[Python트랙] 과목평가4 - 알고리즘 기본

| Background

- 배열에 대한 이해와 활용
- 자료구조에 대한 이해와 활용

| Goal

- 반복문과 조건문을 이용하여 배열의 요소에 접근할 수 있다.
- 자료구조를 이해하고 문제해결에 응용할 수 있다.
- 문제의 조건을 정확히 이해하고 해결할 수 있다.

| 환경 설정

1) Pycharm(Python3.7이상)을 이용해서 코드를 작성하고 결과를 확인한다.

- 새로운 Pycharm 프로젝트를 생성 후 코드를 작성한다.

2) 파일 이름 및 제출 방법

- **1, 2번 문제에 대한 소스 파일은 Algo문제번호_지역_반_이름.py로 만든다.**
- **pypy의 경우 폴더, 프로젝트, 파일이름에 한글을 사용할 수 없으므로 algo1.py, algo2.py 로 만들고 제출시 변경한다.**

- 3번은 파이참에서 텍스트 파일로 작성한다. (메모장 사용시 최종 저장여부 확인 필요)

Algo1_서울_1반_이싸피.py

Algo2_서울_1반_이싸피.py

Algo3_서울_1반_이싸피.txt

- 위 3개의 파일만 지역_반_이름.zip으로 압축하여 제출한다.

서울_1반_이싸피.zip

(탐색기에서 파일 선택 후 오른쪽 클릭 – 보내기 – 압축(zip)폴더 선택)

3) 채점

- **주석이 없는 경우, 주석이 코드 내용과 맞지 않는 경우, 지정된 출력 형식을 만족하지 않는 경우 해당 문제는 0점 처리될 수 있다.**
- **import를 사용한 경우 해당 문제는 0점 처리될 수 있다. (import sys도 예외 없음)**

4) 테스트케이스는 부분적으로 제공되며, 전체가 공개되지는 않는다.

5) 각 문제의 배점이 다르므로 표기된 배점을 반드시 확인한다.

- **1번 40점, 2번 35점, 3번 25점**

성실과 신뢰로 테스트에 임할 것 (부정 행위시 강력 조치 및 근거가 남음)

※ 소스코드 유사도 판단 프로그램 기준 부정 행위로 판단될 시, 0점 처리 및 학사 기준에 의거 조치 실시 예정

[Python트랙] 과목평가4 - 알고리즘 기본

| 문제 1 : 사격 게임 (배점 40점)

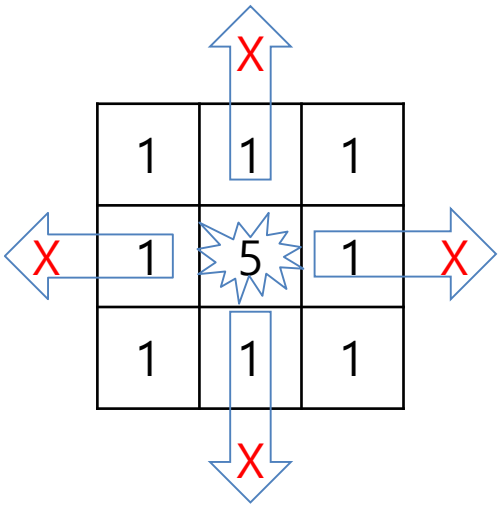
제약 사항 : 내장함수 min(), max(), sum(), abs() 사용금지

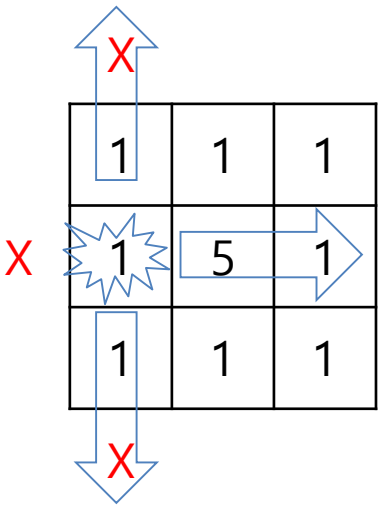
김싸피는 놀이공원에서 사격 게임을 하고 있다. 김싸피는 사격을 잘하기 때문에 보너스 단계에 도달했는데, 보너스 게임에서는 한발의 총알을 쏠 수 있다.
보너스 게임의 점수는 다음과 같이 계산된다.

