**2023년 2학기**

**프로그래밍과 문제해결**

**Assignment #2**

**담당교수: 윤은영**

**학번: 20230563**

**학과: 무은재학부**

**이름: 김홍근**

**POVIS ID: hongsimi7**

**명예서약(Honor code)**

**“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”**

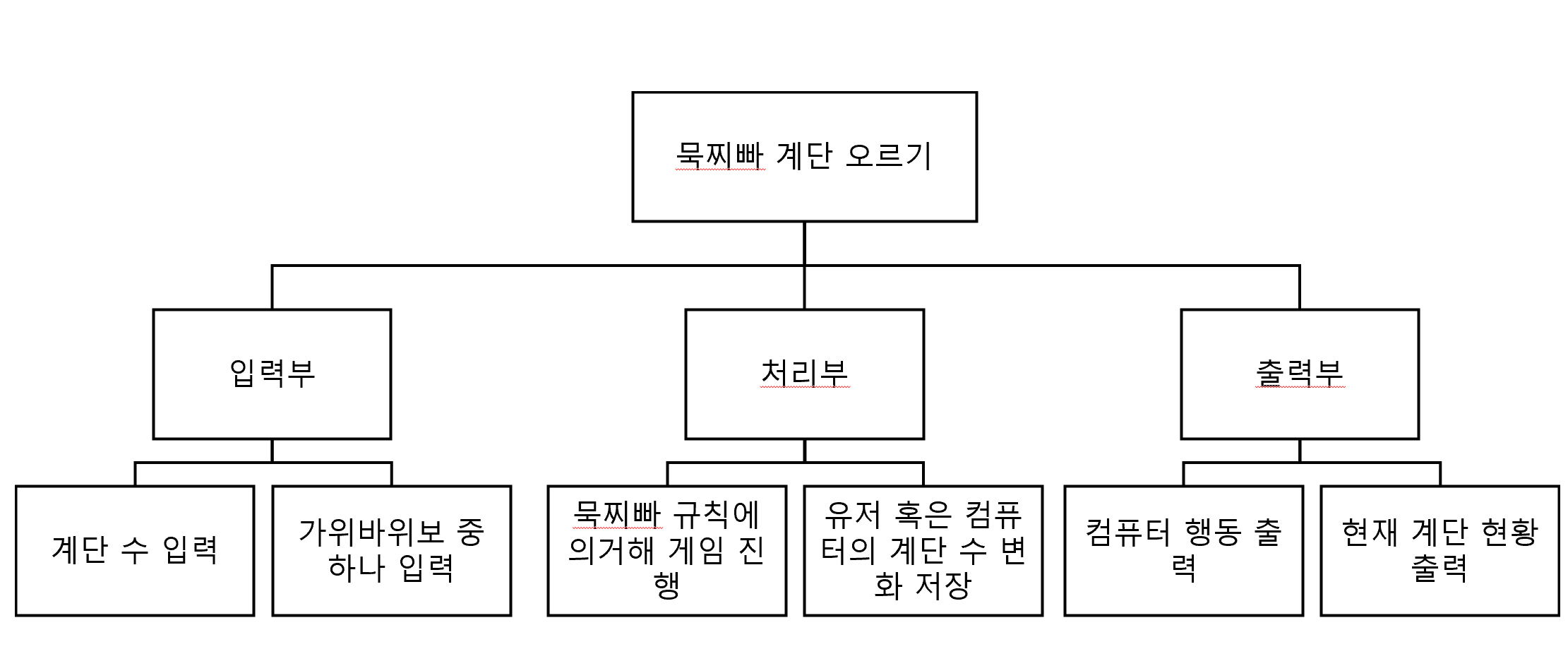
**Problem 1: 묵찌빠 계단 오르기**

**1. 문제의 개요**

본 프로그램을 간략히 설명하면 다음과 같다.

* 플레이어는 10 이상 30 이하의 계단 수를 입력 받아 계단을 출력한다.
* 공격권 결정 가위바위보를 통해 공격권을 유저와 컴퓨터 중 공격권을 결정한다.
* 묵찌빠 수행 중 같은 선택을 할 경우, 전 차례에 공격권을 가진 사람이 승리한다.
* 묵찌빠 수행 중 다른 선택을 할 경우, 오르는 계단 수를 하나 추가하고 다시 묵찌빠를 수행한다.
* 묵찌빠를 수행한 만큼 계단 수를 오르며, 입력받은 계단을 먼저 오르는 사람이 승리한다.

이 때 사용되는 구상 가능한 구조 차트(structure chart)는 아래와 같이 표현될 수 있다.



* 입력부: 유저가 게임에서 진행할 10이상 30이하의 계단 수를 입력 받는다.

공격권 선정 및 묵찌빠 진행을 위해 유저에게서 가위바위보 중 하나를 입력 받는다.

* 처리부: 묵찌바 규칙에 의거해 게임을 진행한다.

묵찌빠 중 같은 선택이 나왔다면, 전 차례에 공격권을 가진 사람이 승리한다.

묵찌빠 중 다른 선택이 나왔다면, 다시 한 번 묵찌빠를 시행한다.

