

과제 #1

M3239.000300 데이터 사이언스를 위한 소프트웨어 플랫폼

Due: 2021년 3월 29일 23시 59분

1 문제 1: 최대공약수

자연수 여러 개의 리스트를 입력으로 받아 모든 쌍에 대해 최대공약수를 구한 결과를 반환하는 함수 gcd 를 작성하라. 입력 리스트의 길이가 N 이라면 길이가 N 인 리스트를 N 개 포함한 (즉, $N \times N$ 크기의) 리스트를 반환한다. 결과 리스트의 (i, j) 원소는 입력 리스트의 i 번째 원소와 j 번째 원소의 최대공약수를 의미한다. 실행 예시는 다음과 같다.

```
spdsta@login:~/2021-1/HW1$ python -i prob1.py
>>> gcd([1, 2, 4, 8])
[[1, 1, 1, 1], [1, 2, 2, 2], [1, 2, 4, 4], [1, 2, 4, 8]]
>>> gcd([36, 100, 50, 128])
[[36, 4, 2, 4], [4, 100, 50, 4], [2, 50, 50, 2], [4, 4, 2, 128]]
>>>
```

2 문제 2: 리스트 합치기

Python의 list는 다른 list를 원소로 가질 수 있으며, 이와 같은 list를 nested list라고 부른다. 아래는 정수 원소를 가지는 nested list의 구성 예시이다.

```
[1, 2, 3]
[1, [2, 3], [[4]]]
[[[1], 3], [[4, 2], 5], 6, [[7]]]
```

위와 같은 정수 nested list를 입력으로 받아 저장된 모든 정수를 합쳐서 반환하는 함수 list_accumulator를 작성하라. 실행 예시는 다음과 같다.

```
spdsta@login:~/HW1$ python -i prob2.py
>>> list1 = [1, 2, 3]
>>> list2 = [1, [2, 3], [[4]]]
>>> list3 = [[[1], 3], [[4, 2], 5], 6, [[7]]]
>>> list_accumulator(list1)
6
>>> list_accumulator(list2)
10
>>> list_accumulator(list3)
28
>>>
```

3 문제 3: 문자열 찾기

문장 내에서 단어의 등장 위치를 찾는 함수 find_word 를 작성하라. 문장에서 단어는 공백, 쉼표(,), 및 마침표(.)로 구분된다. 또한, 단어가 일치함을 확인할 때 대소문자는 구분하지 않는다. 주어진 단어가 문장 내에서 몇번째 단어로 등장하는지 그 위치들을 구해 정수의 리스트로 반환한다 (위치는 0부터 시작한다).

```
spdsta@login:~/2021-1/HW1$ python -i prob3.py
>>> s = "Some feelings are shallow, some feelings are deep. Some make us smile, some make us weep."
>>> find_word(s, "feelings")
[1, 5]
>>> find_word(s, "FeelInGs")
[1, 5]
>>> find_word(s, "some")
[0, 4, 8, 12]
>>> find_word(s, "python")
[]
>>>
```

문장은 영어 대소문자 및 공백, 쉼표, 마침표로만 구성되어 있음이 보장된다. 단어는 영어 대소문자로만 구성되어 있음이 보장된다.

4 구현 및 제출

prob1.py, prob2.py, prob3.py를 작성, 압축하여 eTL 과제함에 제출한다. 각각의 파일에는 아래와 같은 내용이 구현되어 있어야 한다.

파일	설명
prob1.py	gcd 함수 구현
prob2.py	list_accumulator 함수 구현
prob3.py	find_word 함수 구현

제출 시 아래와 같은 사항을 유의하라.

- 제출 파일에는 반드시 함수를 정의하는 코드만 포함되어야 하며, 테스트 코드 등이 포함되어서는 안 된다. 과제 채점 시 실행 예시 스크린샷에서 보인 바와 유사한 방법으로 기계적인 채점을 수행할 예정이다. 이 때 제출한 코드에 불필요한 내용이 포함되어 채점이 방해될 경우 해당 과제가 0점 처리될 수 있다. 제출용 함수 외에 별도의 함수를 구현하여 사용하고자 하는 경우, 제출용 함수의 구현 내에 nested function으로 정의하여 이를 사용하도록 한다.
- 각 함수의 인자 타입과 반환값을 잘 확인하고 이에 맞게 구현하여야 한다. gcd는 list의 list를, list_accumulator는 int를, find_word는 list를 반환하여야 한다. 반환값을 올바르게 전달하지 않는 경우 과제가 0점 처리될 수 있다(예: 값을 반환하는 대신 계산 결과를 화면에 출력).
- 함수가 실행되는 동안 화면에 아무것도 출력되지 않아야 한다. 제출 전 디버그용 print문 등을 삭제하였는지 확인하고 제출하라.
- 이번 과제의 경우 기본 Python 문법에서 제공하는 기능만을 사용하여 과제를 구현하여야 하며, 모듈을 import하여 사용하는 것은 허용하지 않는다. 기본 object들이 제공하는 method(예: string object의 split method)와 기본 문법에서 제공하는 함수(예: isinstance)를 사용하는 것은 가능하다. 이와 같은 기능들은 강의 내용에서 다루지 않았더라도 과제 구현 시 사용해도 된다.
- 올바르게 않은 입력은 들어오지 않는다고 가정한다. 예를 들어 1번 문제에서 음수에 대한 처리, 2번 문제에서 정수나 리스트가 아닌 원소에 대한 처리를 구현할 필요는 없다.
- 제출 시 세 python 파일이 저장된 압축 파일의 이름은 학번.zip (혹은 학번.tar, 학번.tar.gz)으로 설정하라 (예: 2020-20000.zip).
- 그레이스 테이블을 사용할 경우 제출 당일 메일로 조교에게 이를 알려야 하며, 메일 제목은 [SPDS] HW1-Graceday 학번 이름의 포맷으로 작성하라 (예: [SPDS] HW1-Graceday 2020-20000 안규수).