
<컴퓨터학 실험 I>

테트리스 프로젝트 2주차

목차

- 테트리스 프로젝트 2주차 목표 - 랭킹 시스템
- 테트리스 프로젝트 2주차 구현 결과
- 테트리스 프로젝트 2주차 구현 프로그램 및 Flow chart
 - 랭킹 정보를 위해 사용되는 structure
 - 구현내용
 - 랭킹 시스템 예제
- 테트리스 프로젝트 2주차 실습 평가
- 테트리스 프로젝트 2주차 숙제
- 테트리스 프로젝트 2주차 결과보고서
- 테트리스 프로젝트 3주차 예비보고서

테트리스 프로젝트 2주차 목표

□ 랭킹 시스템(Ranking System)

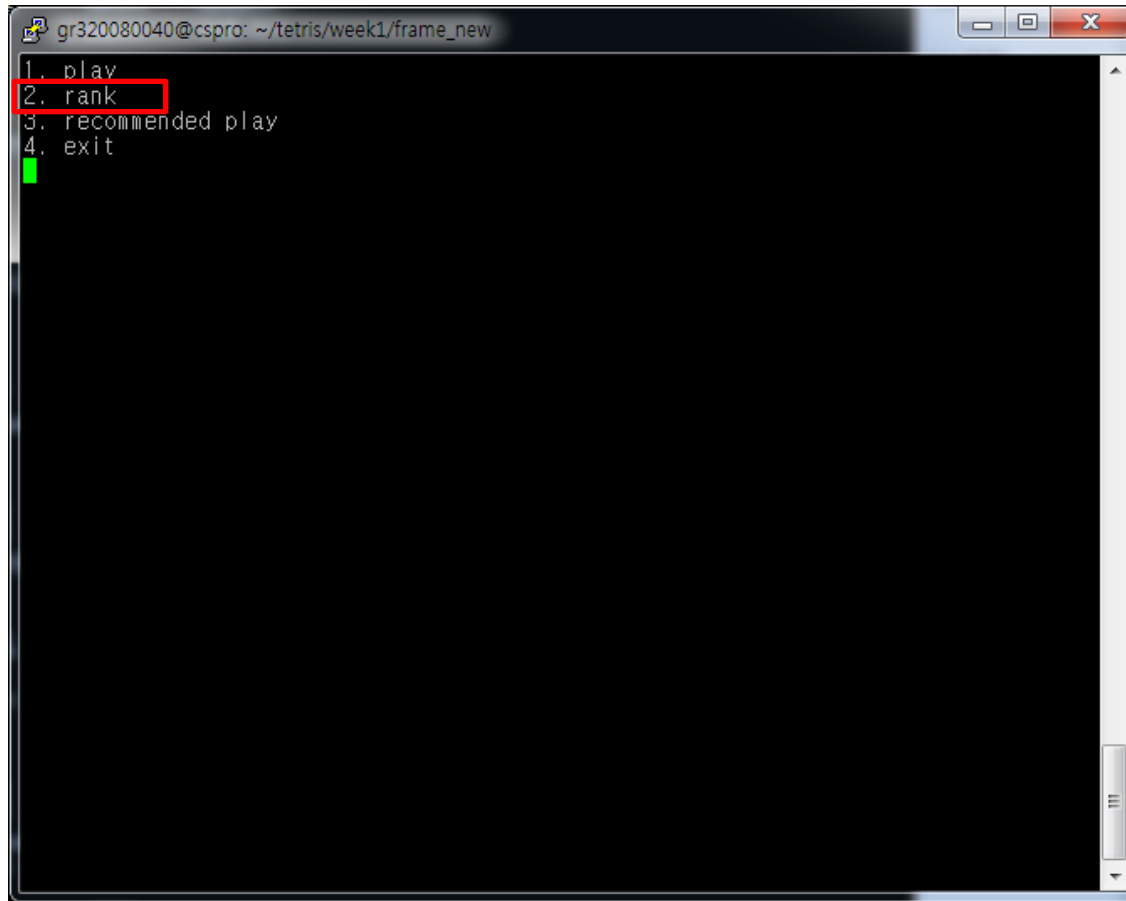
- 테트리스 게임을 play한 후(game over시), 사용자 이름을 입력받고, 사용자 이름과 점수로 구성되는 랭킹 정보를 등록하고 확인할 수 있는 **랭킹 시스템**을 구현한다.
- 사용자 이름과 점수로 구성된 랭킹 정보들은 다음 번에 테트리스 게임을 실행할 때도 유지하기 위해 rank.txt 파일에 기록된다.
- 테트리스 게임을 실행할 때, rank.txt를 읽어 들여, **자료구조를 구축**하고 랭킹 정보를 프로그램 수행 중에도 유지한다.
- 테트리스 게임이 종료(game over)되면, 사용자 이름을 입력 받고 랭킹 정보(사용자 이름과 점수)를 랭킹 정보를 저장하고 있는 자료구조에 추가한다.

- 입력파일 예제(입력 파일 양식은 학생들마다 달라질 수 있음)
 - rank.txt



테트리스 프로젝트 2주차 실습 구현결과(2/12)

- 랭킹을 확인하기 위해 2를 입력



```
gr320080040@cspro: ~/tetris/week1/frame_new
1. play
2. rank
3. recommended play
4. exit
█
```

A terminal window titled 'gr320080040@cspro: ~/tetris/week1/frame_new' displays a menu with four options: '1. play', '2. rank', '3. recommended play', and '4. exit'. The option '2. rank' is highlighted with a red rectangular box. A green cursor is positioned on the line following the menu.

테트리스 프로젝트 2주차 실습 구현결과(3/12)

□ 출력1

- 1위~5위까지 랭킹을 확인하기 위해 1과 5를 입력하고 점수 순으로 1위부터 5위까지 랭킹 정보를 확인.

```
gr320080040@csp: ~/tetris/week3/week3_hw
1. list ranks from X to Y
2. list ranks by a specific name
3. delete a specific rank
X: 1
Y: 5
name | score
-----|-----
Jacob | 20340
Park  | 17990
Kim   | 7430
Jacob | 620
Chris | 450
```

주어진 메뉴(1, 2, 3)에 대해 키보드로부터 입력 받아 선택된 메뉴가 화면에 출력되지 않고, 바로 두 개의 랭킹(X, Y)를 입력 받는 문구가 출력되는 이유는 메뉴 선택 시 wgetch() 함수를 사용하기 때문에 scanf() 함수를 사용했을 때와 달리 화면에 그 결과가 출력되지 않는다.
(이 슬라이드 이후 메뉴 선택하는 모든 출력 예제에서 이와 같은 현상을 관찰할 수 있다)

