



[파이썬 트랙] 2회차 – 알고리즘 기본

| Background

- ✓ 배열에 대한 이해와 활용
- ✓ 반복문에 대한 이해와 사용
- ✓ 입출력 함수에 대한 이해와 사용

| Goal

- ✓ 반복문을 이용하여 배열의 요소에 접근할 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 프로젝트와 파일이름에 한글을 사용할 수 없으므로 algo1.py, algo2.py 로 만들고 제출시 아래와 같이 변경한다.
 - 3번은 텍스트 파일로 작성한다.
 - 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 50점, 2번 30점, 3번 20점
- ## 성실과 신뢰로 테스트에 볼 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

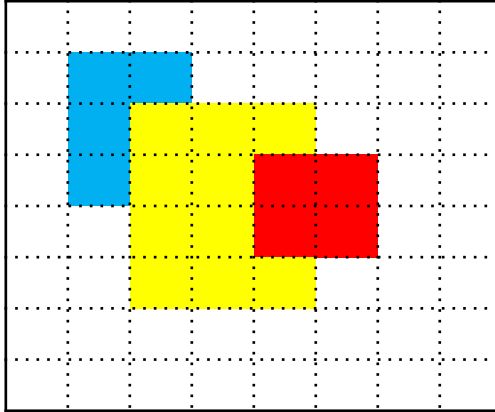
[파이썬 트랙] 2회차 – 알고리즘 기본



| 문제 1: 도장 찍기 (배점 : 50점)

싸피 유치원에 다니고 있는 파이썬반 원생들은 미술 시간에 도장 찍기를 하며 즐거운 시간을 보냈다. 원생들이 사용한 도장들은 모두 네모 모양이지만, 크기는 제 각각이다.

다음 그림1은 8x8 크기의 도화지에 파란색, 노란색, 빨간색 순서로 도장을 찍은 모습이다.



<그림1>

그림 1에서 도장을 찍어서 칠해진 칸 수는 모두 18 칸이다.

싸린이는 도화지에 도장을 찍어서 칠해진 칸이 모두 몇 칸인지 궁금해졌다.

도화지에 찍힌 도장의 위치와 크기가 주어지면,
싸린이를 위해 도장으로 칠해진 칸수를 구해보자.

도장의 위치는 왼쪽 모서리 좌표이고, 도장의 크기는 너비와 높이로 표현된다.

그림 1에서 파란색 도장의 왼쪽 모서리는 (1행, 1열)이고, 너비 2, 높이 3이다.

[파이썬 트랙] 2회차 – 알고리즘 기본



[입력]

- 첫 줄에 테스트케이스 수가 주어진다.
- 각 테스트케이스의 첫 줄에는 도화지의 크기 N 과, 도장을 찍은 횟수 M 이 주어진다.
($5 \leq N \leq 50$, $1 \leq M \leq 20$)
- 도화지는 $N \times N$ 정사각형 모양이다.
- 다음 M 개의 줄마다 도장의 위치와 크기인 왼쪽 모서리 좌표(행, 열), 너비, 높이가 공백으로 구분되어 주어진다.
- 도장이 찍히는 곳은 도화지의 경계를 벗어나지 않는다.

[출력]

'#' 과 테스트 케이스 번호를 출력하고 공백을 두고 도장으로 칠해진 총 칸 수를 출력한다.

[입력예시]

```
3
5 1
1 1 3 2
8 3
1 1 2 3
2 2 3 4
3 4 2 2
8 4
5 3 3 1
2 4 3 3
6 2 4 1
1 2 4 4
```

(algo1_sample_in.txt 참고)

[출력예시]

```
#1 6
#2 18
#3 26
```

(algo1_sample_out.txt 참고)

[파이썬 트랙] 2회차 – 알고리즘 기본





| 문제 2 : 두더지 잡기 게임 (배점 : 30점)

김싸피는 두더지 잡기 게임을 하려 한다.