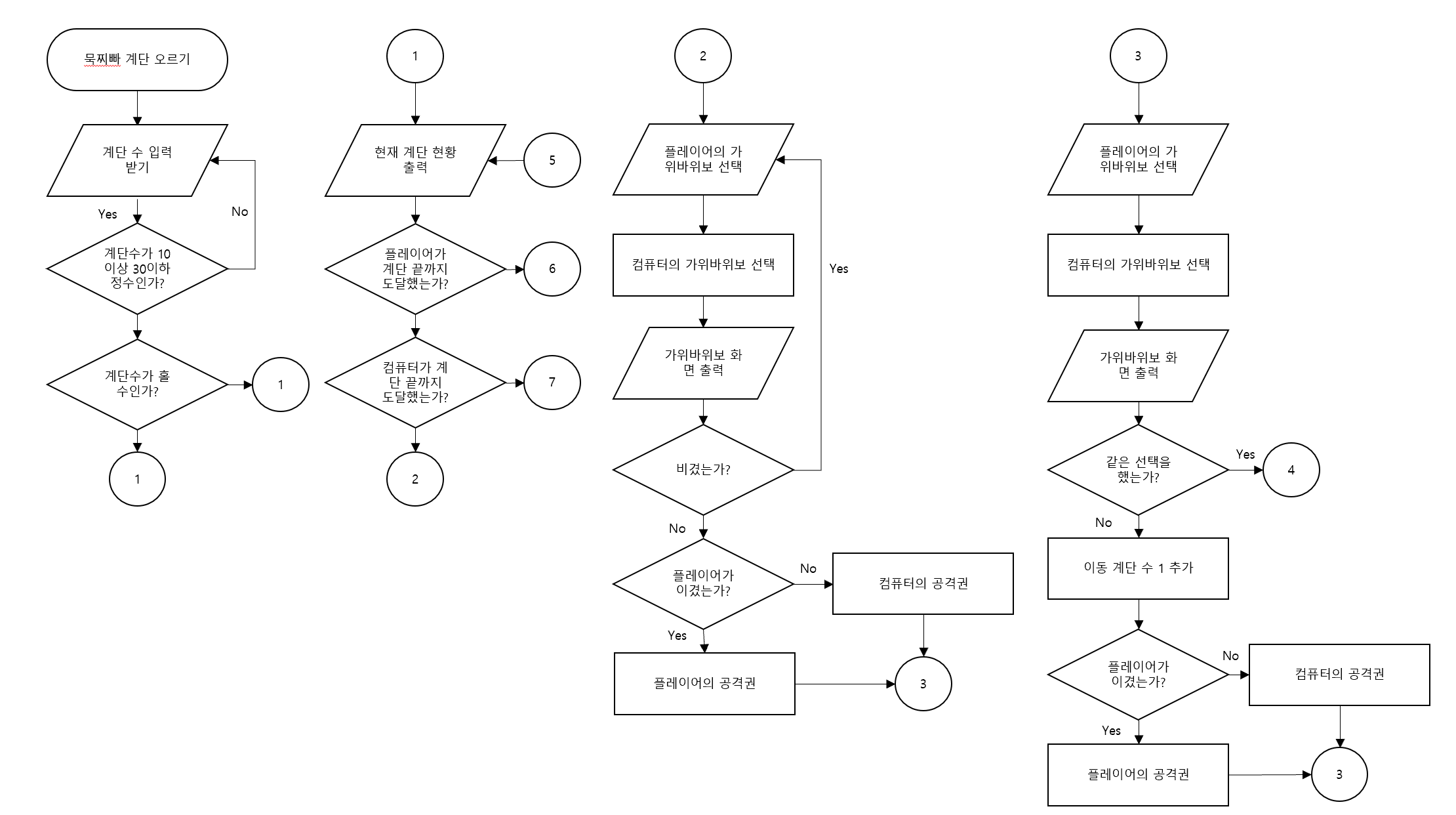
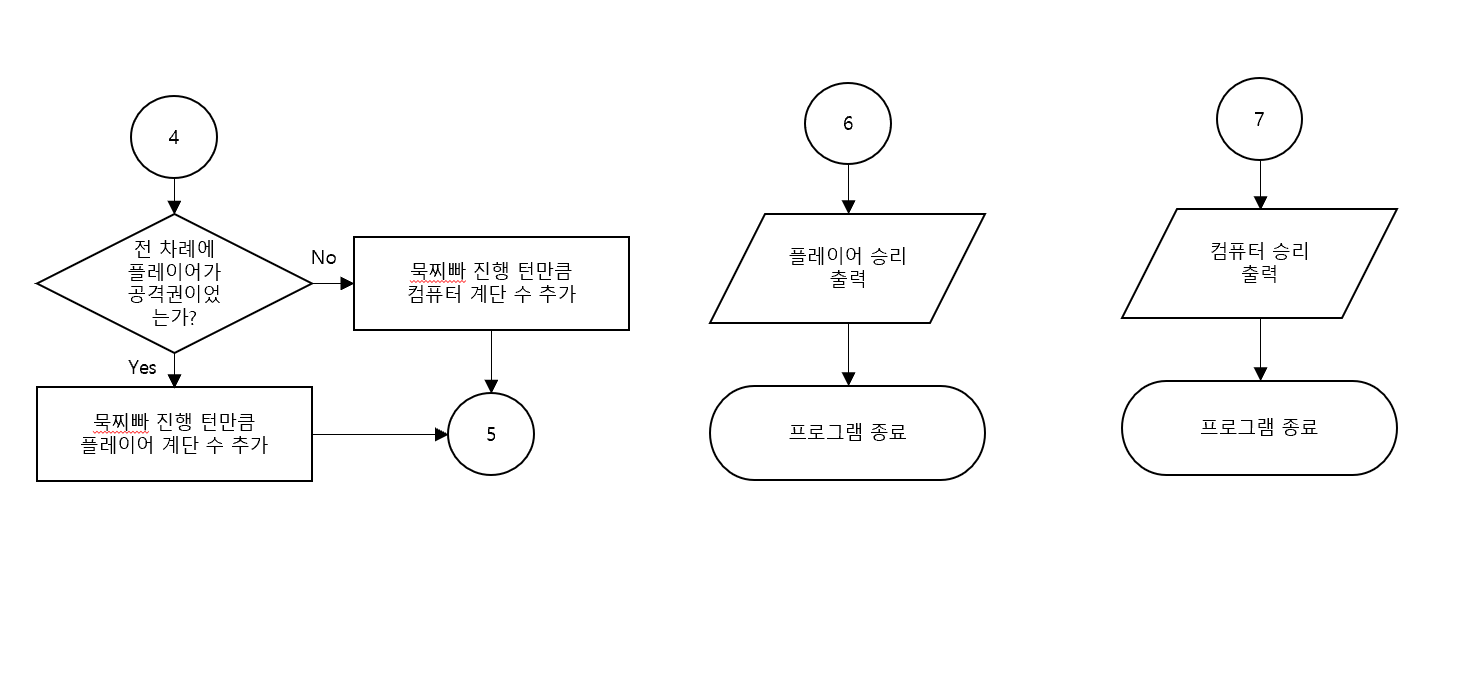
묵찌빠를 진행한 시행만큼 계단 수를 오르기 위해 계단 수 변화를 저장한다.