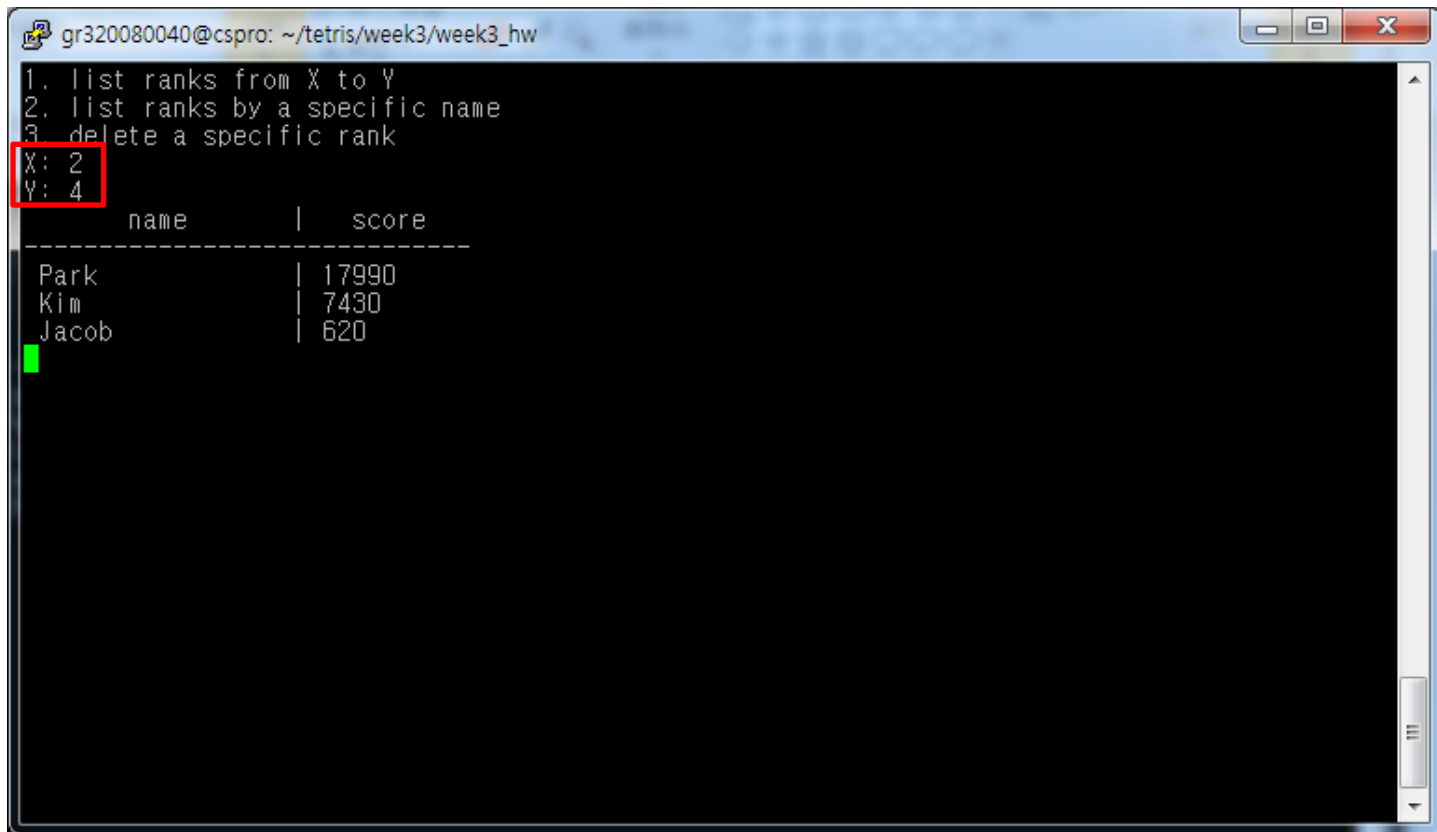
사용자 이름

점수

테트리스 프로젝트 2주차 실습 구현결과(4/12)

□ 출력2

- 2위~4위까지 랭킹을 확인하기 위해 2와 4를 입력하고 점수 순으로 2위부터 4위까지 랭킹 정보를 확인.



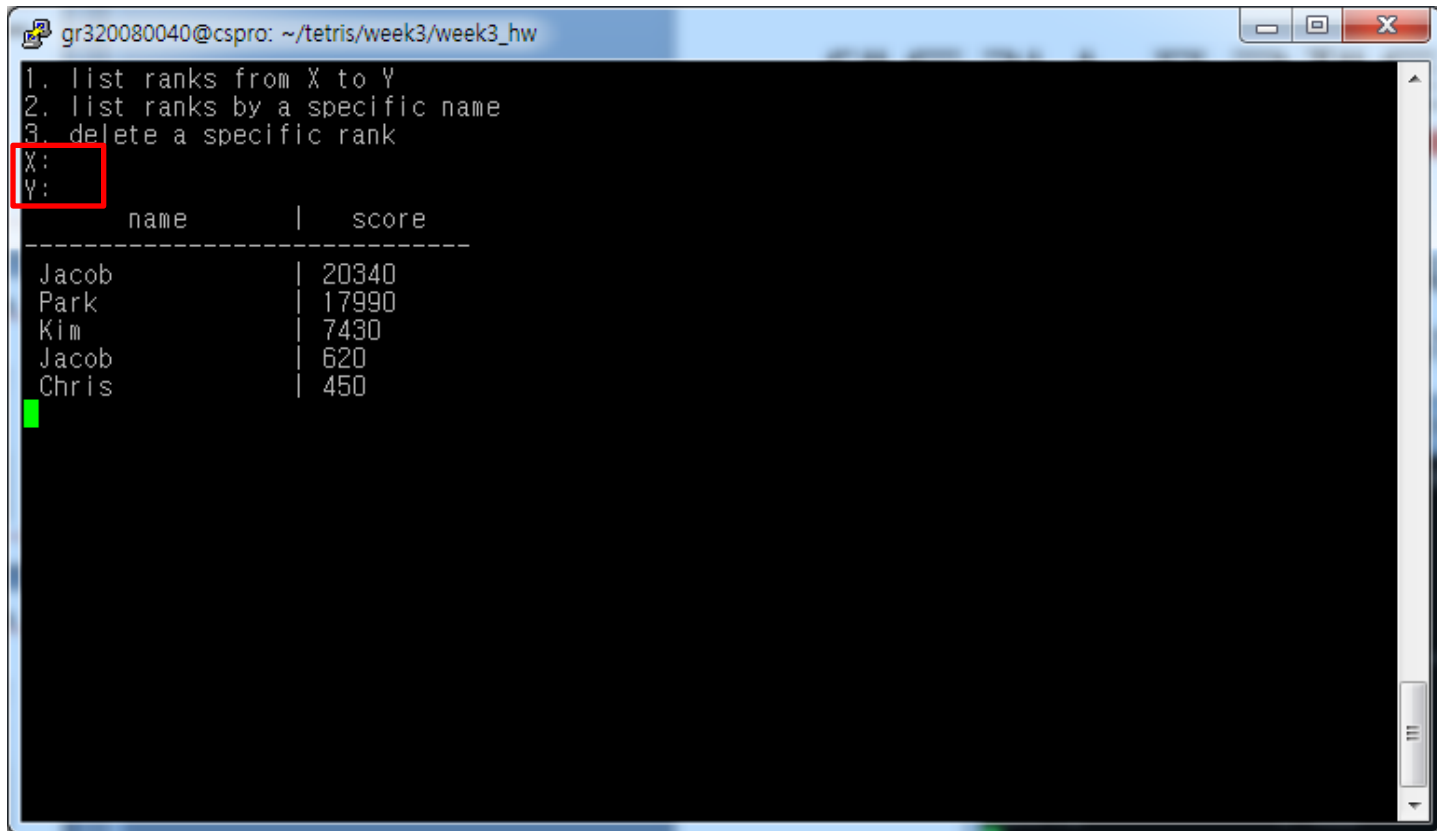
```
gr320080040@cspiro: ~/tetris/week3/week3_hw
1. list ranks from X to Y
2. list ranks by a specific name
3. delete a specific rank
X: 2
Y: 4
-----
name      | score
-----
Park      | 17990
Kim       | 7430
Jacob     | 620

```

테트리스 프로젝트 2주차 실습 구현결과(5/12)

□ 출력3

- X와 Y 모두 입력하지 않았을 경우.
- 점수 순으로 1위~5위까지 모든 순위 출력

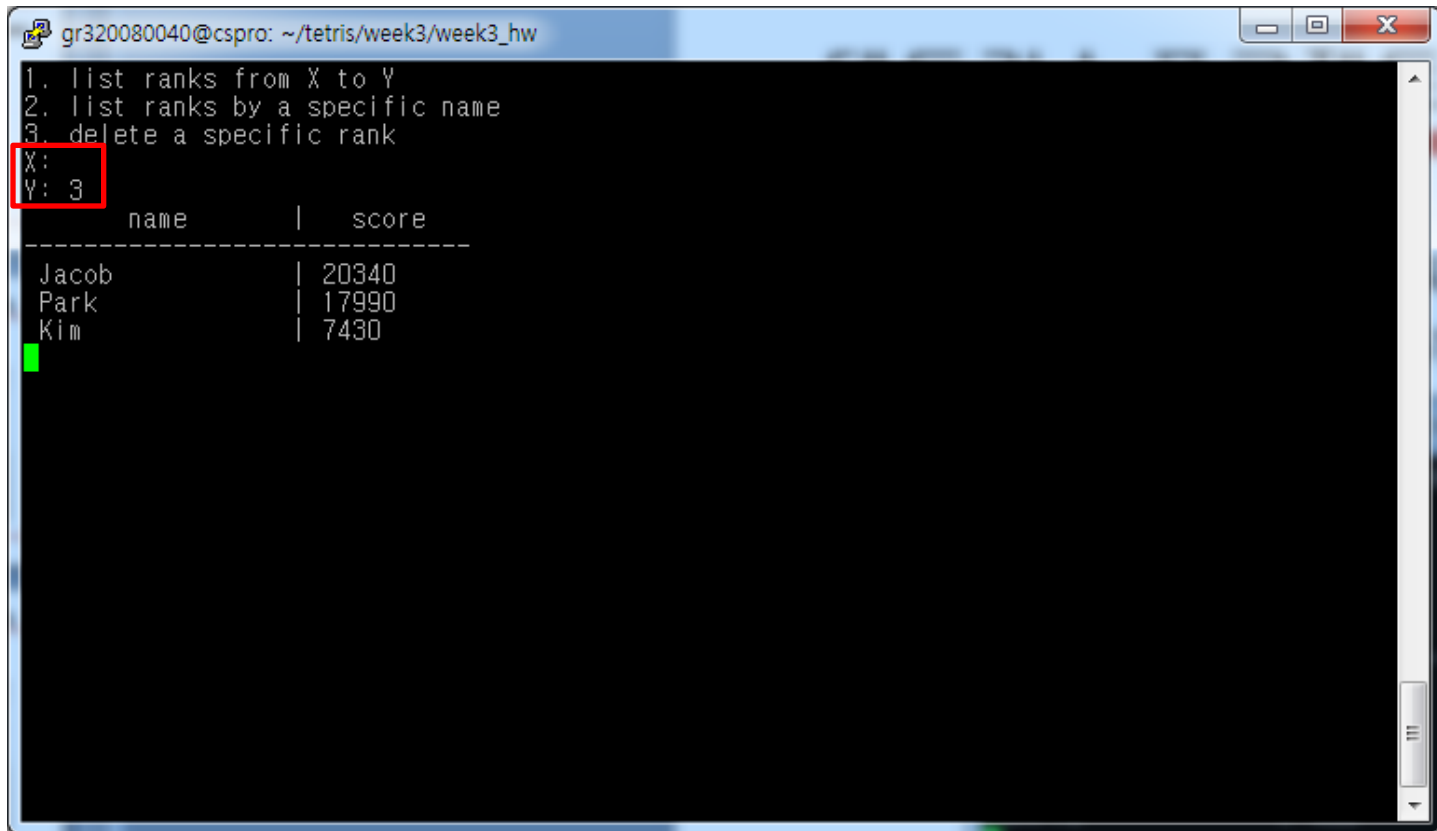


```
gr320080040@cspiro: ~/tetris/week3/week3_hw
1. list ranks from X to Y
2. list ranks by a specific name
3. delete a specific rank
X:
Y:
name | score
-----
Jacob | 20340
Park  | 17990
Kim   | 7430
Jacob | 620
Chris | 450
█
```


