객체지향 프로그래밍 1차 과제

실습 교실 : 328호

실습 조교 : 이성빈 조교

학과 : 디지털 미디어 학과

학번 : 201421105

이름: 김동해

1. 서론

2018-2학기 프로그래밍 첫 과제는 간소화한 장기 게임을 만드는 것이다. 장기 게임에서 (말 마)와 같은 역할을 하는 말 2개를 만들고 2개의 말이 움직일 수 있는 모든 경우의 수를 검사해 경우의 수에 상대팀 말이 있으면 경기를 종료하도록 게임을 만들었다. 나는 전체적인 장기 게임을 만들기 위해 사용 해야하는 Class, 사용 해야하는 Method들을 하나씩 정리했고 생각한 Method들을 어떤 Class에 넣을지 생각하고 이 전체적인 구상을 먼저 했다. 나는 총 3개의 Class를 만들었다. Board Class, Horse Class, TestDriver Class이다. Horse Class는 말에 대한 정보를 나타낸 Class이고, Board Class는 경기의 룰 그리고 판을 만드는 Method가 있으며 TestDriver Class에는 Main method가 있으며 이 Main method에서 Board객체를 만들어 경기를 실행하고 있다.

2. 분석

각 Class와 각 Class에 들어있는 Method에 대한 설명을 하겠다.

Board Class

Board Class

Cho : Horse

Han : Horse

boardSIze : int

Instance 변수부분에 Horse 객체 2개와 boardsize를 선언했다.

Public board() : 말을 생성한 method, Construct 부분이다.

Public makeBoard() : 입력 받은 숫자로 board판을 만든다.

Public setIintialHorsePosition() : 말이 처음에 시작하는 위치를 입력받는다.

Public inputFromUserBooardSize() : boardsize를 입력 받는다. 4/5/6일 경우만 입력 가능

Public setIintialHorsePosition2() : 말이 처음에 시작하는 위치를 입력받는다.

Public GameStart() : 실제로 경기를 하는 Method이다. 이 Method는 입력 받은 2개의 객체를 이용해 현재 위치를 출력하고 현재 위치에서 boardsize를 넘지 않고 움직일 수 있는 경우의 수를 모두 출력한다. 그 후에 그 움직일 수 있는 경우의 수에서 랜덤 함수를 이용해 랜덤으로 아무 위치나 이동하게 한다. 만약 내가 움직일 수 있는 위치에 상대의 말이 있을 경우 게임을 종료한다.

Horse Class

Horse Class

x : private int

y : private int

Public GameEnd() : 내가 현재 움직일 수 있는 위치에 상대의 말이 있는지 확인 하는 Method이 이다. 실제로 내가 움직일 수 있는 위치에 상대의 말이 있으면 false를 반환해 경기를 종료한다.

Public RealityMove() : 말이 실제로 움직이게 하는 method이다. 움직이는 것은 랜덤으로 돌리기 위해 랜덤 함수를 사용하고 만약 isValid() method가 true일 경우에 실제로 setX(), setY()를 이용해 실제로 움직이도록 설계했다.

Public CurrentMove() : 말이 현재 위치에서 boardSize 판을 벗어나지 않고 움직일 수 있는 모든 경우의 수를 나타내는 Method이다. 하나의 말이 움직일 수 있는 총 가지 수가 8개이다. 그래서 반복문을 통해 8번을 반복해 이 8개의 경우의 수에서 boardsize를 벗어나지 않게 하는 모든 경우의 수를 출력한다.

Public setX(), setY() : 말의 x위치와 y위치를 변경시키려고 할 때 사용한다.

Public getX(),getY() : 말의 x위치와 y위치를 나타낸다.

Public isValid() : 말이 움직이는 위치가 x좌표와 y좌표를 두 좌표가 boardsize를 넘는 지를 검사하는 method로 넘으면 false를 반환하고 그 범위 내에서 움직이면 true를 반환한다.

Instance 변수에 말의 x 좌표와 y좌표를 선언했다.

Public Horse() : 말의 위치를 나타내고 위치를 this.을 사용해 초기화 했다., Construct 부분이다.

Public current() : 말의 현재 위치를 나타낼 때 사용한다.