* 출력부: 묵찌빠 게임 중 컴퓨터의 가위바위보 선택을 화면에 출력한다.

현재 계단 현황을 화면에 출력한다.

**2. 알고리즘**

본 프로그램 작성을 위한 알고리즘을 Flowchart 형태로 나태내면 다음과 같다.

**3. 프로그램의 구조 및 설명**

a) 계단 수 및 현재 플레이어와 컴퓨터의 계단 위치 출력

- print\_stairs 함수를 통해 플레이어와 컴퓨터의 계단 위치, 계단 수를 입력 받아 화면에 출력한다.

- 계단 수가 짝수일 때와 홀수일 때가 다르며, 말이 겹쳐졌을 때, 색이 반만 칠해져 있는 말을 사용한다.

b) 가위바위보 화면 출력

- print\_scissors, print\_rock, print\_paper 함수를 통해 가위바위보를 출력한다.

c) 컴퓨터의 가위바위보 선택

- computer\_choice 함수를 이용해 컴퓨터가 가위바위보 중 하나를 선택하게 한다.

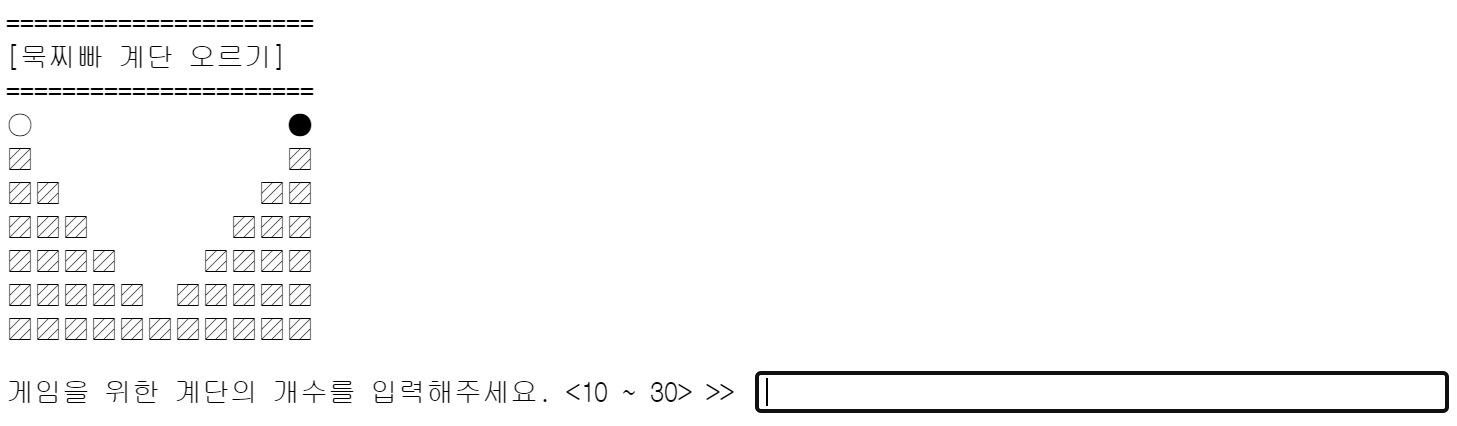
- computer\_choice 함수를 만들 때, rand 함수를 사용해 랜덤으로 가위바위보가 선택되게 한다.