테트리스 프로젝트 2주차 실습 구현결과(6/12)

□ 출력4

- X를 입력 안 하고, Y로 3을 입력하는 경우.
- 점수 순으로 1위~3위까지 출력

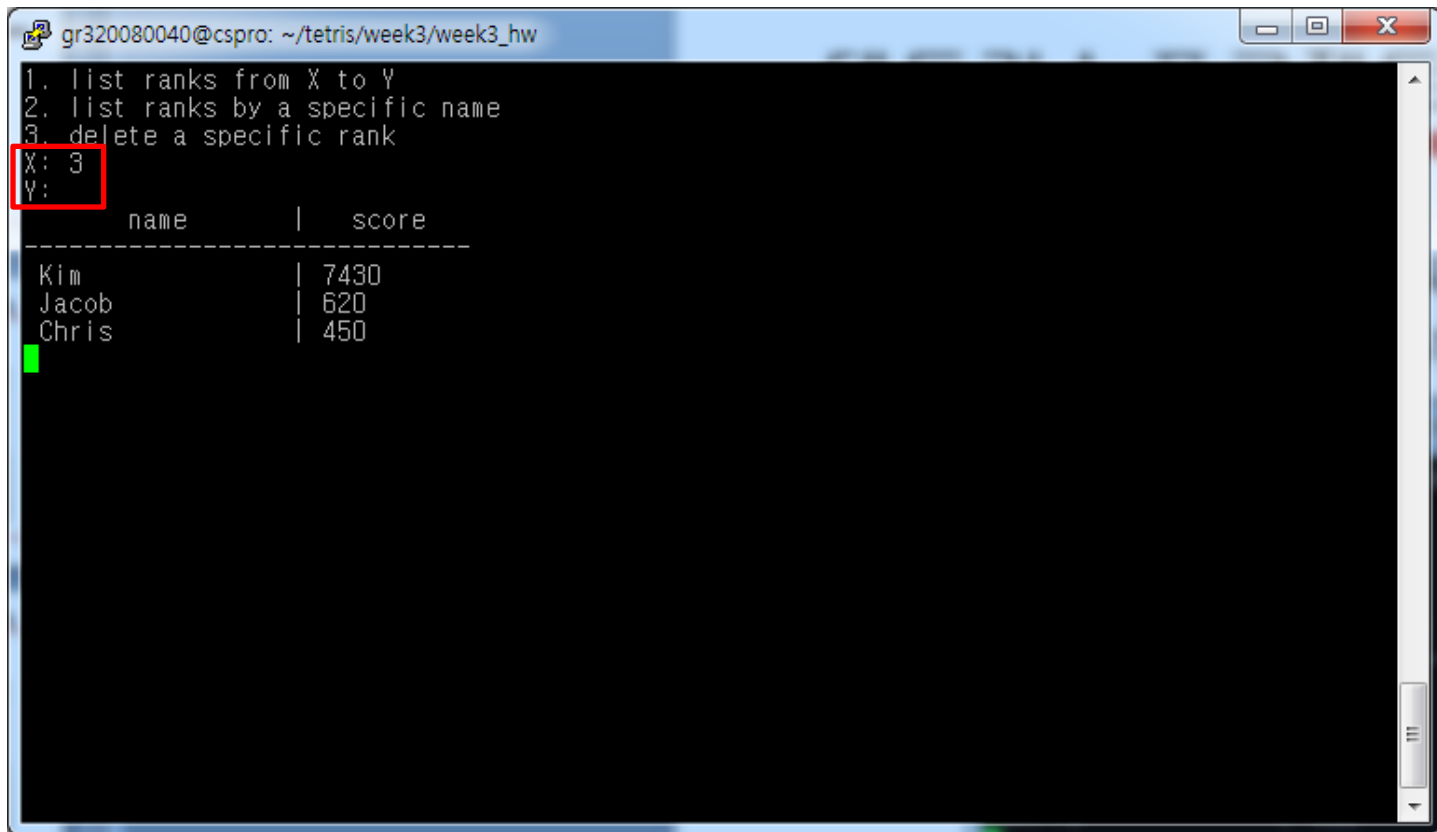


```
gr320080040@cspiro: ~/tetris/week3/week3_hw
1. list ranks from X to Y
2. list ranks by a specific name
3. delete a specific rank
X:
Y: 3
name | score
-----
Jacob | 20340
Park  | 17990
Kim   | 7430
```

테트리스 프로젝트 2주차 실습 구현결과(7/12)

□ 출력5

- X로 3을 입력하고, Y를 입력 안 하는 경우.
- 점수 순으로 3위~5위까지 출력

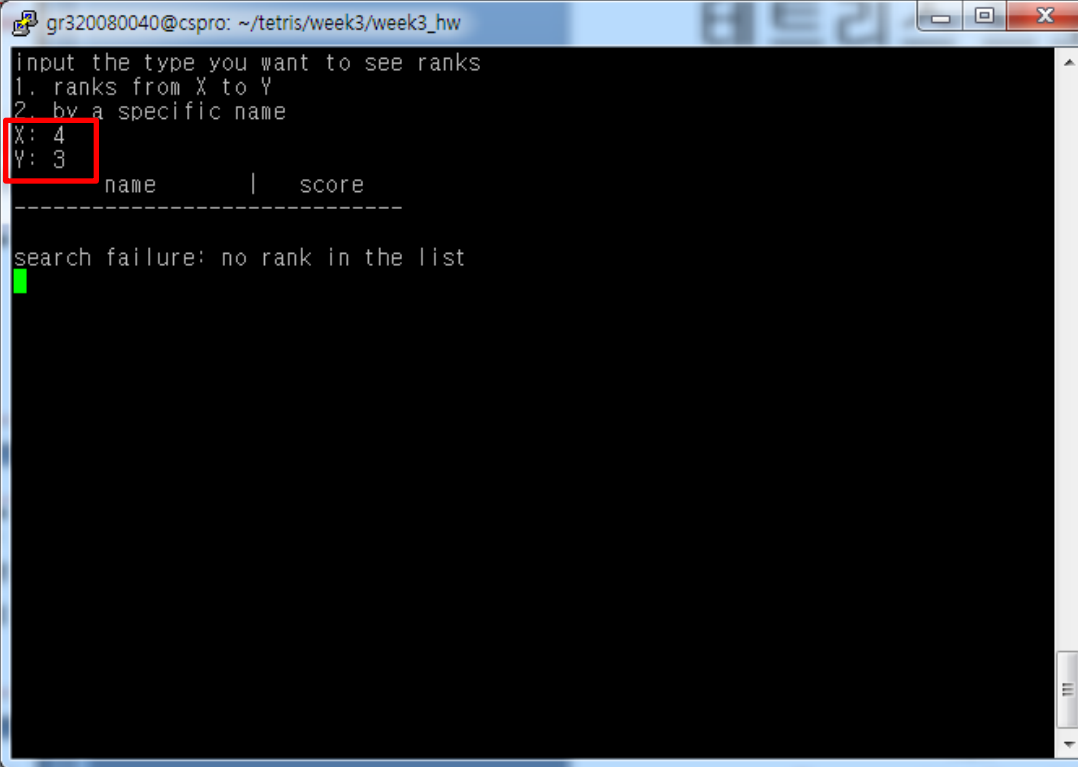


```
gr320080040@csp: ~/tetris/week3/week3_hw
1. list ranks from X to Y
2. list ranks by a specific name
3. delete a specific rank
X: 3
Y:
name | score
-----
Kim  | 7430
Jacob | 620
Chris | 450
█
```

테트리스 프로젝트 2주차 실습 구현결과(8/12)

□ 출력6

- X로 4을 입력하고, Y로 3을 입력 하는 경우.
- 어떤 랭킹도 출력하지 않고, “search failure: no rank in the list”라는 메시지 출력.