- 어떤 칸을 맞히면 그 칸과 상하좌우 각 방향으로 두 칸 씩 8칸, 모두 9칸의 점수의 합이 보너스 점수가 된다.
- 상하좌우 중 일부 칸이 없는 경우, 존재하는 칸의 점수만 비교한다.

NxN 칸으로 구분된 과녁의 점수가 주어졌을 때, 김싸피가 얻을 수 있는 보너스 점수의 최댓값을 구하시오.

[예시]
다음과 같은 3x3 과녁에서는 5점 또는 5점 주변의 1을 맞추면, 보너스는 9점이다. (X는 존재하지 않는 칸)





[Python트랙] 과목평가4 - 알고리즘 기본

[입력]

첫 줄에 테스트케이스 개수 T가 주어진다. (1<=T<=10)

이어 각 테스트케이스별로 첫 줄에 N, 이어 N개의 줄에 칸 칸의 점수 Aij가 N개씩 주어진다. (3 <=N <= 50, 1<=Aij<=50)

[출력]

#과 1번 부터인 테스트케이스 번호, 빈 칸에 이어 보너스 점수의 최댓값을 출력한다.

[입력 예시]

```
3
3
1 1 1
1 5 1
1 1 1
3
1 1 1
1 1 1
1 1 1
5
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 9
```

(algo1_sample_in.txt 참고)

[출력 예시]

```
#1 9
#2 5
#3 15
```

(algo1_sample_out.txt 참고)

[Python트랙] 과목평가4 - 알고리즘 기본

| 문제2 : 행렬 탐색 (배점 35점)

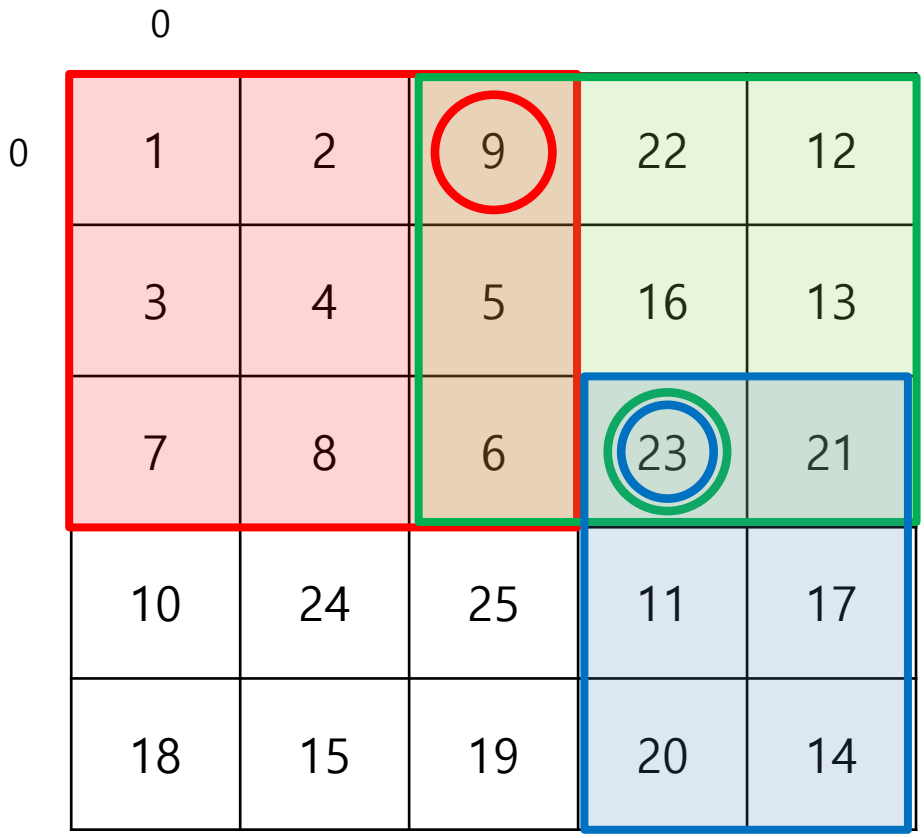
김싸피는 평소 행렬을 가지고 여러가지 놀이를 하는 것을 좋아한다. 이번에는 규칙에 맞게 움직일 때 최대값의 합을 구하는 놀이를 생각해냈다. 김싸피의 행렬문제를 풀고 합을 구해보자.