d) 화면 지우기

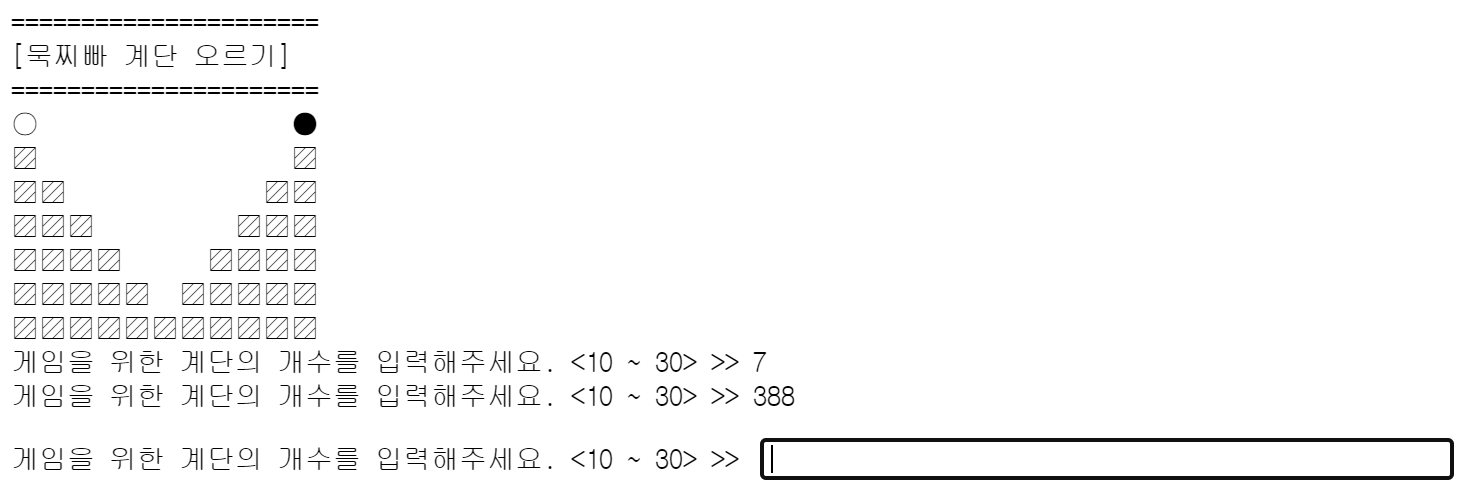
- clear\_screen 함수를 이용해 ‘Enter’ 키를 받으면 출력된 화면을 지우도록 한다.

**4 프로그램 실행 방법 및 예제**

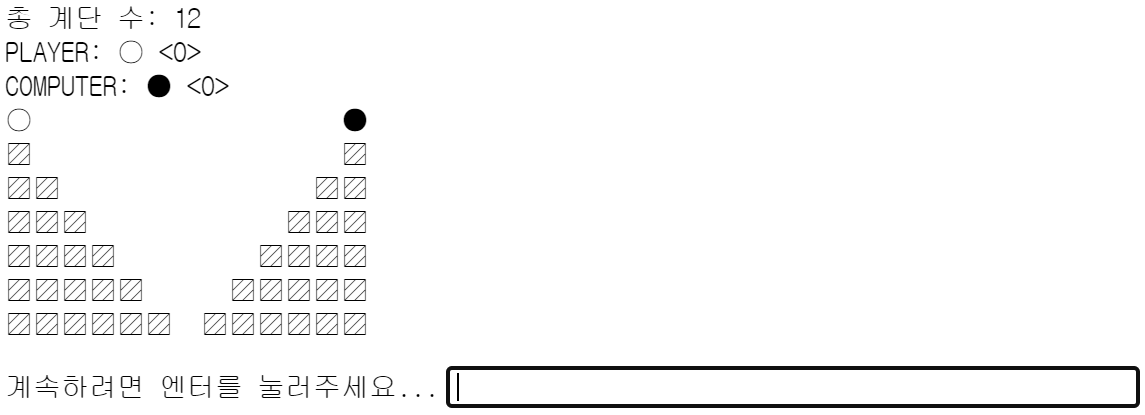
* Anaconda prompt의 Jupyter notebook 환경에서 프로그램을 실행한다.
* 프로그램을 실행했을 때, 시작 화면은 다음과 같다.



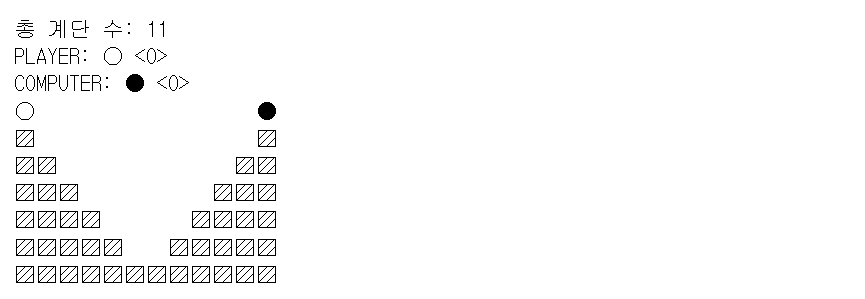
* 만약 10이상 30이하의 정수를 입력 받지 못했다면, 다음과 같이 다시 입력 받는다.