김싸피가 얻을 점수를 출력하는 프로그램을 작성 하시오.

두더지 게임은 아래 조건에 맞게 동작한다.

- 게임판의 크기는 **10 * 10** 이다.
- 두더지는 머리를 게임판 위로 **N번** 내민다
- 두더지가 머리를 게임판 위로 내밀었을 때 망치로 두더지를 때리면 1점을 획득할 수 있다.
- 두더지는 머리를 내밀고 나서 **K초** 동안 머무른 뒤 다시 게임판 아래로 들어간다.
- 두더지를 망치로 때리거나, 두더지가 들어가면 다음 두더지가 **즉시** 게임판 위로 머리를 내민다.
- 망치는 **1초에 한 칸 씩** 이동할 수 있으며, 망치의 위치가 (A_1, B_1) 이고, 두더지의 위치가 (A_2, B_2) 일 때, 두더지를 때리기 위해서 걸리는 시간은 $|A_1 - A_2| + |B_1 - B_2|$ 이다.
- 망치를 움직이는 방향은 **가로방향의 차이만큼 모두 움직이고나서 세로 방향으로 움직인다.**
- 두더지가 머리를 내밀면 망치는 **즉시** 두더지를 향해 이동한다.





	0	1	2	3	4
0					
1					
2			2		
3			1		
4					

위 예시에서 망치가 (3, 3)에 위치해 있고, 두더지는 (1,2)에 위치해 있다. 망치와 두더지의 거리는 3이며, K가 3이상이라면 두더지를 잡아 점수를 획득할 수 있다.




이때, 망치는 (3,2), (2,2), (1,2) 순서로 이동한다.

[파이썬 트랙] 2회차 – 알고리즘 기본



	0	1	2	3	4
0					
1					
2					
3					
4					

K가 2라면 망치가 (2,2) 에 위치할 때, 두더지가 게임판 아래로 들어가게 되므로 점수를 획득할 수 없다. 다음 두더지가 (3,4)에서 나온다면 망치는 (2,2)에서 그 즉시 (2,3), (2,4), (3,4) 순서로 이동한다.

	0	1	2	3	4
0					
1					
2				1	2
3					
4					

[파이썬 트랙] 2회차 – 알고리즘 기본



[입력]

첫 줄에 테스트케이스 수가 주어진다.

각 테스트 케이스의 첫 줄에 N 이 주어진다. 다음 줄에 망치의 시작 위치 r, c 가 띄어쓰기로 구분되어 주어지고, 이후 N 줄에 걸쳐서 두더지의 좌표 A, B, K 가 띄어쓰기로 구분되어 주어진다.

($1 \leq N \leq 10$, $1 \leq K \leq 20$, $0 \leq A, B < 10$)

[출력]

각 줄마다 "#T" (T는 테스트 케이스 번호)를 출력한 뒤, 획득점수를 출력한다.

[입력 예시]

```
3
2
3 3
1 2 2
3 4 3
2
3 3
1 2 3
3 4 3
3
3 4
0 0 2
1 1 1
3 4 2
```

(algo2_sample_in 참고)

[출력 예시]

```
#1 1
#2 1
#3 0
```

(algo2_sample_out 참고)

[파이썬 트랙] 2회차 – 알고리즘 기본



| 문제 3 : 버블정렬(배점 : 20점)

* 다음과 같은 배열 a가 있다. 배열 a를 오름차순으로 버블정렬 하려고 한다.

55	7	78	12	42
----	---	----	----	----

1. 버블정렬을 위한 슈도코드(pseudo code)나 파이썬 코드를 완성 하시오.

a : 정렬하려는 대상 배열, N : 정렬하려는 배열의 크기

def bubble_sort(a, N):

2. 버블 정렬은 인접한 두개의 원소를 비교하여 자리를 교환하는 방식의 알고리즘이다.

교환이 일어났을 때 배열의 상태를 모두 나열 하시오.

예시)

55 7 78 12 42

- 7 55 78 12 42 55, 7 교환