 것
- 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.
- 문제의 instruction 이 불명확하거나 clarification 이 필요할 경우 slack 을 활용하여 질문할 것.
- **늦은 제출은 받지 않음.**
- 표절 검사를 수행하여 발각될 경우 성적 **F** 부여.

문제 1.

정수(int)로만 구성된 리스트를 입력으로 받고, 리스트의 메서드(method)를 활용하여 아래의 지시를 순서대로 따르는 함수 p1을 완성하세요.

1. 입력으로 받은 리스트를 오름차순으로 정렬
2. 리스트에서 1234 제거
3. [4321, 2222]를 추가하여 리스트를 확장(extend)
4. 1번 index 위치에 1111 추가 (주의: python의 index 규칙을 잘 확인할 것)

<가정>

1. 리스트는 정수로만 구성됨
2. 각 숫자들은 리스트 내에 한 번씩만 등장하여 숫자가 중복되는 경우는 없음
3. 입력되는 리스트에는 1234가 항상 존재

예시1)

```
>>> P1([1234, 1011, 3124])  
[1011, 1111, 3124, 4321, 2222]
```

예시2)

```
>>> P1([7777, 5555, 8888, 1234, 9876])  
[5555, 1111, 7777, 8888, 9876, 4321, 2222]
```

문제 2.

세 개의 리스트를 입력으로 받고, 가장 길이가 긴 리스트의 길이를 리턴하는 함수 p2를 작성하세요.

<조건>

1. 항상 3개의 리스트가 입력으로 주어진다고 가정

예시1)

```
>>> P2([1], [2, 3], [4, 5, 6])
```

```
3
```

예시2)

```
>>> P2([10, 11, 12, 13], ['a'], [1.11])
```

```
4
```

문제 3.

정수로 구성된 리스트를 입력으로 받고 리스트 안의 값이 음수이면 양수로, 양수이면 음수로 변경하여 리턴하는 함수 p3를 작성하세요.

<조건>

1. 값이 0일 때는 변경 없음
2. 입력 받은 원본 리스트를 변경하지 말고, 새로운 리스트를 리턴하여야 함

예시1)

```
>>> P3([-1, 2])  
[1, -2]
```

예시2)

```
>>> P3([66, 0, -11])  
[-66, 0, 11]
```

문제 4.

중첩 리스트(nested list)를 입력으로 받고, 아래 조건을 만족하는 리스트를 리턴하는 함수 P4를 작성하세요.

<조건>

1. 입력 리스트는 [[영단어, 글자수]] 꼴이며, 자세한 건 아래의 예시를 참고
2. 영단어는 알파벳 소문자로만 이루어져 있음
3. 입력 리스트에서 **단어만** 모아서 **리스트로** 리턴
4. 리스트는 알파벳 순으로 정렬되어 있어야함

예시1)

```
>>> P4(['apple', 5], ['banana', 6])  
['apple', 'banana']
```

예시2)

```
>>> P4(['cup', 3], ['ace', 3], ['nice', 4], ['good', 4])  
['ace', 'cup', 'good', 'nice']
```

문제 5.

알파벳으로만 구성된 리스트를 입력으로 받고, 대문자가 처음으로 등장하는 index를 리턴하는 함수 p5을 작성하세요. (for loop을 통해서 해결 가능하지만, while loop을 활용하는 훈련을 해보는 것도 좋습니다. 어떻게 구현하여도 점수와 관계는 없습니다.)

<조건>

1. 입력 리스트는 알파벳 대문자 혹은 소문자로만 구성됨
2. 대문자가 존재하지 않는 경우는 -1을 리턴

예시1)

```
>>> P5(['a', 'B', 'c', 'D'])
```

```
1
```

예시2)

```
>>> P5(['z', 'b', 'k', 'D', 'l', 'q'])
```

```
3
```

예시3)

```
>>> P5(['a', 'b'])
```

```
-1
```

문제 6.

정수 2개를 입력으로 받고, 아래 조건을 만족하는 리스트를 리턴하는 함수 P6를 작성하세요.

<조건>

1. 입력은 항상 정수 2개
2. 입력 받은 두 정수 중 큰 수부터 시작하여 작은 수까지 포함된 리스트를 리턴
3. 리턴하는 리스트는 내림차순으로 정렬되어 있어야 함
4. 입력 받는 두 정수가 같은 경우는 해당 정수가 1번만 포함된 리스트를 리턴

예시1)

```
>>> P6(3, 9)
```

```
[9, 8, 7, 6, 5, 4, 3]
```

예시2)

```
>>> P6(2, -1)
```

```
[2, 1, 0, -1]
```

예시3)

```
>>> P6(100, 100)
```

```
[100]
```

문제 7.

리스트를 입력으로 받고, 리스트의 각 원소에 1을 더한 후 리턴하는 함수 p7를 작성하세요.

<조건>

1. 입력 받는 리스트 자체는 변경되어서는 안 됨
2. 리스트는 숫자로만 구성됨

예시1)

```
>>> P7([5, 4, 7, 3, 2, 3, 2, 6, 4, 2, 1, 7, 1, 3])  
[6, 5, 8, 4, 3, 4, 3, 7, 5, 3, 2, 8, 2, 4]
```

예시2)

```
>>> P7([0, 0, 0])  
[1, 1, 1]
```


문제 8.

숫자로 구성된 리스트를 입력으로 받고, 해당 리스트에서 음수만 제거하는 함수 p8를 작성하세요.

<조건>

1. 입력으로는 숫자만 주어짐

예시1)

```
>>> P8([1, 2, 3, -3, 6, -1, -3, 1])  
[1, 2, 3, 6, 1]
```

예시2)

```
>>> P8([-5, 1, -3, 2])  
[1, 2]
```

문제 9.

중첩 리스트(nested list)를 입력으로 받고, 안쪽 리스트 중 첫 리스트의 첫 원소와 마지막 리스트의 마지막 원소를 리턴하는 함수 p9를 작성하세요.

<조건>

1. 바깥 리스트는 항상 2개의 리스트를 원소로 가짐
2. 안쪽 리스트들은 각각 최소 1개의 원소를 가짐

예시1)

```
>>> P9(['km','miles','league'],['kg','pound','stone'])  
['km','stone']
```

예시2)

```
>>> P9([[0,1,2,3],[4,5,6]])  
[0,6]
```

예시3)

```
>>> P9([[0,0,0],[1]])  
[0,1]
```

문제 10.

rat_1, rat_2는 두 마리의 rat의 무게를 열흘 간 매일 측정하여 기록해놓은 리스트이다. 두 리스트와 1과 10 사이의 숫자가 하나가 주어졌을 때, 해당 숫자에 해당하는 날에 1번 rat의 무게가 더 높았으면 True, 아니라면 False를 리턴하는 함수 P10을 작성하세요.

<조건>

1. 두 rat의 무게가 완전히 같아지는 날은 없음

<입력 리스트 예시>

```
rat_1 = [5, 6, 7, 6, 7, 8, 10, 9, 8, 10]
```

```
rat_2 = [7, 8, 6, 7, 8, 10, 9, 8, 10, 11]
```

예시1)

```
>>> P10(rat_1, rat_2, 1)
```

```
False
```

예시2)

```
>>> P10(rat_1, rat_2, 5)
```

```
False
```

예시3)

```
>>> P10(rat_1, rat_2, 3)
```

```
True
```