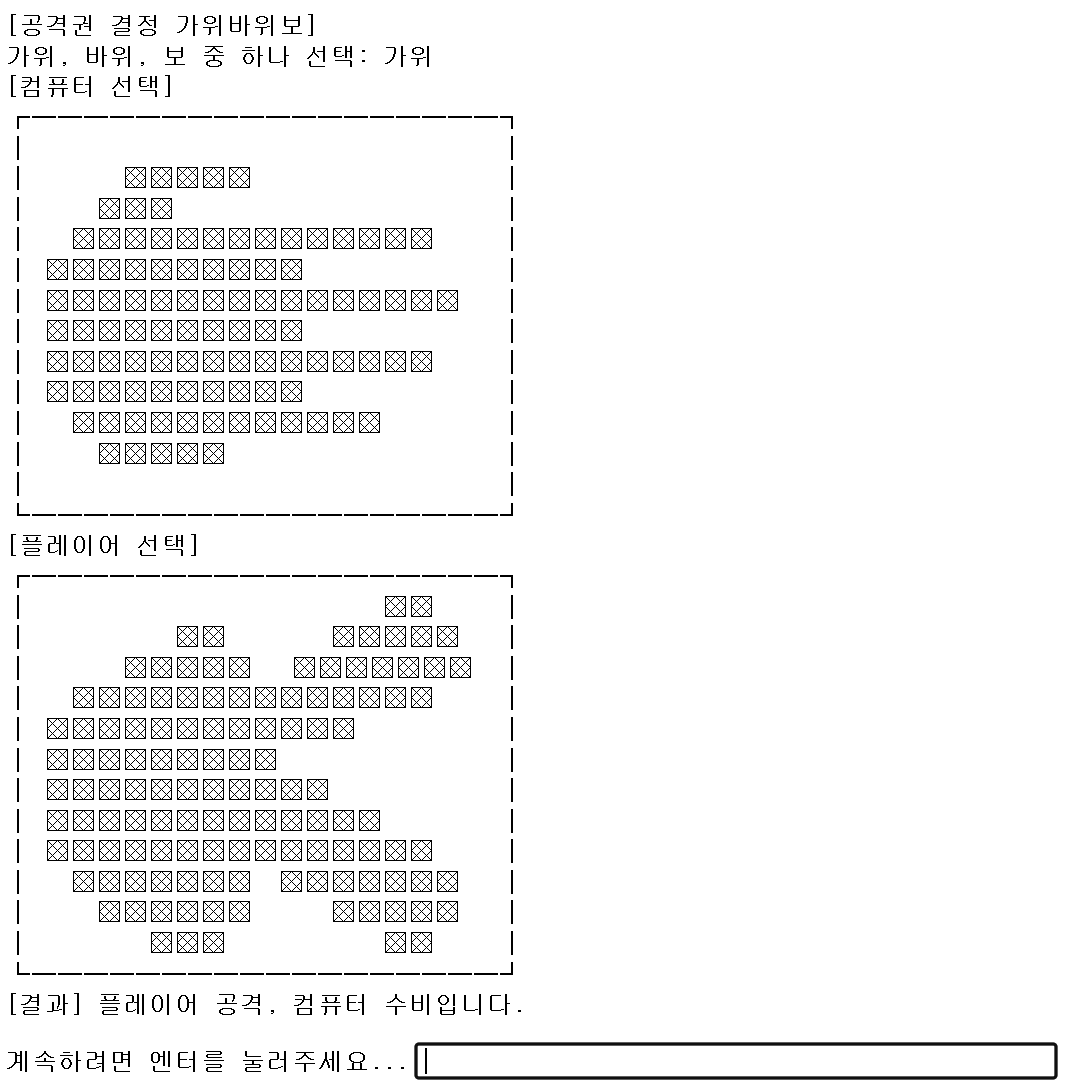
* 범위 내의 개수를 입력 받았다면, 화면이 지워지고 다음과 같이 화면이 출력된다.