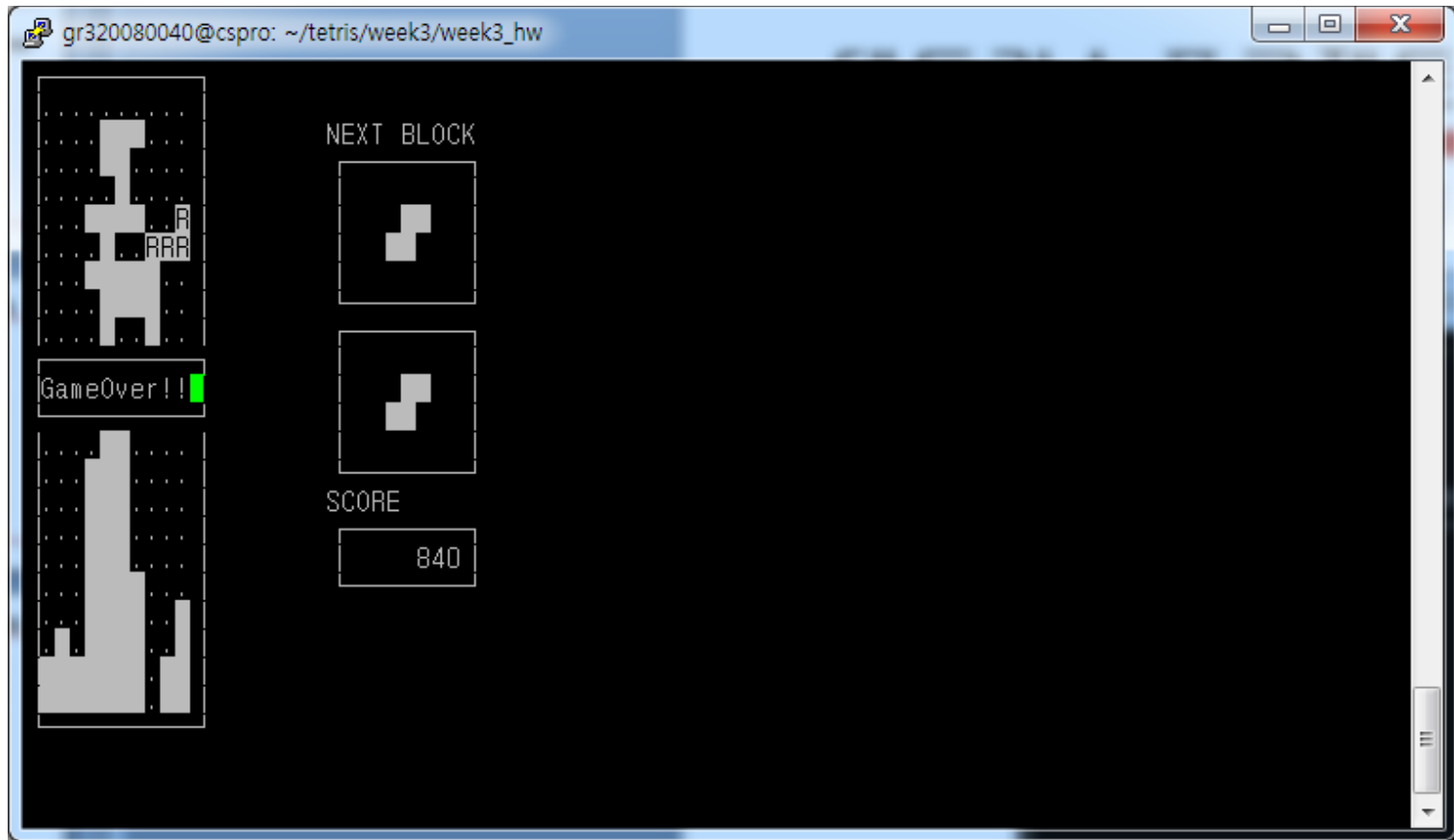


```
gr320080040@csp: ~/tetris/week3/week3_hw
input the type you want to see ranks
1. ranks from X to Y
2. by a specific name
X: 4
Y: 3
-----
name      |  score
-----
search failure: no rank in the list
```

The image shows a terminal window with a black background and white text. The window title is 'gr320080040@csp: ~/tetris/week3/week3_hw'. The prompt is 'input the type you want to see ranks'. There are two options: '1. ranks from X to Y' and '2. by a specific name'. The user has entered 'X: 4' and 'Y: 3', which are highlighted with a red box. Below this, there is a header for a table: '-----' followed by 'name | score' and another '-----'. The output is 'search failure: no rank in the list'.

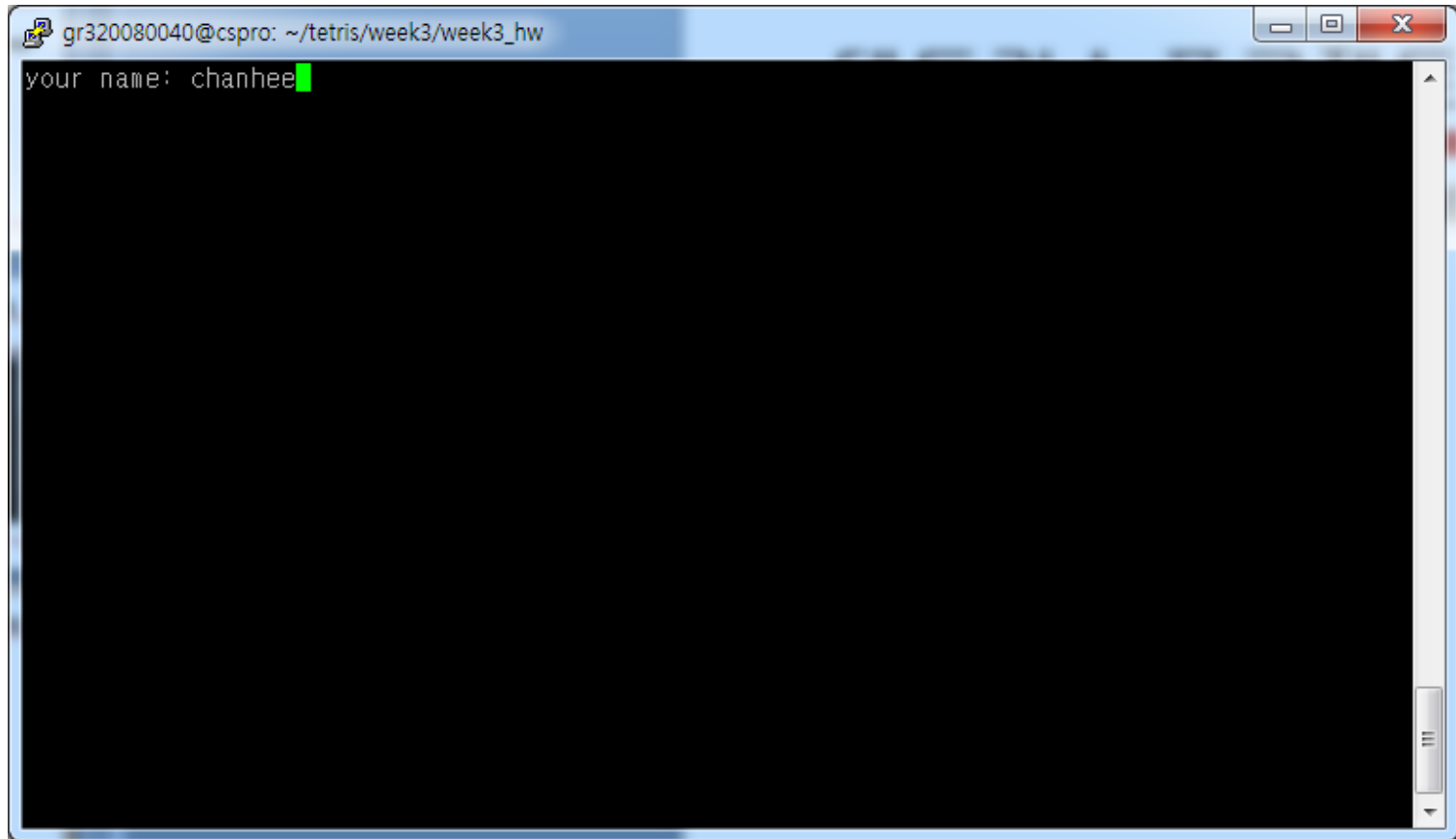
테트리스 프로젝트 2주차 실습 구현결과(9/12)

