

※ < 기말고사 대체 과제 >는 다음과 같이 하나의 압축 파일(과제2_학번_이름.zip)로 제출한다.

- zip 파일에는 1개의 **보고서 파일(.pdf)**, 1개의 **소스 파일(.py)**을 포함한다.
- 제출 파일명 : 과제2_학번_이름.zip
 - 보고서 파일명 : 문제N_학번_이름.pdf
 - 소스 코드 파일명 : 문제N_학번_이름.py
- 제출된 소스 파일(.py)은 **적절한 주석을 포함**해야 한다.
 - 주석 포함 내용 : 프로그램 이름, 학번, 이름, 프로그램의 **변수와 주요 코드에 대한 설명**

※ 보고서에는 다음 사항을 포함하되, 보고서 내용을 바탕으로 프로그램이 작성되어야 합니다.

1. 요구사항 분석 결과 : 프로그램의 조건과 요구사항을 정리합니다.
2. 데이터 설계 결과 : 프로그램의 주요 변수와 코드를 설계하여, 캡처/첨부하여 설명합니다.
3. 입출력 설계 결과 : 입출력값을 설계하고, 실행 예시를 캡처/첨부하여 설명합니다.
4. 알고리즘 : 프로그램의 구조를 순서도로 정리합니다.

※ 제출 기한 : 6월 20일 (일요일) 21시까지. 지각 제출은 6월 21일 오전 09시까지 허용(5점 감점)

- 아주Bb - 과제 제출 메뉴에 제출하며 다시 한번 업로드 한 파일을 반드시 확인하기 바랍니다.
- **COPY 한 코드나 보고서 발견 시, 보여준 사람과 카피한 사람 모두 0점 처리합니다.**
- 과제는 반드시 본인이 직접 작성하여 제출하기 바랍니다.
- 아주 Blackboard는 표절 제출물 자동 검사 기능 'SafeAssign'을 제공합니다.

※ 주의 : 수업 시간에 학습한 내용 이외의 기능을 활용한 경우 COPY로 간주함

※ 채점 기준

- 30점 만점 : 보고서 12점, 프로그램 18점 (평가 기준에 따라 추후 환산 반영됨)
- 감점 사항
 - 제출 형식 위반 (파일 이름 등)
 - 각 문제의 조건 누락 시
 - 코드의 주석 부재
 - 지각 제출
 - 보고서의 항목 누락 및 설명 부족

문제 1 : 탱크게임

문제 개요

탱크게임을 구현하려고 합니다. 탱크의 인덱스는 0부터 n-1로 총 n개의 탱크가 있으며, 각 네모 안의 정수는 탱크의 내구력을 나타냅니다.

포탄을 쏘는 순서는 0번부터 오른쪽으로 차례대로 돌아가며, 마지막 탱크 차례 이후에는 다시 0번 탱크로 되돌아갑니다.

포탄의 위력은 **쏘는 탱크의 내구력 X 2**이며, 포탄을 맞은 탱크의 내구력은 포탄의 위력만큼 줄어듭니다. 매 차례 타겟은 가장 높은 내구력을 가진 탱크입니다.

게임의 현황을 알기 위해 매 차례 순서와 쏜 탱크와 맞은 탱크의 인덱스, 그리고 모든 탱크의 내구력을 출력합니다.

내구력이 0이하인 탱크는 죽게 되어 포탄을 쏠 수 없으며, 마지막까지 남은 탱크가 한 대 있으면 프로그램은 종료합니다.

탱크게임의 자세한 예시는 아래와 같습니다.

| | 0 | 1 | 2 | 3 | 4 | 5 |
|-----------|----|----|----|----|----|----|
| 시작 | 57 | 68 | 72 | 40 | 20 | 17 |

시작) 탱크 6대의 내구성은 57 68 72 40 20 17로 입력

| | 0 | 1 | 2 | 3 | 4 | 5 |
|-----------|----|----|-----|----|----|----|
| #1 | 57 | 68 | -42 | 40 | 20 | 17 |

Turn 1) 0번 탱크 -> 1번 탱크 공격

| | 0 | 1 | 2 | 3 | 4 | 5 |
|-----------|-----|----|-----|----|----|----|
| #2 | -79 | 68 | -42 | 40 | 20 | 17 |