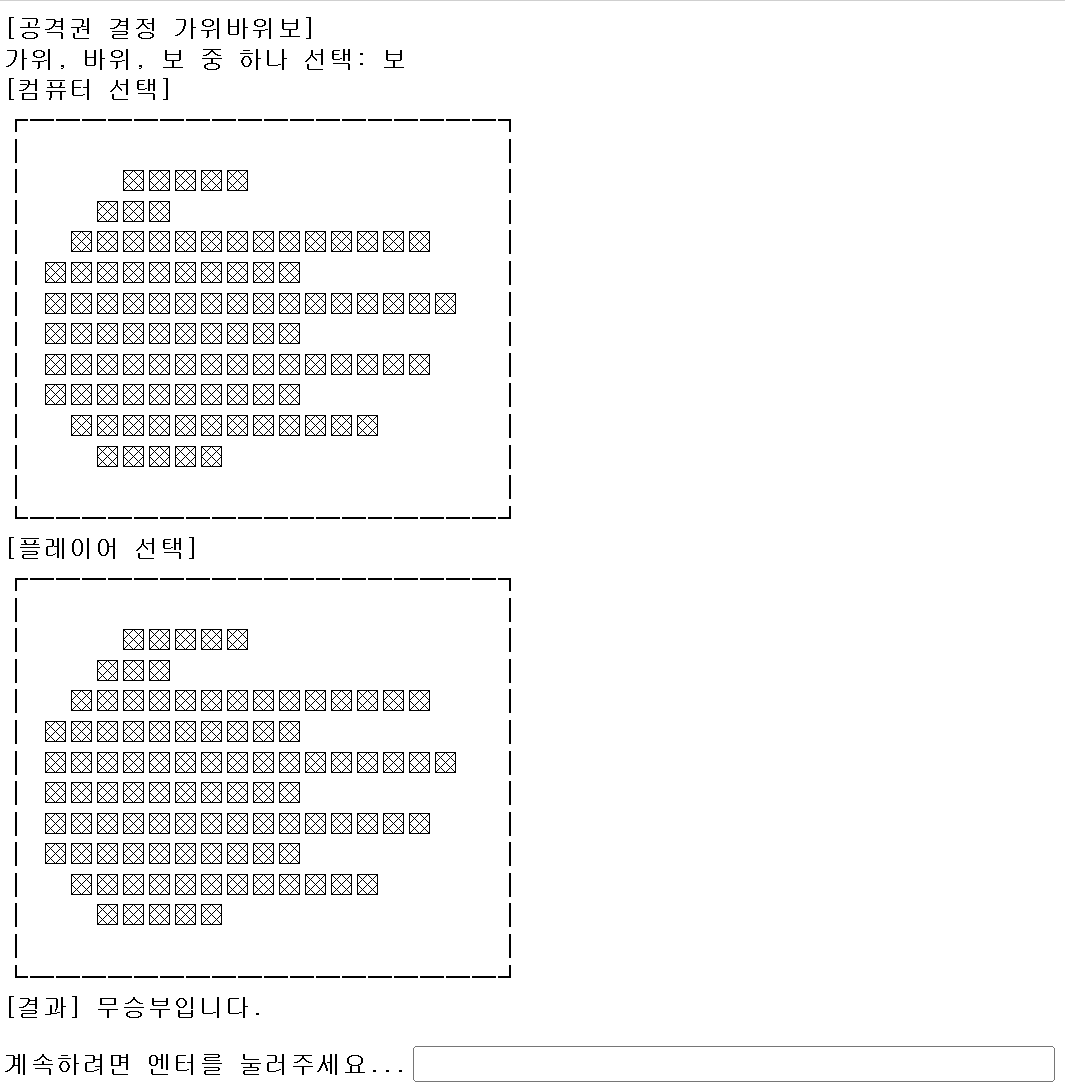


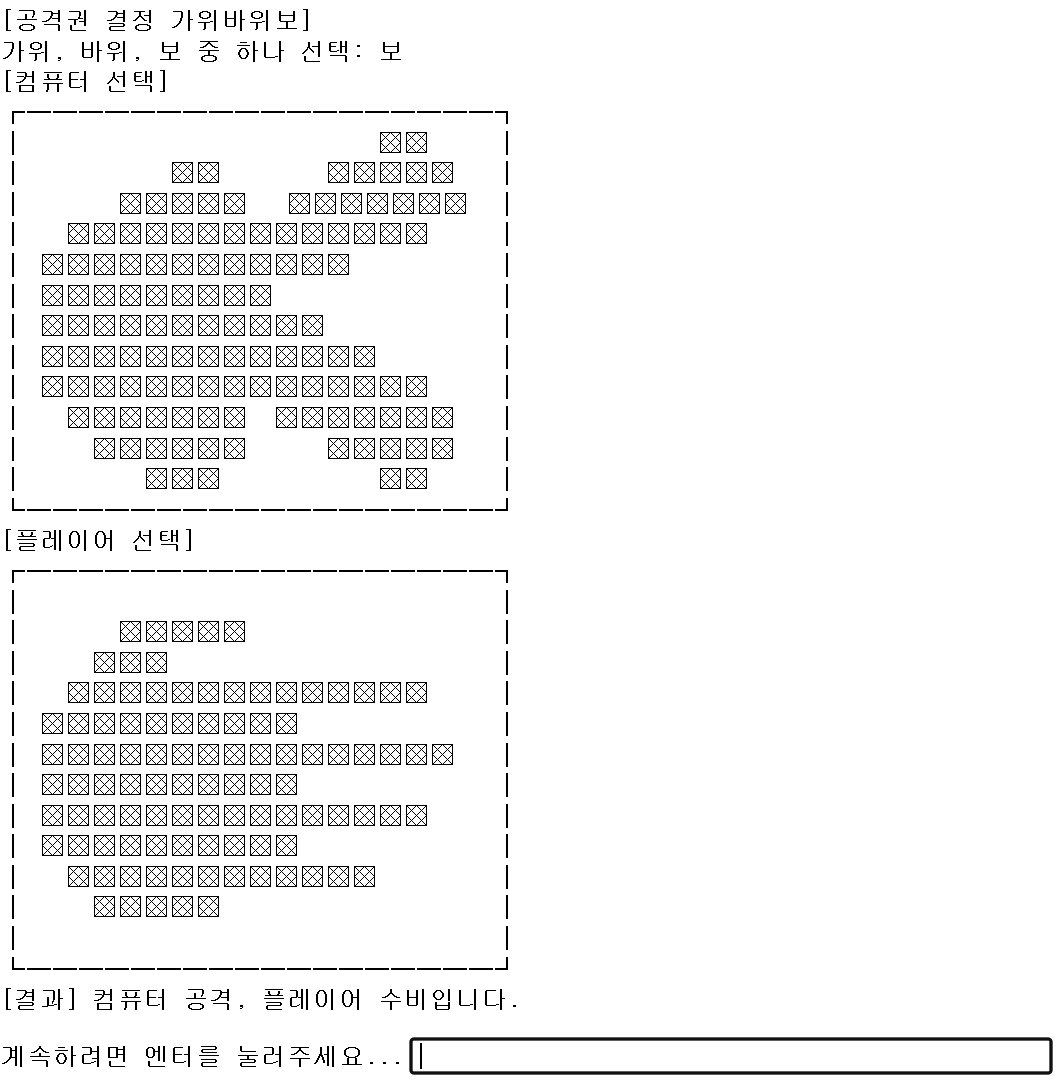
- 만약 입력 받은 수가 홀수라면 다음과 같이 출력된다.



- 게임이 시작되었다면, 다음과 같이 공격권 가위바위보를 실시한다. 플레이어의 선택 후 플레이어와 컴퓨터가 각각 무엇을 냈는지 출력한다. 아래의 세 예시는 차례대로 승리, 무승부, 패배이다. 만약 무승부 시에는 공격권 가위바위보를 다시 시행한다.



