

## Assignment 2: Announcement

March 25, 2021

Coding homework 2 관련 유의하실 사항 및 각종 공지입니다. 잘 읽어보시고 과제 수행에 참고해주세요. 한국어 이해가 어려운 수강생께서는 같은 조원들의 도움을 받으시거나, 제게 개인적으로 문의주시기 바랍니다.

1. 문제에서 `print`, `len` 이외의 built-in function을 사용하는 것 혹은 임의의 module을 import 하는 것을 모두 제한하고 있는 만큼, 발견되면 ‘이번 과제’를 0점 처리 하겠습니다.
2. Ch 4의 26문제를 한 스크립트 파일로, Ch 6의 13문제를 다른 스크립트 파일로 작성해서 두 파일을 zip, tar.gz 등으로 압축하여 제출해주세요. 압축하지 않고 그냥 이메일을 보내시게 되면 악성 파일로 진단하는 문제가 있다고 합니다.
3. Python interpreter는 (적어도 기본적인 CPython 구현체의 경우에는) 기본적으로 상당히 작은 값 (ex. 1,000)으로 recursion depth를 제한하고 있습니다. 이번 과제 채점 시에는 (1) 1,000 이하의 크기의 입력만 사용할 것이며 (2) recursion depth 제한을 2,000 정도로 여유를 두고 실행하려고 합니다. 함수를 재귀적으로 작성하고 싶으신 경우 참고하시기 바랍니다.
4. 기본적으로 제출하신 코드를 모두 읽어볼 예정입니다. 그러나 정성적으로만 코드를 평가하기에 시간상, 정확도상 문제가 있을 수 있으니 blackbox test로 약간의 보조를 받으려고 합니다. 구체적으로, 5개의 간단한 테스트케이스로 주어진 입력에 대해 올바른 출력 (혹은 값의 반환)이 이뤄지는지를 확인할 것입니다.
  - 맞은 테스트 케이스의 개수에 따른 득점은 다음과 같이 정해보았습니다.

맞은 TC	5개	4개	3개	2개 이하
득점	10점	7점	3점	0점

문제당 10점, 39개 문제의 총점은 390점이 될 것입니다.

- 부담스러운 크기의 입력을 시험할 생각이 전혀 없습니다! 최대한 간명하게 작성해주세요.
- ‘예외적인 상황’은 고려하지 않으셔도 괜찮습니다. 상식적인 수준의 입력만 사용할 것입니다. 다만 예컨대 비어 있지 않은 리스트가 주어진다는 식으로 명시되어 있지 않은 경우 리스트를 인자로 받는 함수에 빈 리스트를 넣어보거나 할 수는 있습니다.
- 문제에 나와있는 입출력 예시들은 1개 정도 TC에 포함될 수도 있고 그렇지 않을 수도 있습니다. 다만, 여러분이 작성하신 함수에 치명적인 문제가 없는지 확인해보기에는 대단히 유용합니다.
- 수업시간에 풀어주신 문제도 그렇지 않은 문제와 동일하게 채점하겠습니다.

5. 채점을 굉장히 단순한 방식으로 수행할 것이라 Ch 4의 1번 문제와 같이 무언가 반환해야 하는 함수의 경우 출력하는 것이 없어야 하고, 다음 2번 문제와 같이 리턴값 없이 출력만 하는 함수의 경우 리턴값이 없어야 합니다. 그렇지 않으면 채점상 불이익이 있을 수 있습니다!
  - 채점은 여러분이 작성하신 스크립트 파일을 모듈로 import하여 각 문제별로 맞는 함수를 호출해보는 식으로 수행할 것입니다. 때문에 여러분이 작성하신 함수 중 ‘하나라도’ syntax error가 있으면 아예 채점이 되지 않습니다. 제출하시기 전에 꼭 확인 부탁드립니다.
  - 들여쓰기에 있어 공백 4칸이 표준으로 규정되어 있습니다(PEP 8). 이를 공백 2칸으로 하시거나 탭만 사용하시더라도 스크립트 실행에는 문제가 없고, 또 대부분의 편집기는 여러분이 탭을 사용하시더라도 자동으로 공백 4칸으로 바꿔줄 것이며, 서로 상관없는 블록인 경우 한쪽은 탭, 한쪽은 공백으로 들여쓰기가 되었더라도 아무 문제가 없습니다. 그러나 간혹 같은 블록에서 탭과 공백이 혼용될 수 있는데 이때는 indentation/tab error 등이 발생합니다.
  - 따라서 가급적 들여쓰기 방식을 통일하시기 바랍니다. 들여쓰기 문제로 실행이 되지 않는 경우 전체 점수에서 100점을 감점하고 들여쓰기를 통일한 뒤 다시 채점하겠습니다.
  - 그 이외의 syntax error로 인해 채점이 불가능한 경우 문제가 되는 함수 1개당 100점을 감점하는 대신 해당 함수를 새로 작성하셔서 다시 제출하실 수 있도록 하겠습니다.
  - 혹여 구현하지 못하신 함수가 있어 몇몇 함수가 빠지는 것은 괜찮습니다.
  - 모든 함수의 정의를 한 스크립트 파일에 넣음으로써 이런 불의의 문제가 발생할 수 있는 대신, 앞에서 정의한 함수를 뒤의 함수를 정의할 때 호출하는 식으로 활용하실 수 있습니다. 물론, callee의 구현에 문제가 있었다면 caller에도 예기치 못한 문제가 발생할 수는 있을 것입니다.
6. 출력이 있는 함수의 경우, 전체적인 출력 형식을 가급적 맞춰주시기 바랍니다.
7. 입력의 크기상 정상적인 코드는 눈 깜짝할 새에 전부 실행을 마칠 것이기 때문에, 대략 1초 가량을 기다려도 결과가 나오지 않는 경우는 틀린 것으로 간주하겠습니다.
8. 채점 환경이 딱히 변수가 될 것 같지 않아, 제 노트북(macOS 11.2 arm64, python 3.9.1, clang 12.0.0, (CPython))에서 채점하려고 합니다. 변동사항이 생기면 결과를 공지할 때 다시 말씀드릴 예정이니, 채점 환경 때문에 점수에 이상이 생긴 것으로 판단하시는 경우 다른 환경에서 확인해보신 후 말씀해주세요.
9. 예컨대 Ch 4의 11번과 같은 문제는 True or False 문제인데, 항상 True를 반환하도록 하기만 해도 대략 2.5개를 맞을 수 있을 것을 기대할 수 있습니다. 이런 코드를 발견하면 해당 문제를 0점 처리 하겠습니다.
10. 다른 학생들과 공유할만한 아름다운 코드를 작성하신 경우에는 혹 감점된 점수가 있다면 몇 점 덜어내는 식의 보너스 점수를 드릴 수 있도록 하겠습니다.

이번 과제를 채점해보며 문제점이 많은 것으로 드러나거나 더 나은 평가 방법이 있다고 생각되면 다음 과제부터는 다른 방법을 찾아서 진행해보려고 합니다. 조원들과 함께 즐겁게 과제를 수행하시기 바라겠습니다!