Turn 2) 1번 탱크 -> 0번 탱크 공격

| | 0 | 1 | 2 | 3 | 4 | 5 |
|-----------|-----|-----|-----|----|----|----|
| #3 | -79 | -12 | -42 | 40 | 20 | 17 |

Turn 3) 2번 탱크는 파괴되었기에.
3번 탱크 -> 1번 탱크 공격

| | 0 | 1 | 2 | 3 | 4 | 5 |
|-----------|-----|-----|-----|---|----|----|
| #4 | -79 | -12 | -42 | 0 | 20 | 17 |

Turn 4) 4번 탱크 -> 3번 탱크 공격

| | 0 | 1 | 2 | 3 | 4 | 5 |
|-----------|-----|-----|-----|---|-----|----|
| #5 | -79 | -12 | -42 | 0 | -14 | 17 |

Turn 5) 5번 탱크 -> 4번 탱크 공격.
끝. 5번 탱크 승리

프로그램 기능

▸ 탱크들의 내구력 정보인 n 개의 자연수를 입력받습니다. `map()`함수를 이용하면 쉽게 데이터 입력을 처리할 수 있으며, 사용방법은 출력예시를 참고해주세요. 간단하고 요긴하게 씁니다.

- `map(함수, iterable 객체)` : `iterable` 객체의 각 인덱스 원소를 함수로 mapping하는 함수.

- `iterable` 객체 : 순회할 수 있는 자료형으로 리스트, 문자열, 집합 등이 있음.

▸ 포탄은 0번 탱크부터 쏘며, 순서는 오른쪽으로 돌아갑니다.

▸ 포탄을 맞은 탱크는 포탄의 위력만큼 내구력이 줄어들며, 0이하의 값을 가진 탱크는 파괴되어 포탄을 쏠 수 없습니다.

▸ 매 차례 게임의 현황을 위한 정보들을 출력합니다.

▸ 살아남은 탱크가 한 대라면 프로그램을 종료하고 결과를 출력합니다.

문제 조건 :

▸ 입력은 자연수만 받으며, 그 외 입력은 고려하지 않습니다.

▸ 가장 큰 탱크가 여러 개 있을 경우, 가장 왼쪽의 탱크부터 겨냥합니다.

▸ **아래의 기능을 수행하는 함수를 포함하여 작성**하며, 기능을 함수로 분리하여 작성하면 더욱 좋습니다. 함수의 매개변수는 각자 고민해주세요.

- `def search_target()` : 목표 탱크의 인덱스를 return합니다.

- `def netx_cannoner()` : 다음 탱크의 인덱스를 return합니다.

▸ 화면의 출력문은 아래의 예시와 똑같은 필요는 없지만, 동일한 정보를 담아 출력합니다.

▸ 채점 시 입력은 랜덤으로 할 것이며, 보고서에 다양한 입력의 프로그램 실행 예시 3개를 포함해야 합니다.

출력예시 1) 문제 개요에서의 예시 동일하게 입력.

```
In [6]: tanks = list(map(int, input("탱크들을 입력해주세요. : ").split()))
        print("탱크게임이 끝났습니다.", play(tanks), "번 탱크가 이겼습니다.")
```

탱크들을 입력해주세요. : 57 68 72 40 20 17

1 : 0 -> 2

57 68 -42 40 20 17

2 : 1 -> 0

-79 68 -42 40 20 17

3 : 3 -> 1

-79 -12 -42 40 20 17

4 : 4 -> 3

-79 -12 -42 0 20 17

5 : 5 -> 4

-79 -12 -42 0 -14 17

탱크게임이 끝났습니다. 5 번 탱크가 이겼습니다.