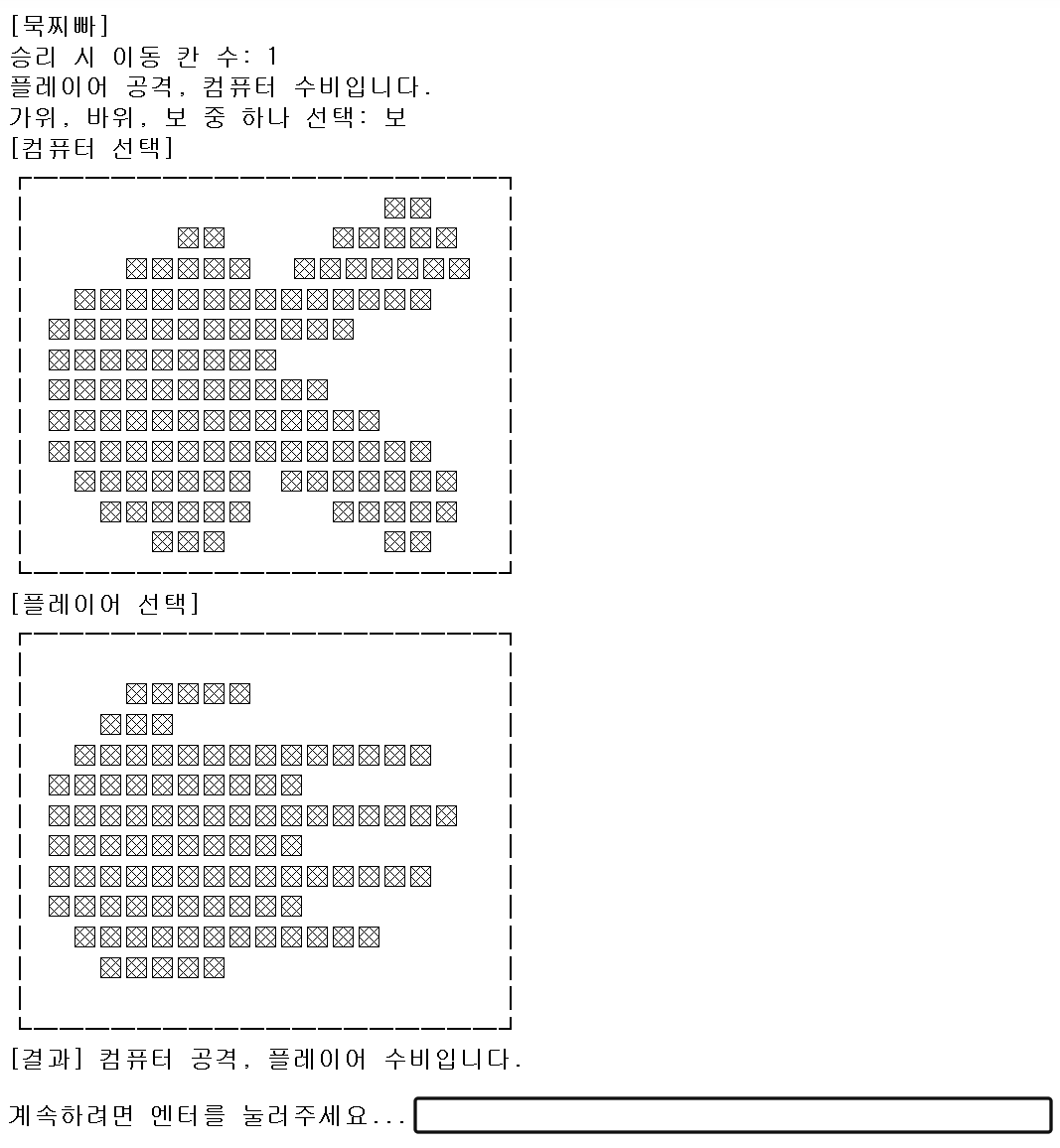




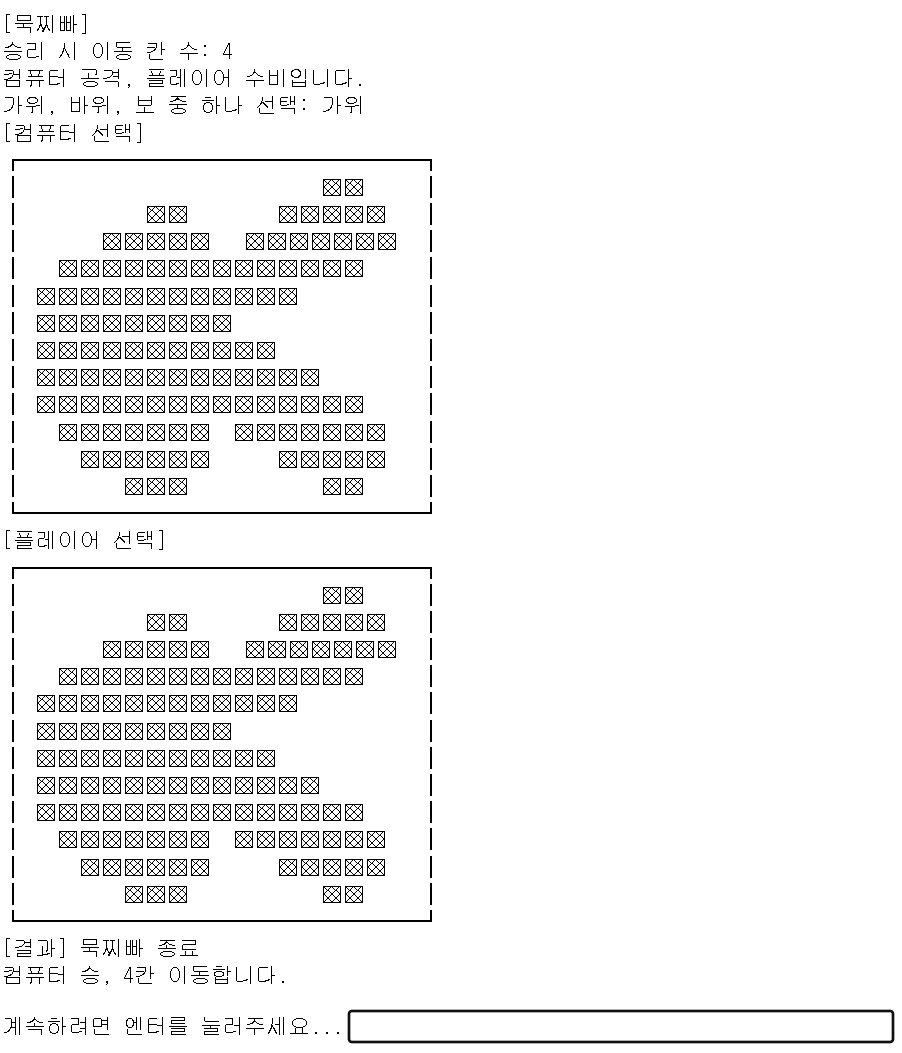
- 공격과 수비가 결정된 상태에서 엔터를 입력 받으면, 화면을 지우고 다음과 같이 화면이 출력되며, 다시 가위바위보를 받는다. 승리 시 이동 칸 수는 1로 시작되고, 승부가 나지 않을 때마다 1씩 추가된다.