- 새로운 랭킹 정보를 등록하기 위해 게임 종료



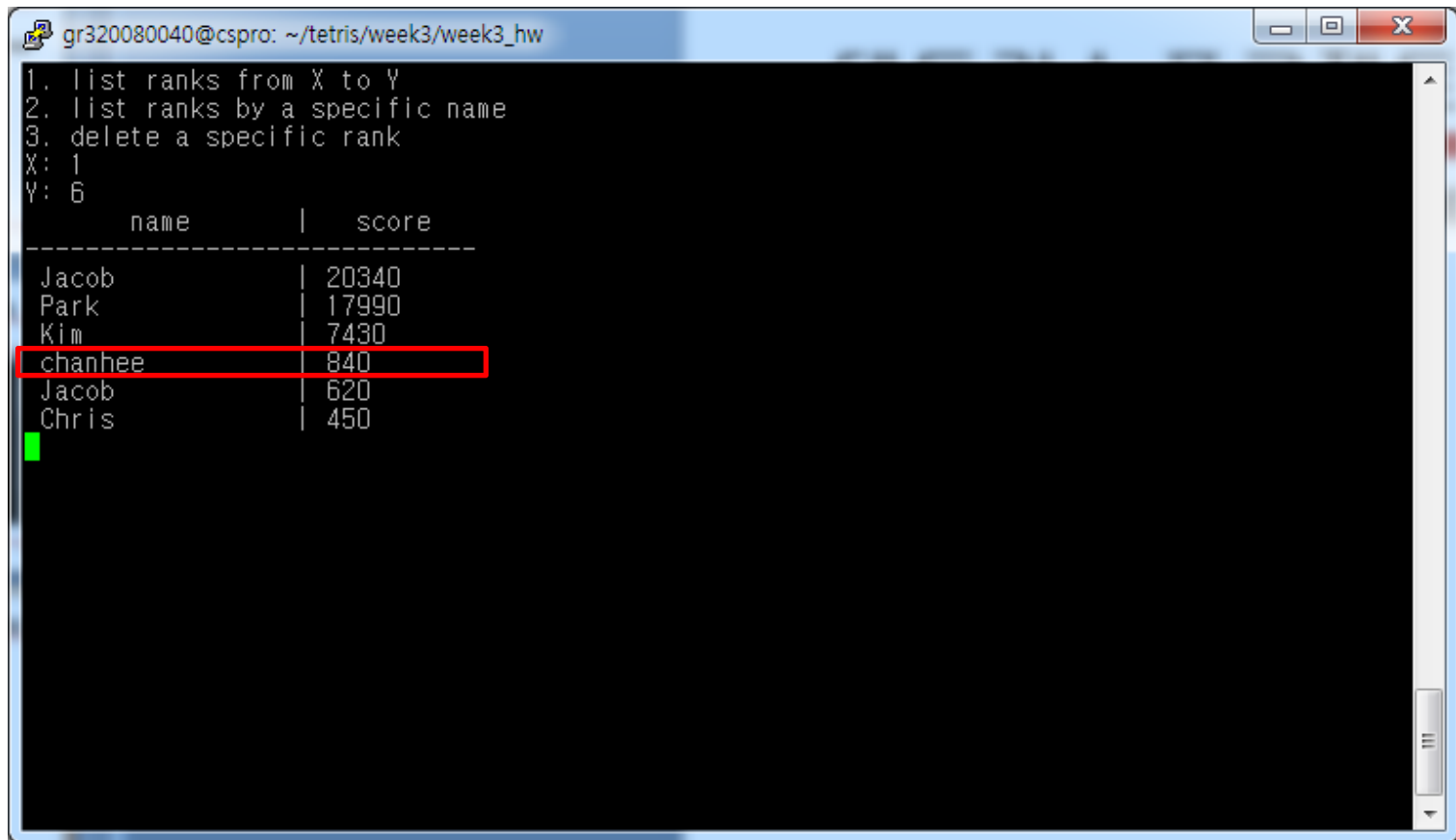
테트리스 프로젝트 2주차 실습 구현결과(10/12)

- 사용자 이름 입력(예제에서는 “chanhee”)



테트리스 프로젝트 2주차 실습 구현결과(11/12)

- chanhee의 랭킹 정보를 확인



```
gr320080040@csp: ~/tetris/week3/week3_hw
1. list ranks from X to Y
2. list ranks by a specific name
3. delete a specific rank
X: 1
Y: 6
  name | score
-----|-----
Jacob  | 20340
Park   | 17990
Kim    | 7430
chanhee| 840
Jacob  | 620
Chris  | 450
```

테트리스 프로젝트 2주차 실습 구현결과(12/12)

