

기말고사 대체 과제

웹파이션프로그래밍
2021년 1학기

소프트웨어융합학과
이성원 교수

주의 사항

- 답안은 문제와 함께 배포한 answer.py의 # [START]와 # [END] 사이 구간에만 작성함
- 답안 파일의 # [START]와 # [END] 사이 구간 외 부분은 수정/삭제하지 않음
- 답안의 마지막 줄인 print("==> CLEAR!") 문장까지 에러/중단 없이 실행되어야 함
- 답안 파일 제출시 answer.py의 파일 이름을 그대로 사용함
- 문제와 대소문자가 다르거나 오탈자가 있는 경우 등에 대해서 부분 점수 반영 없음
- 전역 변수(global variable)의 사용은 금지함 (전역 변수는 삭제하고 채점함)
- 문제별 부분 점수는 없음 (총40점 만점 기준, 각 문항은 5점이며, 5점 혹은 0점만 부여)
- 소스 코드 유사도 검사 프로그램의 실행 결과, 유사도가 높은 학생들은 오프라인 (대면) 재시험을 실시하며, 컨닝 혹은 대리 작성으로 판정되면 학교 교칙에 의거 처리함
- 파이선 언어가 기본 제공하는 라이브러리 외에 인터넷 등에서 구한 오픈소스/라이브러리를 추가로 설치하여 답변 작성에서 활용하는 것은 금지함 (scipy, numpy, matplotlib 등)
- 과제의 문항과 관련하여 수업 질의응답 게시판에 질문을 하는 것은 금지함

채점 제외 대상

- eCampus의 제출 마감 시한을 넘긴 경우
- 배포한 답안지 이외의 양식으로 답안을 작성한 경우
 - 배포한 answer.py 화일이 아닌 경우, 채점 대상에서 제외함 (ipynb 화일 채점 안함)
- 배포한 답안지의 # [START]와 # [END] 사이 구간에 답안을 작성하지 않은 경우
- 제출한 답안을 Visual Code에서 그대로 읽은 후, “Run and Debug”를 실행 했을때, 배포한 답안의 마지막 줄의 print 구문까지 정상적으로 실행/완료되지 않는 경우
 - 주의 사항을 위반하여, 전역 변수를 사용한 경우, 해당 전역 변수를 삭제한 후 실행 했을 때, 마지막 줄의 print 구문까지 정상 실행이 안되면 채점 대상에서 제외함
 - 주의 사항을 위반하여, 프로그램 내부에서 input() 등으로 사용자의 입력을 요청하는 경우도, 채점 대상에서 제외함

문제1

- MyMatrix 클래스를 만들고, 해당 클래스의 멤버 함수로 getNewMatrix()를 다음의 요구사항에 맞춰 작성함
- INPUT PARAMETER: 양수이며 홀수이고 3이상인 정수 하나 (편의상 N)
- RETURN VALUE: List in List (편의상 L) 혹은 정수 -1
 - 정수 -1 : N이 정수가 아닌 경우, N이 0 혹은 1 혹은 짝수 혹은 음수인 경우
 - List in List : N에 따른 $N \times N$ matrix 임
 - (아래의) N이 3 혹은 5인 경우의 예시를 참조함
 - List in List 안의 각 값 별 위치는, N이 3인 경우 예시의 값 별 색에 준해서 작성함
 - matrix 안의 각 값들은 모두 정수임
 - 반드시 알고리즘에 의해서 생성해야 하며, 특정 N에 대한 matrix 값을 미리 저장해 두었다가 return 하는 경우는 0점 처리함