출력예시 2)

```
In [31]: tanks = list(map(int, input("탱크들을 입력해주세요. : ").split()))
        print("탱크게임이 끝났습니다.", play(tanks), "번 탱크가 이겼습니다.")
```

탱크들을 입력해주세요. : 100 10 10 10 10

1 : 0 -> 1

100 -190 10 10 10

2 : 2 -> 0

80 -190 10 10 10

3 : 3 -> 0

60 -190 10 10 10

4 : 4 -> 0

40 -190 10 10 10

5 : 0 -> 2

40 -190 -70 10 10

6 : 3 -> 0

20 -190 -70 10 10

7 : 4 -> 0

0 -190 -70 10 10

8 : 3 -> 4

0 -190 -70 10 -10

탱크게임이 끝났습니다. 3 번 탱크가 이겼습니다.

문제 2 : 별장 지도

문제 개요

김아주(48 세)의 오랜 꿈은 별장을 짓는 것입니다. 코로나로 인한 도시생활에 피로를 느낀 김아주는 별장 짓기를 실천하기 위해 부지를 찾아 나섰습니다.

김아주는 벌레를 싫어해 별장 부지 안에 산이 있기를 싫어하지만, 낚시를 좋아해 호수는 별장 부지에 있어도 상관없다고 합니다.

마을과 산, 호수, 별장은 도식화의 편리를 위해 모든 변이 x 축과 y 축에 평행한 직사각형이며, 마을은 가로 70km, 세로 30km의 조용하고 자그마한 마을이다.

마을 지도의 정보를 입력 받아 지도를 그리고, 김아주가 원하는 별장 부지를 지도에 그리는 프로그램을 만들어주세요.

프로그램 기능

- x 축과 y 축은 아래의 사진과 같습니다.



▸ main()함수에서 산, 호수, 별장 좌표 입력과 지도 출력, 프로그램 종료 중 하나의 옵션을 고를 수 있습니다. 그 외의 입력은 고려하지 않습니다.

▸ 마을지도의 정보는 dictionary 자료형으로 관리하며 산, 호수, 별장을 key로, 그 위치인 좌측 상단, 우측 하단 두 개의 좌표는 (x, y) 형태로 튜플 자료형으로 저장하며, 2 개의 좌표를 리스트로 묶어서 value로 저장합니다.

```
>>> town_map.keys()
dict_keys(['mount', 'lake', 'villa'])
>>> town_map['mount'] # 산이 두 개인 경우
[(0, 0), (10, 10)], [(60, 20), (70, 30)]
```

▸ 좌표는 0~100의 정수가 입력되며 그 외의 입력은 고려하지 않습니다.

▸ 좌표는 좌측 상단과 우측 하단 두 개의 점을 순서대로 띄어 써서 입력하고, 그 외의 입력은 고려하지 않습니다.

Ex) >>> x1 y1 x3 y3

▸ 산과 호수의 좌표가 입력될 때, 아래의 경우를 고려해야 해야 합니다.

- 1) 좌표가 사각형을 형성하는지
- 2) 산과 호수가 겹치는지

문제 조건

▸ 마을의 크기는 가로 70, 세로 30으로 고정하며, 지도의 경계는 별표(*)로 표시합니다.

- 즉 지도의 사이즈는 가로 72, 세로 32가 됩니다.

▸ 산은 캐럿기호(^)로 면적 모두를 표시하며, 별장 부지 안에 산이 있을 수 없으며, 마을 안에 여러 개의 산이 있을 수 있습니다.

▸ 호수는 물결(~)로 면적 모두를 표시하며, 별장 부지 안에 있을 수 있으며, 마을 안에 여러 개의 산이 있을 수 있습니다.

▸ 별장은 알파벳 소문자 오(o)로 표시하며, 울타리만 지도에 표시합니다.

▸ 서로 다른 산과 호수는 면적이 겹칠 수 없다.

▸ main함수를 포함한 최소 3개의 사용자 정의 함수를 사용하여 프로그램을 구현합니다.

- 사용자 정의 함수는 더 많아도 좋습니다.