- Input 파일 rank.txt에 기록된 chanhee의 랭킹 정보 확인

A screenshot of a terminal window titled "gr320080040@cspiro: ~/tetris/week3/week3_hw". The terminal displays the contents of a file named "rank.txt":

```
Jacob 20340  
Park 17990  
Kim 7430  
chanhee 840  
Jacob 620  
Chris 450  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```


The entry "chanhee 840" is highlighted with a red rectangular box. At the bottom left of the terminal, it shows the command being executed: "`rank.txt`" 7L, 66C. On the bottom right, there are status indicators "1,1" and Korean text "모두" (All).

테트리스 프로젝트 2주차

구현 프로그램 설명, Flow Chart 및 함수표

Function() : 구현할 함수

랭킹 시스템 예제(1/7)

□ 가정

- 가정1: 랭킹 시스템을 구현하기 위한 자료구조로 linked list를 사용.
- 가정2: rank.txt에는 빈 파일이다.

□ 테트리스 게임 play 전.

Empty Head node

□ 1번째 테트리스 게임 play 이후, 사용자 cs2011이 1500점 획득하고 game over.

- 사용자 cs2011, score 1500을 갖는 head node가 생성된다.

name: cs2011
Score: 1500

랭킹 시스템 예제(2/7)

- 2번째 테트리스 게임 play 이후, 사용자 cs2012가 900점 획득하고 game over.
 - Link를 따라가면서, score를 비교하여, 새로운 node를 삽입할 적절한 위치를 탐색한다.

name: cs2011
Score: 1500



이 위치에 새로운 node를 삽입

- 사용자가 cs2012, score로 900을 갖는 새로운 node 삽입한다. 그 결과로 정렬된 linked list가 생성된다.

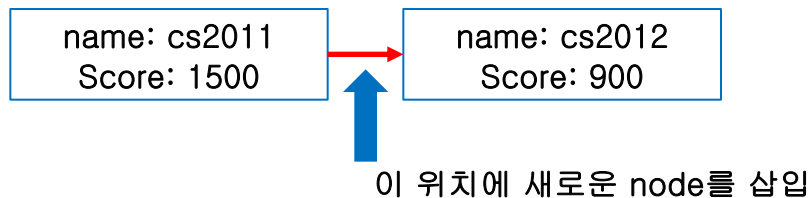
name: cs2011
Score: 1500



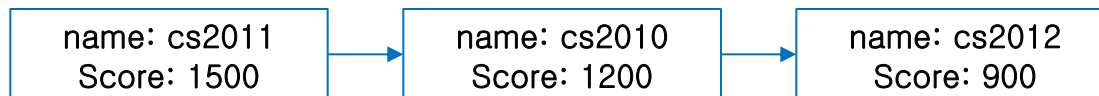
name: cs2012
Score: 900

랭킹 시스템 예제(3/7)

- 3번째 테트리스 게임 play 이후, 사용자 cs2010이 1200점 획득하고 game over.
- Link를 따라가면서, score를 비교하여, 새로운 node를 삽입할 적절한 위치를 탐색한다.