N이 3인 경우,

3	3	3
3	9	3
3	3	3

$[[3,3,3],[3,9,3],[3,3,3]]$

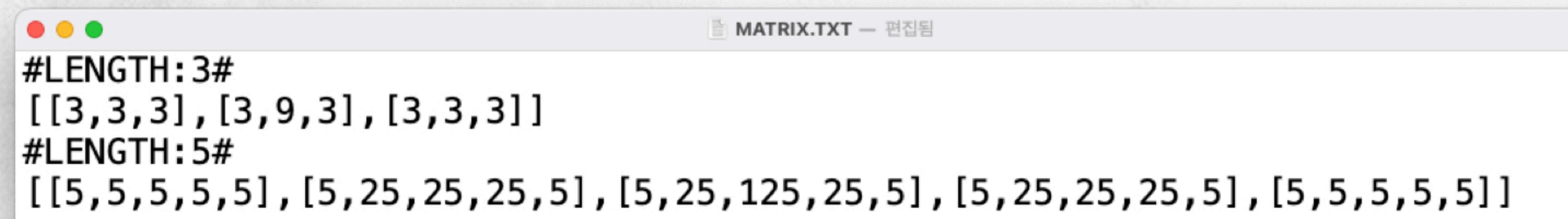
N이 5인 경우,

5	5	5	5	5
5	25	25	25	5
5	25	125	25	5
5	25	25	25	5
5	5	5	5	5

$[[5,5,5,5,5],[5,25,25,25,5],[5,25,125,25,5],[5,25,25,25,5],[5,5,5,5,5]]$

문제2

- MyMatrix 클래스의 멤버 함수로 saveFileMatrix()를 다음의 요구사항에 맞춰 작성함
- INPUT PARAMETER: 문제1을 통해서 생성한 matrix 한 개 (편의상 M)
- RETURN VALUE: 정수 한 개
 - 답안 프로그램이 실행된 후, MATRIX.TXT 파일에 저장한 matrix의 갯수
 - 아래 그림의 예시라면, 첫번째 수행시 정수 1, 두번째 수행시 정수 2를 return 함
- 수행 동작:
 - 프로그램이 수행중인 디렉토리에 위치한 'MATRIX.TXT' 파일에 M을 추가로 저장함
 - MATRIX.TXT 파일이 없으면, text 모드로 신규 생성하고 M을 저장함
 - MATRIX.TXT 파일이 있으면, 파일의 맨 끝에 M을 추가로 저장함
 - M을 추가로 저장할 때는 아래의 그림과 같이 저장함
 - 하나의 matrix 저장시 먼저 #LENGTH:N#과 같이 헤더를 저장함
 - 그림의 예시에서 첫번째 수행시 N은 3이고, 두번째 수행시 N은 5임
 - 다음 줄에 matrix의 내용을 빈칸 없이 저장함
 - 모든 항목의 대소문자 구분이 명확해야 하며, 모든 줄에 빈칸(space)은 없음
 - 모든 줄의 끝은 줄 바꾸기(\n)를 실행함



```
#LENGTH:3#
[[3,3,3],[3,9,3],[3,3,3]]
#LENGTH:5#
[[5,5,5,5,5],[5,25,25,25,5],[5,25,125,25,5],[5,25,25,25,5],[5,5,5,5,5]]
```

문제3

- MyMatrix 클래스의 멤버 함수로 readFileMatrix()를 다음의 요구사항에 맞춰 작성함
- INPUT PARAMETER: N x N matrix에서 N에 해당하는 정수 한 개
- RETURN VALUE: list in list 혹은 정수 -1
 - 문제2의 MATRIX.TXT 파일에서 N에 해당하는 matrix의 정보를 읽은 후,
 - getNewMatrix()의 return 값과 동일한 형태의 list in list로 만들어서 return 함
 - INPUT PARAMETER에 해당하는 matrix가 파일에 없으면 정수 -1을 return 함

문제4

- MyMatrix 클래스의 멤버 함수로 readFileMaxValue()를 다음에 맞춰 작성함
- INPUT PARAMETER: 없음
- RETURN VALUE: (N, v) 형태의 튜플(tuple) 한 개
 - 문제2의 MATRIX.TXT 파일에서 matrix의 element가 최대 값인 경우를 찾아,
 - 해당 최대 값일 때의 $N \times N$ matrix의 N 값과, 해당 최대 값 v 를 return 함
 - 문제2 하단의 그림 예시 기준 return 값은 (5, 125) 임

문제5

- 문제1처럼 MyMatrix 클래스의 getNewMatrix() 멤버 함수를 사용하지 않고, MyMatrix 클래스의 객체 생성시 INPUT PARAMETER로 정수인 N 값 한 개를 받아서, $N \times N$ matrix를 생성한 후, 해당 matrix를 객체 내부 정보로 저장함 [생성자]
 - INPUT PARAMETER: $N \times N$ matrix에서 N에 해당하는 정수 한 개
 - RETURN VALUE: 없음
- 상기 방법으로 만들어진, 두 개의 MyMatrix 클래스 객체 간에 == 연산자를 통한 비교가 가능하도록 함
 - 각 객체 내부 정보로 저장한 matrix 정보가 모두 같은 경우에 == 연산자는 True 임
- 상기 방법으로 만들어진, 두 개의 MyMatrix 클래스 객체 간에 != 연산자를 통한 비교가 가능하도록 함
 - 각 객체 내부 정보로 저장한 matrix 정보가 모두 다른 경우에 != 연산자는 True 임

문제6

- 문제5까지의 방법을 적용한 MyMatrix 클래스의 멤버 함수로 applyGivenFunction() 을 다음의 요구사항에 맞춰 작성함
- INPUT PARAMETER: 사용자 정의 함수 한 개 (편의상 fn)
 - fn은 정수 하나를 INPUT PARAMETER로 받음
 - fn은 사용자(프로그래머)가 정의한 동작을 INPUT PARAMETER에 적용하여 새로운 정수 값을 만들어서, 해당 값을 RETURN 함
- RETURN VALUE: list in list
 - 문제5의 방법으로 만들어진, 객체 내부 정보로 보관 중인 matrix의 각 element에 대해서 fn 함수를 적용하고, fn 함수를 적용하여 변경된 값으로, 객체 내부 정보로 보관 중인 matrix를 업데이트 함
 - fn 함수 적용을 마친, 업데이트된 객체 내부 정보 matrix를 RETURN 함

문제7

- 문제1에서 문제6 까지의 기능을 적용한, 두 개의 MyMatrix 클래스 객체 간에 + 연산자를 통한 maxtrix 더하기가 가능하도록 함
 - + 연산자는 두 개의 MyMatrix의 내부 정보 matrix를 행렬 더하기 한후, 결과에 해당하는 list in list를 RETURN 함
 - + 연산자는 두 개의 MyMatrix의 내부 정보 matrix가 ($N \times N$ 에 대해서) N 값이 다르면, 행렬 더하기를 하지 않고, 정수 -1을 RETURN 함

문제8

- 문제1에서 문제7까지 구현한 MyMatrix를 Base Class로 하는 Derived Class인 MyDerivedMatrix 클래스를 구현함
- MyDerivedMatrix 클래스도 문제5와 동일하게 생성자를 통한 객체 생성이 가능함
- MyDerivedMatrix 클래스는 다음의 getMemberMatrix() 멤버 함수를 가짐
 - INPUT PARAMETER: 없음
 - RETURN VALUE: MyDerivedMatrix 클래스 객체가 내부 정보로 보관 중인 matrix인 list in list 한 개



Thank you