TestDriver Class

TestDriver Class

Board : Board

MainMethod() : Board객체를 만들어 실제로 경기를 진행한다. 말도 만들고 , board도 실제로 입력 받아 경기를 진행한다. 게임 재시작을 위한 while 문 하나 존재한다.

GameStart()의 flowchart

종료

Cho.RealityMove() 실행

false

Han.GameEnd()

Han.CurrentMove() 실행

Cho.Current() 실행

Cho.RealityMove() 실행

true

false

Cho.GameEnd()

Cho.CurrentMove() 실행

생성된 2개의 Horse 객체

While 문 시작

종료

False 반환

Han.current() 실행

true

False 반환

CurrentMove() flowchart

For 문으로 8번 i=0

isValid() 실행

false

출력. i ++

true

출력 x, i++

8번 실행 후 종료

RealityMove() flowchart

setX(), setY() 실행

true

다시 루프

false

isValid() 실행

While Randommove = 1

Random함수로 8개

GameEnd() flowchart

다시 루프

다시 루프

false

isValid() 실행

For문 8번 i = 0

CurrentMove == 상대 팀 말 위치 확인

False 반환

3개의 Class가 존재하며 실질적으로 경기를 하는 Class는 Board Class이다. Board Class에서 Horse Class의 객체 2개의 만든 다음에 Horse Class의 Method를 이용해 경기를 진행하는 Method를 실행한다. TestDriver Class에서는 Main Method가 있고 이 Main Method 에서 Board 객체를 만들어 경기를 Board 객체가 진행하도록 만들었다.