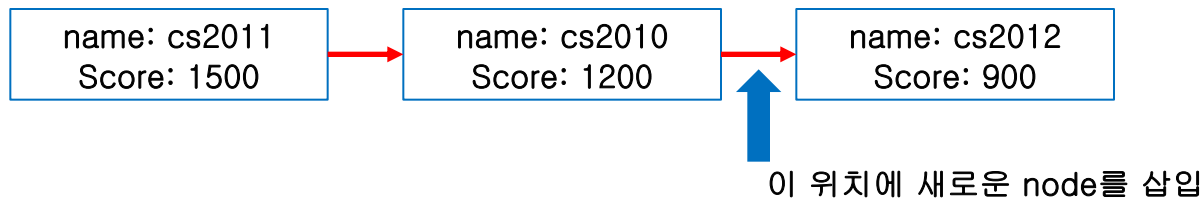
- 사용자가 cs2010, score로 1200을 갖는 새로운 node 삽입한다. 그 결과로 정렬된 linked list가 생성된다.



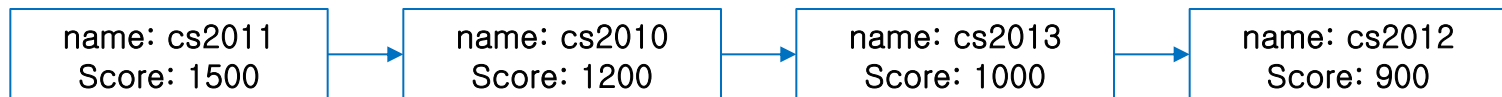
→ : 적절한 위치를 찾기 위해 탐색된 Link

랭킹 시스템 예제(4/7)

- 4번째 테트리스 게임 play 이후, 사용자 cs2013이 1000점 획득하고 game over.
 - Link를 따라가면서, score를 비교하여, 새로운 node를 삽입할 적절한 위치를 탐색한다.



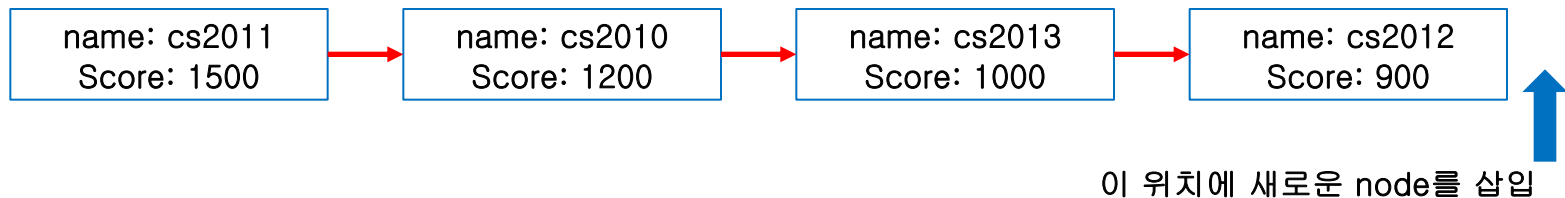
- 사용자가 cs2013, score로 1000을 갖는 새로운 node 삽입한다. 그 결과로 정렬된 linked list가 생성된다.



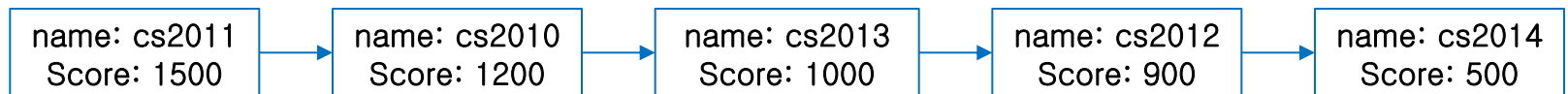
→ : 적절한 위치를 찾기 위해 탐색된 Link

랭킹 시스템 예제(5/7)

- 5번째 테트리스 게임 play 이후, 사용자 cs2014이 500점 획득하고 game over.
 - Link를 따라가면서, score를 비교하여, 새로운 node를 삽입할 적절한 위치를 탐색한다.



- 사용자가 cs2014, score로 500을 갖는 새로운 node 삽입한다. 그 결과로 정렬된 linked list가 생성된다.



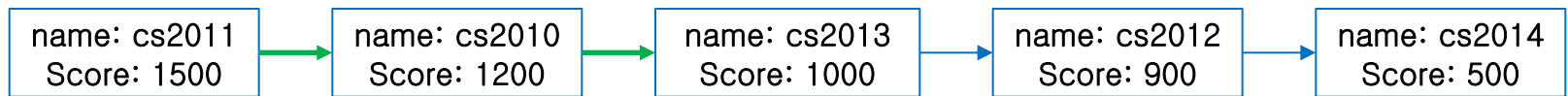
→ : 적절한 위치를 찾기 위해 탐색된 Link

랭킹 시스템 예제(6/7)

□ 랭킹 정보 출력 시 예제

- 랭킹 정보를 유지하기 위한 linked list는 정렬된 상태이므로 출력하길 원하는 랭킹의 범위(2개의 정수)를 입력 받았을 때, 헤드 노드(head node)부터 링크를 따라가면서 원하는 수 만큼의 정보를 출력하면 된다.

□ 예) 2와 3을 입력했을 때(2위~3위까지 순위를 알고 싶을 때)

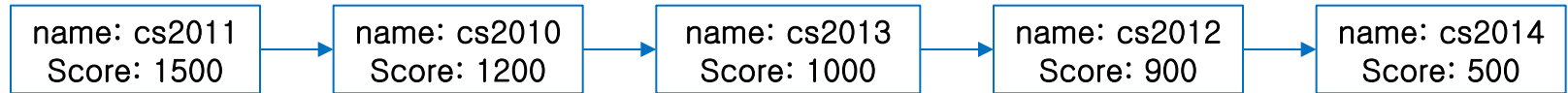


→ : 랭킹 정보를 출력하기 위해 탐색된 Link

탐색 후 두 개의 node 정보 출력

랭킹 시스템 예제(7/7)

- 랭킹 정보들을 저장하고 있는 linked list



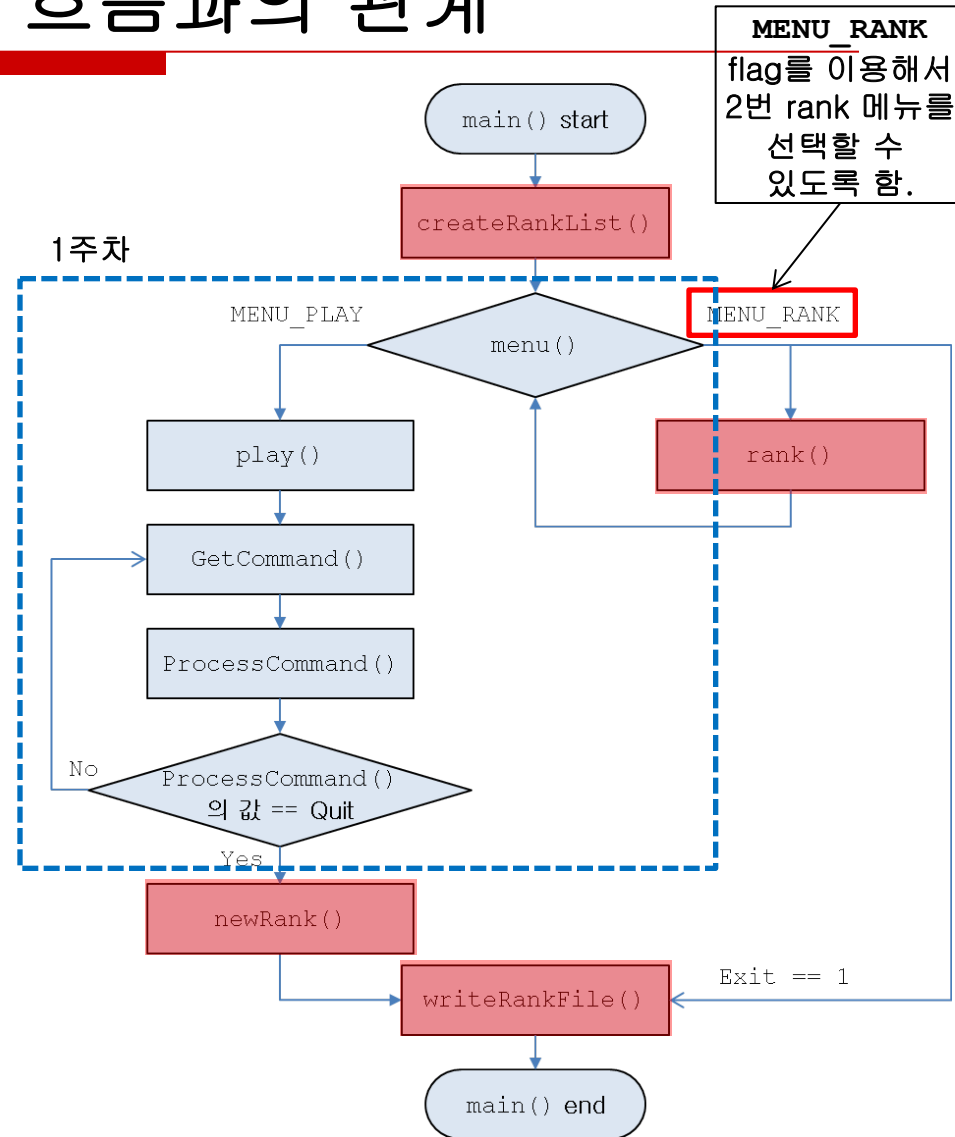
- rank.txt의 예
rank.txt

5		랭킹정보의 개수
cs2011	1500	
cs2010	1200	랭킹정보(사용자 이름, 점수)
cs2013	1000	
cs2012	900	
cs2014	500	

테트리스 프로그램 전체 흐름과의 관계

□ 키입력에 대한 동작 - 메뉴 2번

- **createRankList()**