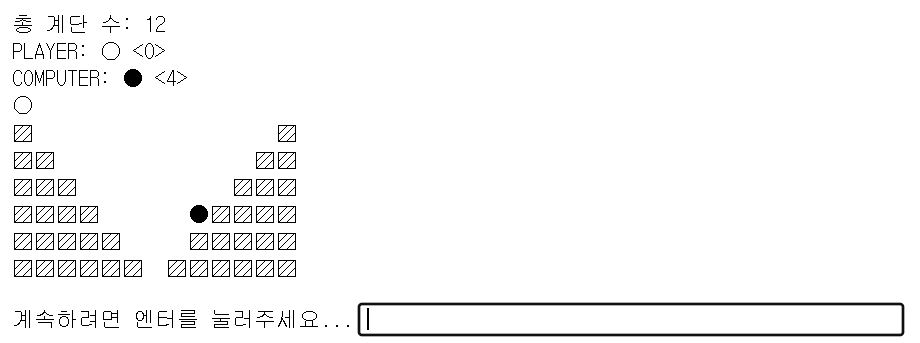
- 다음은 플레이어가 공격을 실패하고 컴퓨터에게 공격권이 넘어간 상태이다. 이 때마다 승리 시 이동 칸 수는 1씩 증가한다.



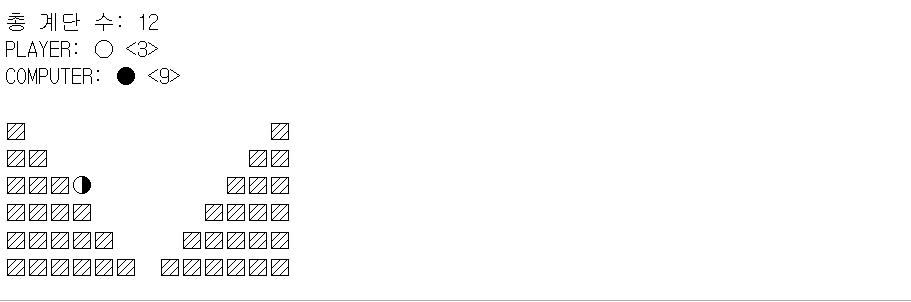
- 만약 같은 선택이 나왔다면 전 차례에 공격을 가진 쪽이 승리하게 되고, 묵찌빠를 진행한 만큼 이동 칸 수를 가지게 된다. 아래는 컴퓨터가 이긴 예시이다.



- 다음은 컴퓨터가 4칸 이동한 예시이다.



- 다음은 말이 겹쳐졌을 때의 예시이다.



- 만약 플레이어 혹은 컴퓨터가 끝까지 도달하였다면, 게임을 승리하게 된다. 다음 예시는 컴퓨터가 승리한 상황이다. 그리고 나서 게임을 종료한다.



**5. 토론**

- 모듈을 사용하는 방법을 알았고, random 모듈을 불러와 컴퓨터의 행동을 랜덤으로 설정할 수 있었다.

- 2차원 리스트의 활용 방법을 알게 되었다.

- 파이썬은 C언어와 다르게 변수 타입을 지정하지 않아도 자동으로 지정된다는 사실을 알았다.

- while True 식을 사용하면, break를 사용하지 않는 이상 계속 프로그램이 돌아간다는 것을 알았다.

**6. 결론**

- 본 과제에서는 if-else문, while문, 함수 지정, print와 input를 사용하는 방법을 익히는데 유용했으며, 알고리즘을 작성 및 디버깅하는 방법을 학습할 수 있었다.

**7. 개선방향**

- 본 과제를 할 때, 무분별한 복사 및 붙여넣기로 효율적인 코드를 짜지 못했다. 반복문을 더 활용할 수 있도록 한다면 더욱 깔끔한 코드로 디버깅이 쉬워질 것으로 예상한다. 또한, 함수 지정으로 더 효율적인 코드를 짤 수 있을 거라 예상한다.