출력예시 1) 산[(1, 1), (11, 6)], 산[(12, 0), (30, 10)] 입력

별장 지도 프로그램입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

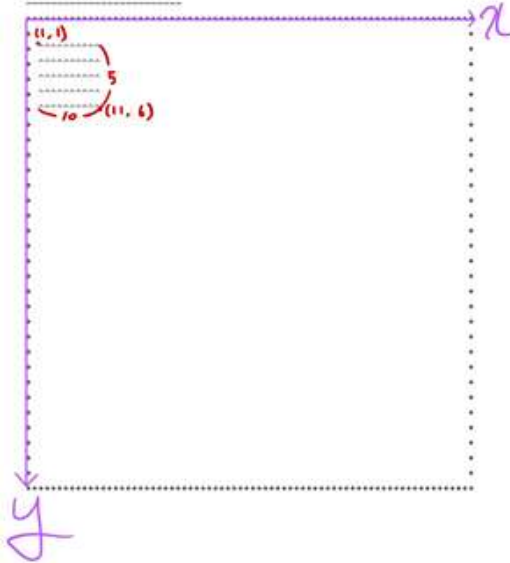
종료를 선택해주세요. : 1

산의 왼쪽 상단(x1, y1)과 오른쪽 하단(x2, y2) 좌표를 띄어쓰기로 입력해주세요. : 1 1 11 6

별장 지도 프로그램입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

종료를 선택해주세요. : 4



별장 지도 프로그램입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

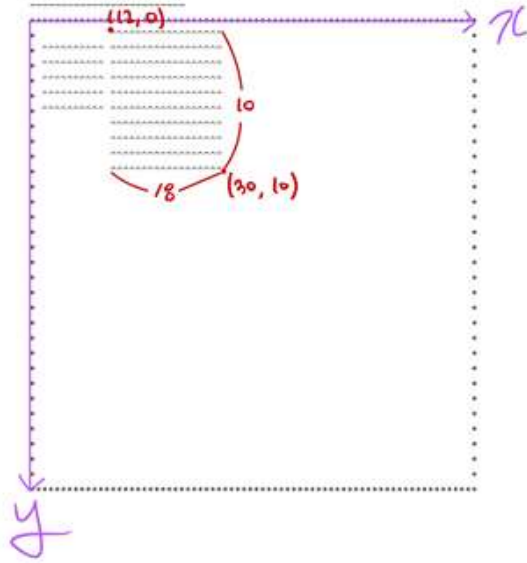
종료를 선택해주세요. : 1

산의 왼쪽 상단(x1, y1)과 오른쪽 하단(x2, y2) 좌표를 띄어쓰기로 입력해주세요. : 12 0 30 10

별장 지도 프로그램입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

종료를 선택해주세요. : 4



출력예시 2) 산[(31, 0), (70, 20)] 추가, 호수[(0, 20), (60, 30)] 입력

별장 지도 프로그램입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

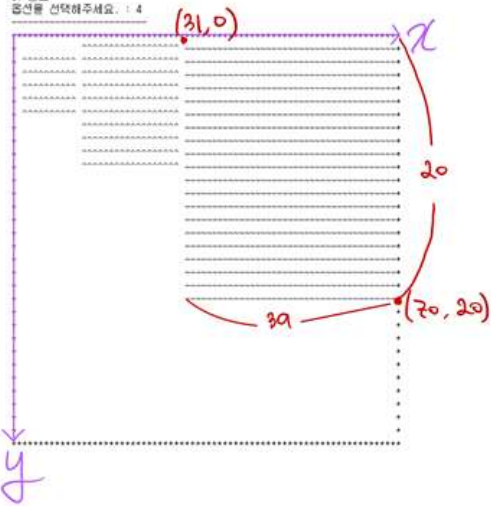
종료를 선택해주세요. : 2

호수의 왼쪽 상단(x1, y1)과 오른쪽 하단(x3, y3) 좌표를 띄어쓰기로 입력해주세요. : 31 0 70 20

별장 지도 프로그램입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

종료를 선택해주세요. : 4



별장 지도 프로그램입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

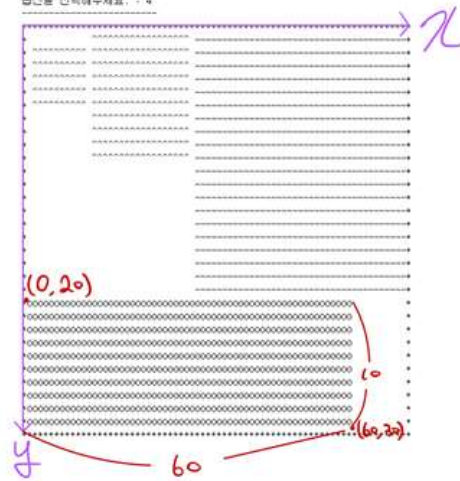
종료를 선택해주세요. : 3

별장의 왼쪽 상단(x1, y1)과 오른쪽 하단(x3, y3) 좌표를 띄어쓰기로 입력해주세요. : 0 20 60 30