TestDriver Class

Board Class

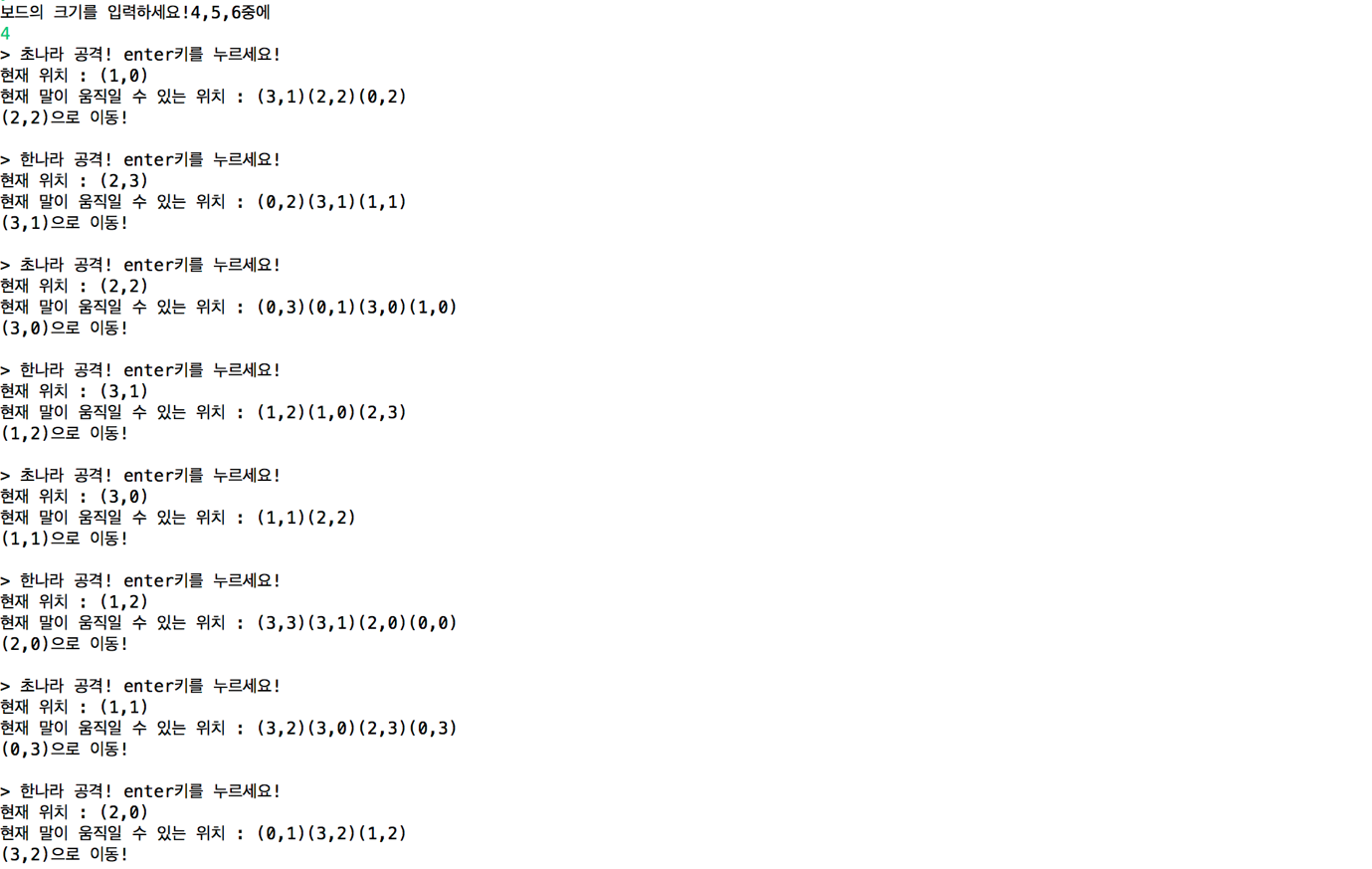
Horse Class

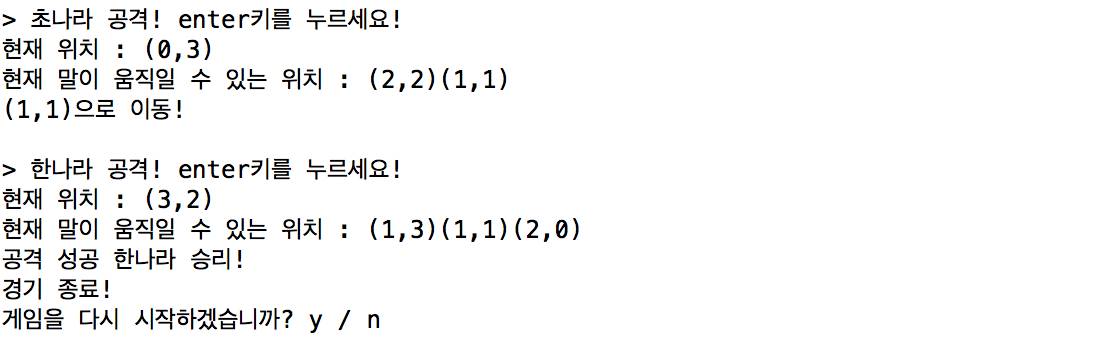
Main Method 에서 board 객체 만들고 경기 진행

Horse 객체를 만들고 경기 진행

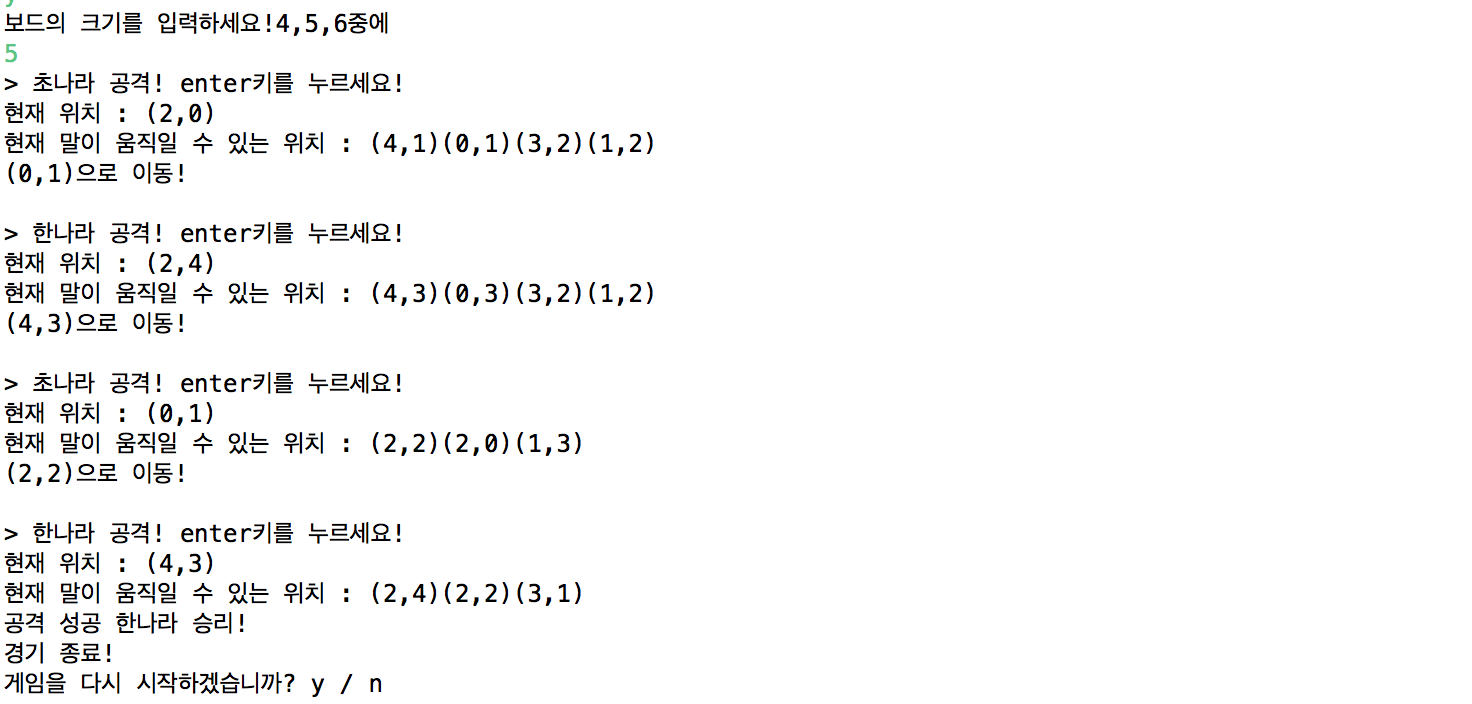
