

CHW 3: Announcement

April 15, 2021

If you have any difficulties (especially in understanding this document) please let me know – you can email me at ludaya [at] snu.ac.kr. I will try my best to reply as soon as possible; yet you might have to expect a delay of one weekday or so.

1. 7-A의 문제를 풀어서 CHW1처럼 답을 서술한 pdf 파일을 `team01_7_A.pdf`로 만들어 주시고, 9-A의 함수들을 구현한 파일을 `team01_9_A.py`처럼 만들어서 zip으로 압축하여 제출하시되 압축을 풀자마자 별도의 하위 디렉토리 없이 두 파일이 보이도록 해주세요. 파일명 형식을 지켜주시면 좋겠습니다.
2. 7-A는 답과 함께 전부 간단한 설명을 달아주셔야 하며, 문제(1 슬라이드)당 5점씩 총 100점으로 하겠습니다. 2문제를 골라 채점하지 않을 것입니다.
3. CHW2를 채점하면서는 늦은 제출을 제외한 감점을 일체 적용하지 않았습니다. 부탁드린 사항이 지켜지지 않은 경우가 꽤 있었습니다만 수강생 대부분이 처음으로 코드를 제출하는 과제여서 충분히 그럴 수 있었다고 생각이 됩니다.
 - 4-A의 함수를 모아서 하나, 6-A의 함수를 모아서 하나 두 개의 `.py` 파일을 제출받는다고 안내드렸으나 그러지 않으신 경우들이 있었습니다. 저번에는 다시 제출을 받거나 채점 스크립트를 수정하여 파일별로 나뉘어져 있는 구현에 대응하도록 했지만 이번 과제부터는 파일 구조 상의 본질적인 차이가 있는 경우 채점을 하지 않겠습니다.
 - 아마도 조원별로 문제를 할당하여 나눠 구현하신 후 하나로 합쳐서 제출하셨을 것으로 생각합니다. 이 과정이 다소 번거롭고 많은 문제의 소지를 배태하는 것 같아 가급적 함수 별로 따로 `.py` 파일을 받으려고 했습니다만, 이번 과제까지는 일단 저번처럼 9-A의 모든 함수 구현을 모아서 제출해주시면 감사하겠습니다. 병합 과정을 다른 조원이 작성한 함수를 같이 검토해보는 기회로 삼으시면 어떨까 합니다.
4. 저번 과제에서 따로 안내를 드리지 않았는데 함수당 네댓 라인, 길어야 십수 줄을 넘지 않는 짧은 분량으로 인해 단정하기가 굉장히 어렵지만 그럼에도 불구하고 copy로 추정되는 사례를 발견할 수 있었습니다. 해당 조의 제출물은 앞으로 조금 더 신경써서 보겠습니다.
5. 제출하시기 전에 import가 되는지, 실행이 잘 되는지 간단히 확인해보실 수 있습니다. 별도의 스크립트를 작성하시거나 그냥 인터프리터를 실행하신 상태에서

```
1 from team01_9_A import *
2 f1(...)
3 f2(...)
```

와 같이 확인해보실 수 있습니다(파일과 같은 디렉토리에 있을 때 기준입니다). 각종 syntax error, invalid character 등으로 인해 import가 되지 않으면 이번에는 9-A에 해당하는 점수의 50%를 감점하고 다시 제출을 받으려고 하니 제출 전에 반드시 자체적으로 확인을 해보시기 바랍니다. 파일을 다 작성해두시고 압축하는 과정에서 누락하시거나 하는 경우도 마찬가지입니다.

6. 7-A에 대해서는 답안을 정리한 pdf 파일만 제출해주시면 되고 다른 파일들은 제출하실 필요가 없습니다. 9-A 파일에는 함수 구현만 포함시키시고, 함수를 호출해보거나 결과를 출력하는 등의 코드는 전부 제외해주시기 바랍니다.
7. 이번에도 blackbox test로 보조를 받으려고 합니다. 테스트케이스로는 저번과 마찬가지로 문제당 5개를 사용할 것이고, 사실 문제 자체는 4-A, 6-A에서 나왔던 것과 동일하기 때문에 일부만 수정하고 나머지는 그대로 사용하게 될 것 같습니다.
8. 함수들을 번호 순으로 배치해주시면 정말 감사하겠습니다. 순서가 뒤섞여있으면 코드 확인이 너무 어렵습니다...
9. 9-A의 문제들에 대한 채점 기준을 미리 설정하였으니 참고해주세요.