별장 지도 프로그램입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

종료를 선택해주세요. : 4



출력예시 3) 호수[(40, 25), (70, 30)], 호수[(30, 25), (31, 26)] : 별장과 겹치는 호수 추가

별장 지도 프로그램입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

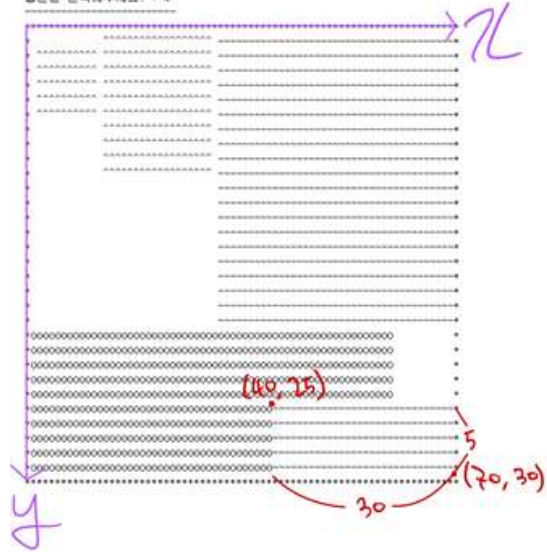
종선을 선택해주세요. : 2

호수의 왼쪽 상단(x1, y1)과 오른쪽 하단(x3, y3) 좌표를 띄어쓰기로 입력해주세요. : 40 25 70 30

별장 지도 프로그램입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

종선을 선택해주세요. : 4



별장 지도 프로그램입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

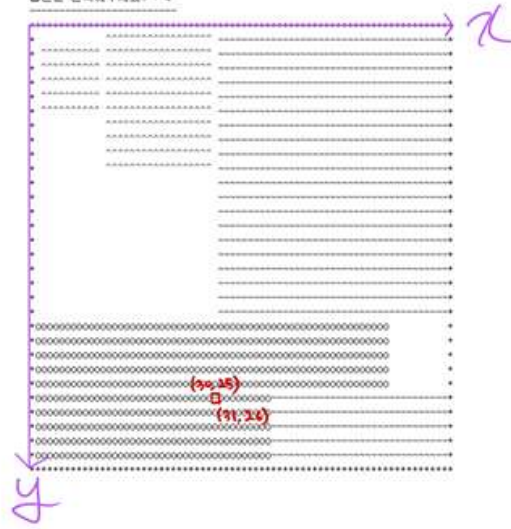
종선을 선택해주세요. : 2

호수의 왼쪽 상단(x1, y1)과 오른쪽 하단(x3, y3) 좌표를 띄어쓰기로 입력해주세요. : 30 25 31 26

별장 지도 프로그램입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

종선을 선택해주세요. : 4



출력예시 4) 산의 면적이 겹칠 경우 (호수도 같음)

별장 지도 프로그램 입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

옵션을 선택해주세요. : 1

산의 왼쪽 상단(x1, y1)과 오른쪽 하단(x3, y3) 좌표를 띄어쓰기로 입력해주세요. : 0 0 5 5
산의 면적이 겹칩니다.

출력예시 5) 두개 이상의 별장이 입력된 경우

별장 지도 프로그램 입니다.

1. 산 위치 입력
2. 호수 위치 입력
3. 별장 위치 입력
4. 마을 지도 출력
0. 종료

옵션을 선택해주세요. : 3

별장의 왼쪽 상단(x1, y1)과 오른쪽 하단(x3, y3) 좌표를 띄어쓰기로 입력해주세요. : 0 15 10 18
이미 별장 부지를 정했거나, 산과 면적이 겹칩니다.