말에 대한 정보

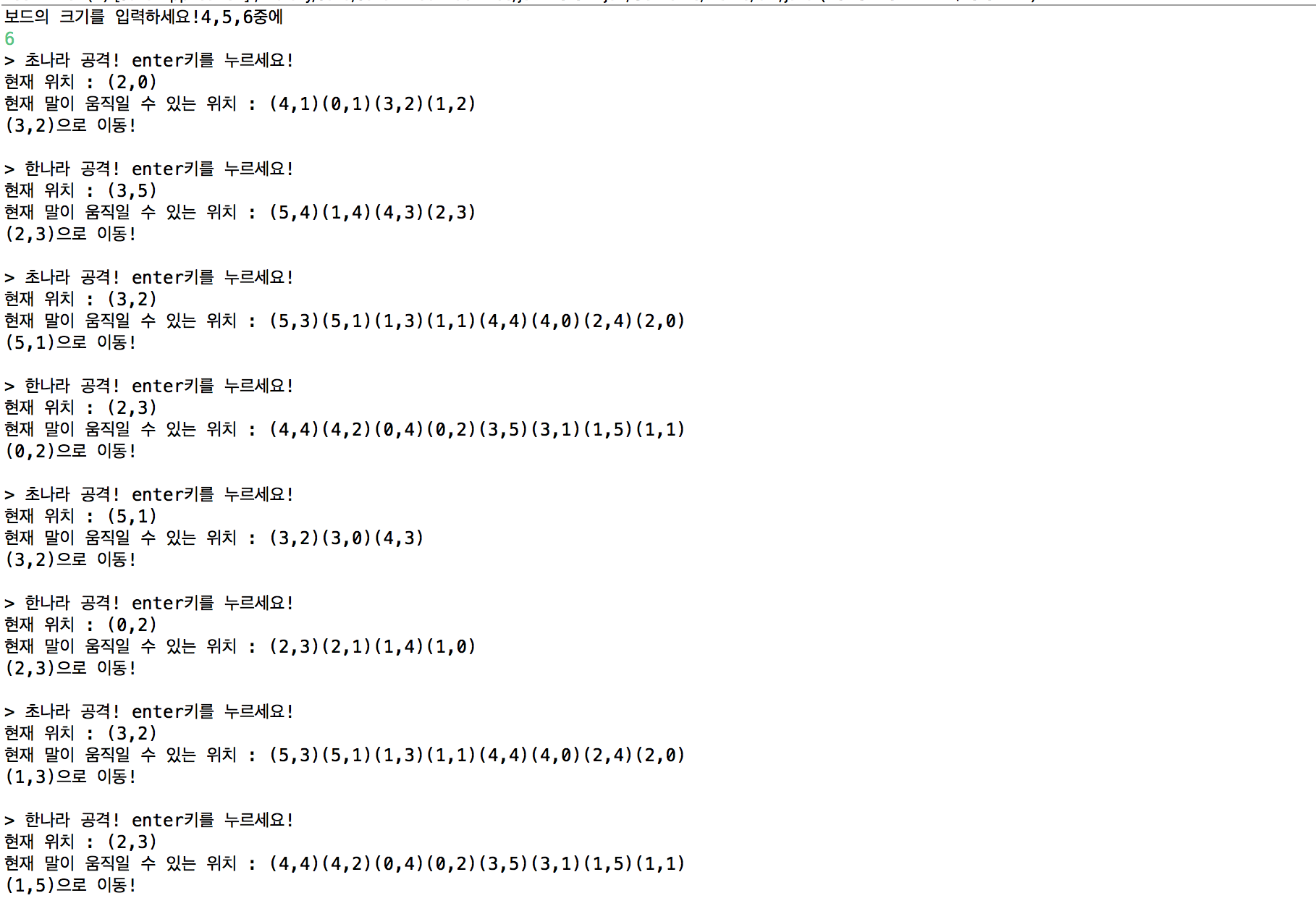
3. 실행 결과 화면

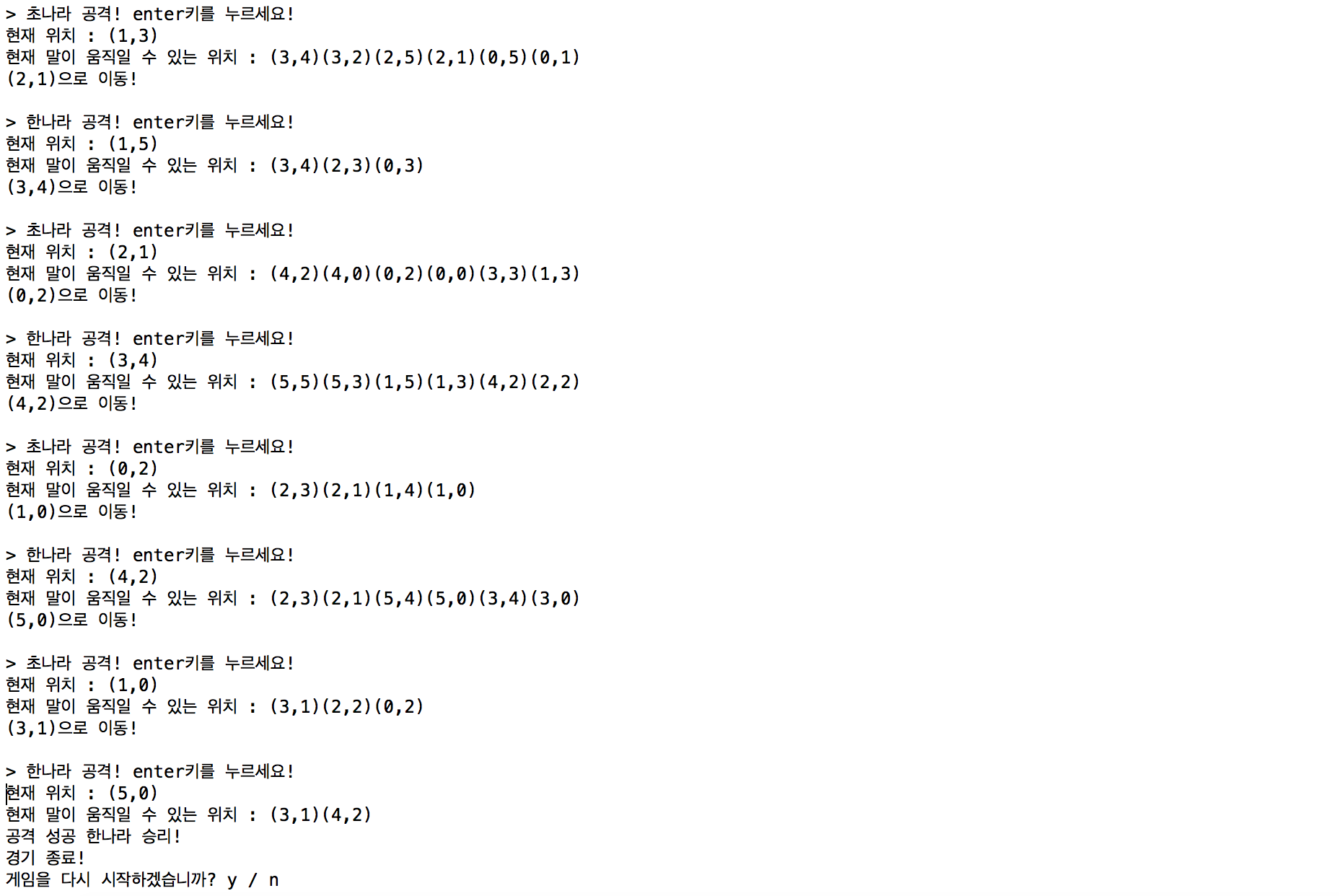




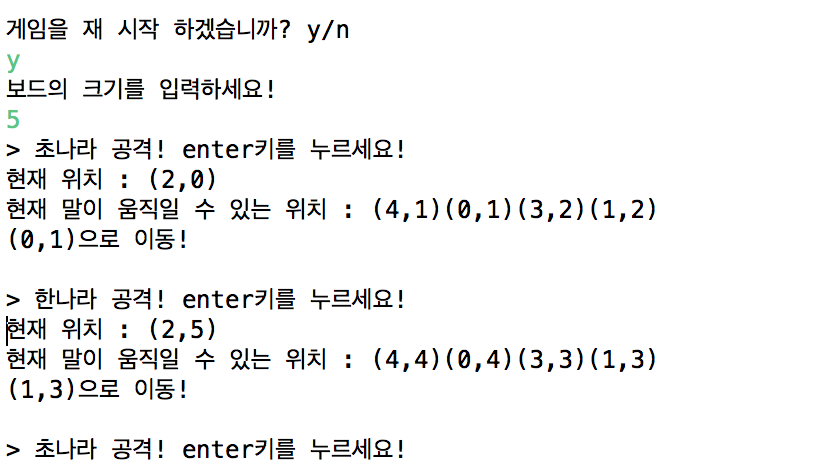
<Inputsize 가 4인 경우> //



<inputsize가 5인 경우>



<inputsize가 6인 경우>



<재 시작을 하는 경우>

4. 결론

미니 장기 게임을 구현하면서 많은 것을 배웠다. 실습 과제는 내가 배운 이론을 Eclipse를 통해 제대로 실행되는지 확인하는 것이였다. 이번 과제에서는 처음부터 과제에 대한 설명만 있고 그것을 처음부터 다 구현하려고 하니 막막했다. 제일 어려웠던 부분이 내가 어떤 Class를 만들고 그 Class가 어떤 역할을 하는지 그 Class에는 어떤 Method가 들어가야하는 지 이것을 처음부터 내 생각대로 만들어야 한다는 것이 힘들었다. 하지만 과제가 나오자마자 바로 시작했고, 처음부터 다 구현해야하는 과제는 시간에 부딪혀 하는 것 보다 많은 시간을 가지고 천천히 구현하면 더 좋은 결과를 만들어 낼 수 있다. 시간이 많아 수정을 자주 할 수 있어 좋은 코드를 만들어 낼 수 있다.