- 이번에는 한 함수를 구현하는 데 다른 구현하신 함수를 가져다 사용하시면 안 됩니다.
 - 9-A의 모든 문제는 기본적으로 “one-liner,” 즉 function body가 단 한 줄인 경우에만 만점을 드리려고 합니다.
 - 한 줄로 올바른 코드를 작성했으면 10점
 - 두 줄로 작성했지만 올바른 코드이면 8점
 - 한 줄로 작성했는데 TC 중 하나에서 틀리면 5점
 - 두 줄로 작성했는데 TC 중 하나에서 틀리면 3점
 - 세 줄 이상이거나 TC 중 두 개 이상에서 틀리면 0점: 이번 과제의 취지를 고려하며 문제들을 살펴보았습니다만 굳이 세 줄까지 필요할 일이 없을 것 같아 이렇게 정했습니다.
- 아시다시피 python에서 한 줄 짜리 블록은 다음과 같이 간단히 나타낼 수 있습니다:

```
1 if True:
2     print('hi')
```

```
1 if True: print('hi')
```

이런 코드는 두 줄로 간주하겠습니다. 세미콜론(;) 등도 마찬가지입니다. exec으로 여러 줄의 코드를 실행하거나 하는 경우도 전부 상식 선에서 판단하여 처리하겠습니다.

- 11번 문제는 조금 특별합니다! 11번 문제는 20점으로 하고, f11a와 f11b 두 가지 방법으로 구현해주시면 되겠습니다.
 - f11a는 sorted를 제외한 그 어떤 내장함수, 메서드를 사용하지 마시고 작성해주세요.
 - f11b는 반드시 all을 사용해서

```
1 def f11b(numbers):
2     return all(...)
```

과 같은 형태로 작성하셔야 하며 (1) subscript(index) (2) sorted, sort 등 일체의 정렬 관련 함수/메서드를 사용하시면 안 됩니다.

- 한 줄로 조건에 맞는 올바른 코드를 작성했으면 f11a, f11b 각 10점
- 두 줄 이상이거나, 위에 명시된 제한사항을 위반하거나, TC에서 한 개 이상 틀리면 0점
- 14번 문제도 11번 문제와 같은 방식으로 채점할 것입니다.
 - f14a는 map, lambda를 반드시 사용하셔야 합니다.
 - f14b는 enumerate을 반드시 사용하셔야 합니다.
 - 그럴 일은 없겠지만 풀이법과는 별 상관이 없는데 억지로 코드에 포함만 시키려고 하신 것으로 판단되는 경우는 점수를 부여하지 않겠습니다.
- 25번 문제도 11, 14번 문제와 같은 방식으로 채점할 것입니다.
 - f25a는 max를 두 번 사용해서 구현해주세요.
 - f25b는 max를 한 번 사용해서 구현해주세요.
 - 둘 다 다른 내장함수, 메서드는 사용하시면 안 됩니다.
- 20번 문제를 matrix[i][j], matrix[i][i]와 같이 구체적인 요소를 인덱싱하지 않고 올바르게 한 줄로 구현하신 경우 특별 점수를 드리겠습니다. 구체적으로, enumerate을 사용하신 경우 5점을 더 드릴 것이고, 그렇지 않은 경우 10점을 더 드릴 것입니다. 이로 인해 원래 정해진 만점을 초과하는 점수를 받는 조가 있으면 만점을 조정하겠습니다. 단 TC에서 하나라도 틀리시는 경우 이 보너스 점수는 받으실 수 없습니다.
- 6번은 채점하지 않습니다.
- 이외에도, 라인 수에 상관없이 지나치게 복잡한 코드는 별도의 감점이 있을 수 있습니다.
- 4-A의 8번 문제가 9-A의 8번으로 또 나왔는데 이 문제에 대해 간단히 말씀을 드리려고 합니다. 다음과 같은 입력에 대해서도 효율적으로 동작할 수 있는 코드를 생각해보시면 좋겠습니다.

```
1 f8(0, 100000000000000, 100000000000001)
2 # expected (immediate) result: 0
```

- 이번 과제에는 해당사항이 없겠지만, 예컨대 지난 과제에서 리스트를 순회하면서 최댓값을 찾는 경우에 문제에서 값들이 가질 수 있는 최솟값을 따로 명시하지 않았음에도 확인된 최댓값을 tracking할 변수를 0 등의 임의의 값으로 초기화하신 경우가 종종 있었습니다. 그런 경우 음수로만 구성된 배열에 대해서는 정작 리스트에 0은 들어있지도 않는데 0을 최댓값으로 출력하게 될 것입니다. 어떻게 초기화해야 이를 피할 수 있겠는지를 고민해보시기 바랍니다.
10. 이상에서 7-A가 100점, 9-A가 300점으로 만점은 400점입니다.
 11. 채점과 연관된 부분은 아니지만, 많은 경우에 map이나 filter 대신 list comprehension만 사용하더라도 충분한 경우가 많은데 이럴 때는 가급적 후자를 사용해보려고 하시면 좋을 것 같습니다.
 12. CHW2에서는 핵심 로직은 맞는데 리턴/출력을 혼동하신 경우 따로 확인 후 절반의 점수를 부여했습니다만 이번에는 그러기 어려울 것 같습니다. 문제 지시사항을 잘 보시고 리턴하기를 원하는 것인지 곧바로 출력하기를 원하는 것인지를 주의해서 구현하시기 바랍니다.