[규칙]
- N*N 크기의 행렬 A에서 M*M 크기의 행렬을 탐색하는 게임을 한다. (M<=N)
- M*M 행렬에서 최댓값을 찾아 다시 그 최대값의 위치를 시작점으로 하는 M*M 행렬을 탐색한다. (시작점은 행렬의 **좌측상단** 이다.)
- 만약 시작점이 탐색하는 행렬내에서 최댓값을 가진다면, 행렬 탐색을 종료한다.
- 최초의 시작점은 (0,0) 이다.
- 범위를 벗어나는 경우는 **A 범위내에서만 탐색을 수행**한다.
- 행렬내 같은 숫자는 없다.

[Python트랙] 과목평가4 - 알고리즘 기본

[예시] 다음은 N이 5인 행렬에서 M 이 3인 경우의 예시다.

1. (0,0) 에서 M*M 행렬 탐색시작 (빨간색 범위)
2. 행렬내 최댓값이 9이므로 누적합 9
3. 9의 위치인 (0,2)에서 다시 M*M 행렬 탐색시작 (녹색범위)
4. 행렬내 최댓값이 23 이므로 누적합 32
5. 23의 위치인 (2,3)에서 M*M 행렬 탐색 시작(파란색 범위)
6. 행렬내 최댓값이 23으로, 이전 최댓값의 **위치와 동일하므로 최대값을 더하지 않고** 탐색종료



[Python트랙] 과목평가4 - 알고리즘 기본

입력

첫 줄에 테스트 케이스 개수 T가 주어진다.

각 테스트 케이스의 첫 줄에는 N, M이 주어진다.

다음 N줄에 각 N개의 정수 Aij (-100<= Aij <= 100)가 띄어쓰기로 구분되어 주어진다.

(1 <= N <= M <= 10)

출력

각 줄에 테스트케이스 번호를 #번호 형태로 출력하고, 한 칸 띄워 누적합을 출력한다.

[입력 예시]

3
5 3
1 2 9 22 12
3 4 5 16 13
7 8 6 23 21
10 24 25 11 17
18 15 19 20 14
…

(algo2_sample_in.txt 참고)

[출력 예시]

#1 32
#2 60
#3 3

(algo2_sample_out.txt 참고)

[Python트랙] 과목평가4 - 알고리즘 기본

| 문제 3 : 서술형 (배점 : 25점)

문자열에서 반복된 문자를 지우려고 한다. 지워진 부분은 다시 앞뒤를 연결하는데, 만약 연결에 의해 또 반복문자가 생기면 이부분을 다시 지운다.

예시)

C**AA**ABBA -> CA**BB**A -> C**AA** -> C

3-1. 이 문제를 해결하는데 적합한 자료구조의 이름과 그 특성에 대해 간단히 서술하시오.

3-2. 다음과 같은 문자열에 대해 문제를 풀 때, 3-1에서 설명한 자료구조의 내부 상태 변화가 예시와 같다. 이후의 내부 상태를 더 이상 변화가 없을 때까지 [ ] 처럼 표현하라.

'ABCCB'

(예시)

[A]

A						

[A B]

A	B					

[A B C]

A	B	C				

...