 - 테트리스 프로그램 시작 시 rank.txt로부터 ranking 정보를 입력 받아 지정된 자료구조를 구축한다.
- **rank()**
 - 정수 2개, x, y($x \leq y$)를 입력 받고 랭킹 정보(x위~y위)를 화면에 출력한다.
- **newRank(int score)**
 - 게임 종료(gameover) 시, 사용자의 이름을 입력 받고, 사용자 이름과 score를 자료구조에 추가, 저장한다.
- **writeRankFile()**
 - 추가된 랭킹 정보가 있으면, 새로운 정보를 rank.txt에 기록하고, 추가된 정보가 없으면 그대로 종료한다.



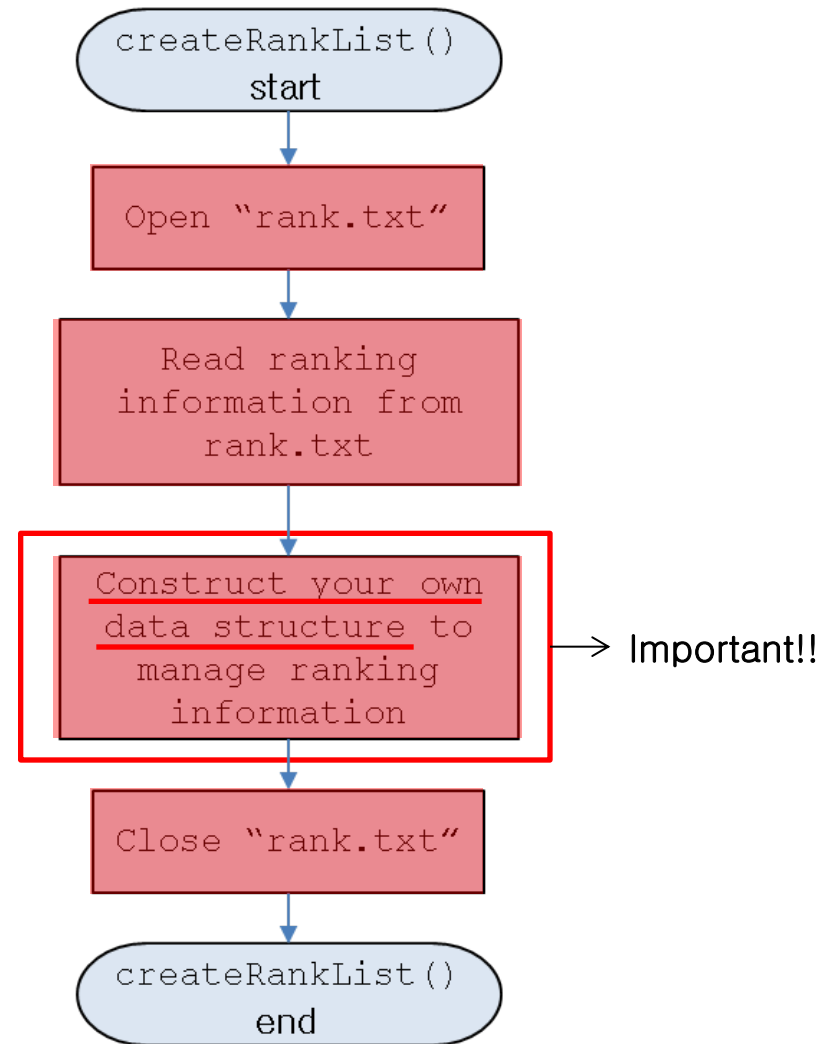
테트리스 프로젝트 2주차 구현 함수

- 테트리스 프로젝트 2주차 실습에서는 앞의 flow chart에서 명시된 4가지 함수들을 구현한다.
- 각 함수에서 대한 모든 flow chart는 예제에서 설명된 자료구조인 linked list를 바탕으로 한다.

2주차 구현 함수 - createRankList()

□ createRankList()

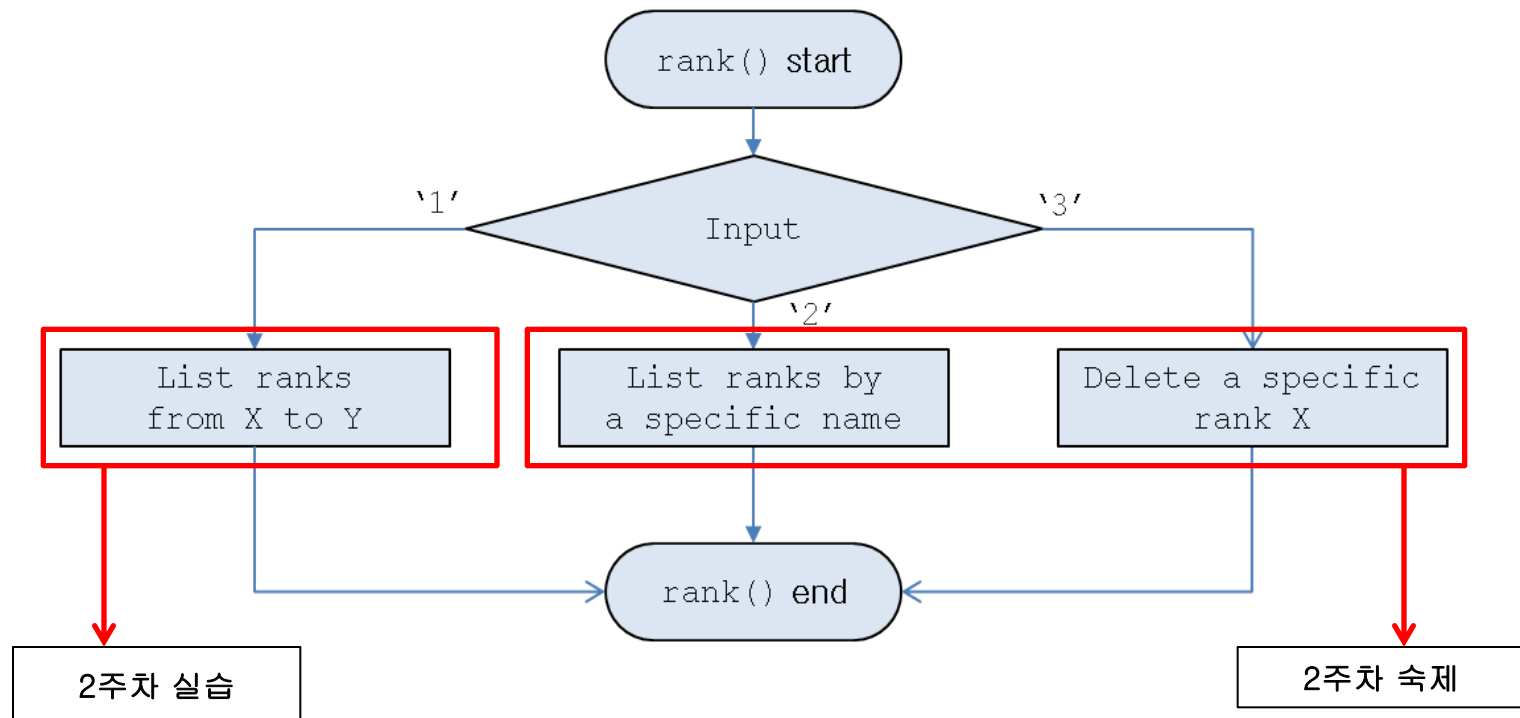
- Input 파일인 “rank.txt”에서 랭킹 정보를 읽어 들어 랭킹 정보를 저장하는 자료구조를 이용하여 랭킹 목록을 만든다.
- “rank.txt” 파일을 연다.
- “rank.txt”파일에서 랭킹 정보들을 읽어 들인다.
- 랭킹 정보들을 랭킹 정보를 저장 및 유지하기 위해 선택된 자료구조에 저장하면서 랭킹 목록을 만든다.
- “rank.txt”파일을 닫는다.



2주차 실습 구현 함수 - rank () (1/3)

□ 함수 rank ()

- 다음 flow chart에서 rank () 에서 수행되는 기본 기능들을 확인할 수 있다.

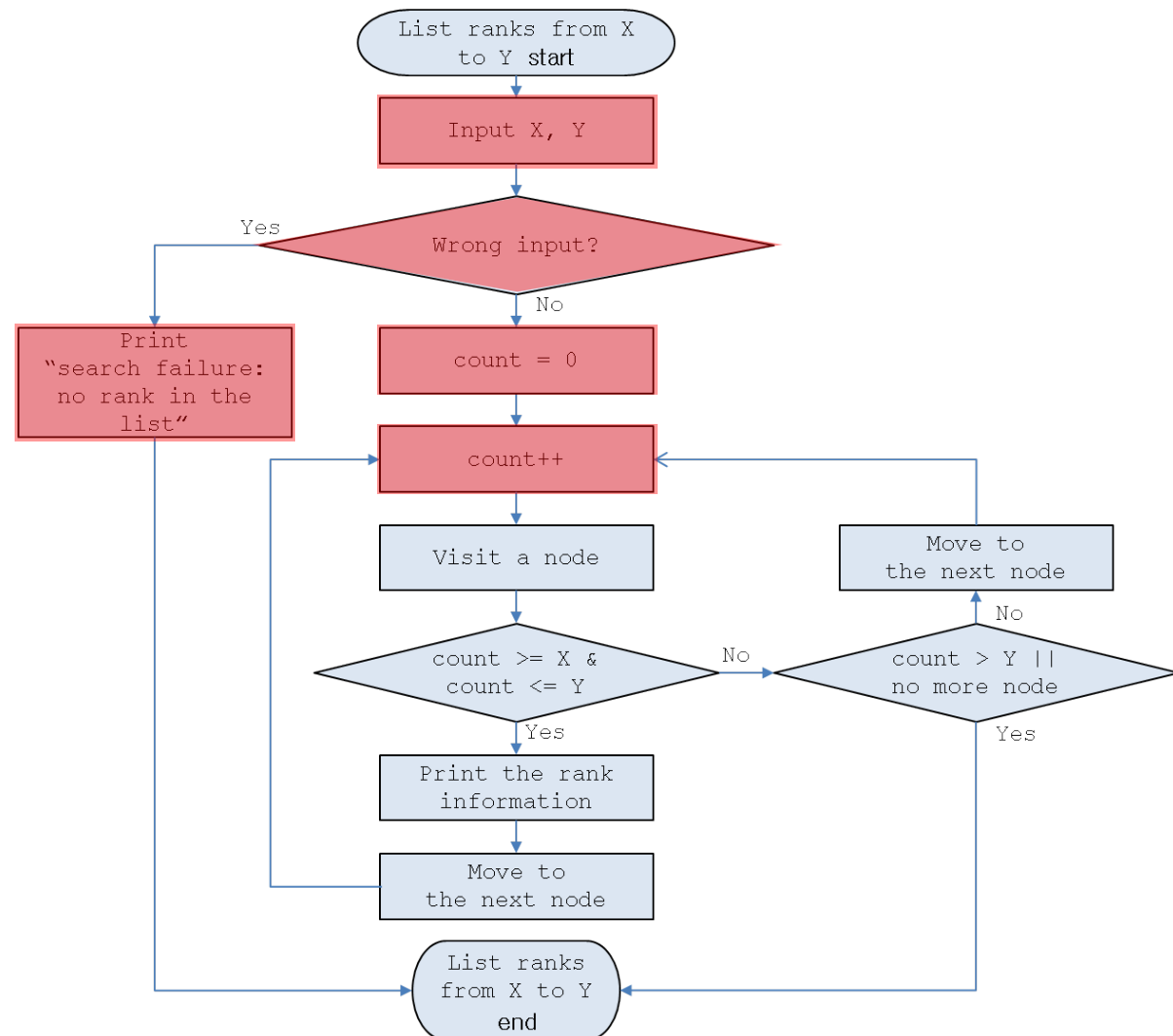


2주차 실습 구현 함수 - rank () (2/3)

□ List ranks

from X to Y

- 점수순으로 X~Y 위까지 출력하는 기능
- 정수 X, Y 를 입력 받는다.
- X, Y가 잘못된 input 인지 체크한다.
- 잘못된 input이라면, 메시지 “search failure: no rank in the list”를 출력한다.
- 변수 count를 0으로 초기화한다.
- 변수 count값을 증가한다.

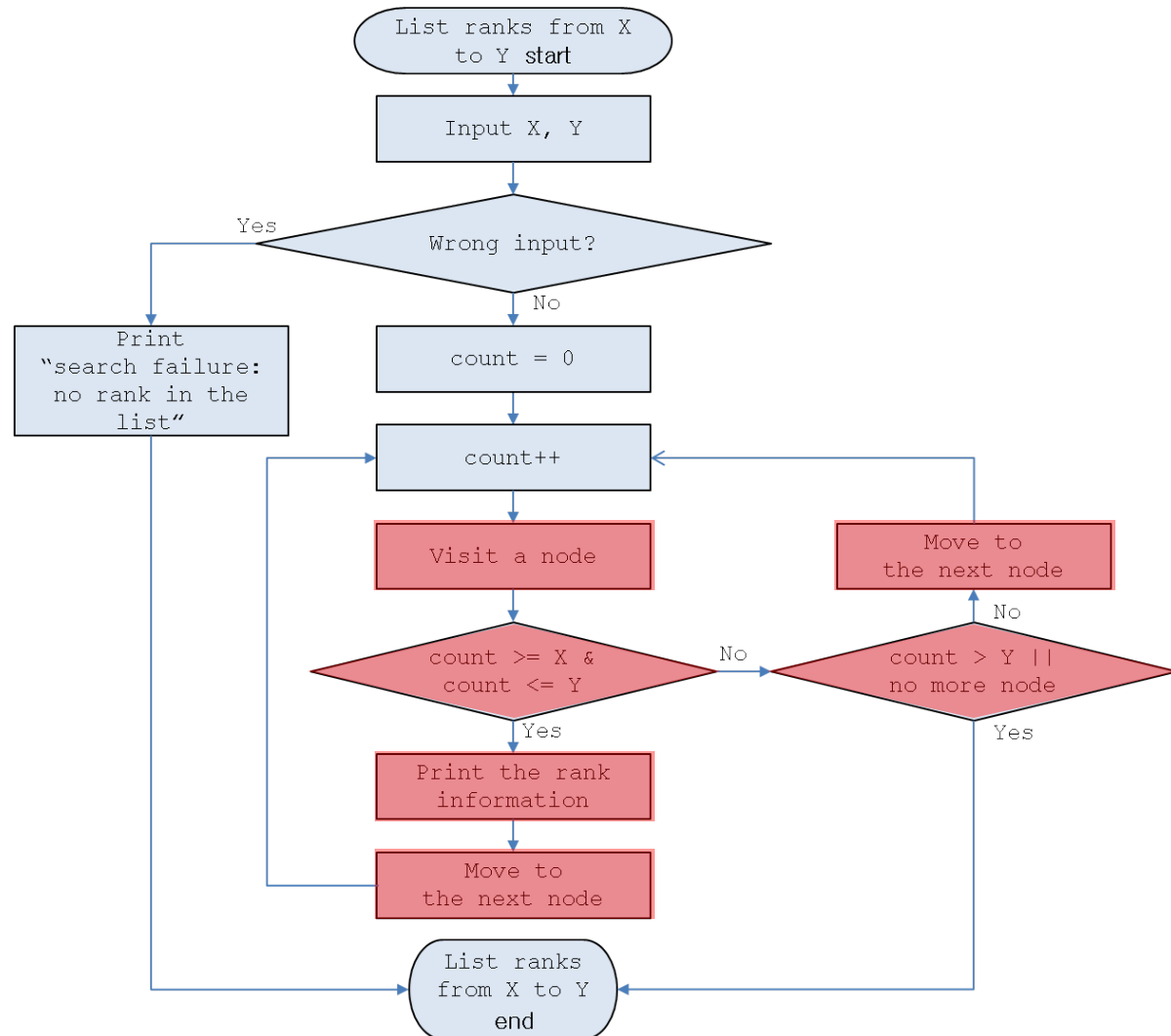


2주차 실습 구현 함수 - rank () (3/3)

□ List ranks

from X to Y

- 랭킹 정보를 저장하고 있는 자료구조의 각 노드를 방문한다.
- 만약 count가 X보다 크거나 같고, Y보다 작거나 같은지 체크한다.
- 그렇다면, 랭킹 정보를 출력하고, 다음노드로 이동한다.
- 그렇지 않다면, 범위를 벗어나거나, 노드가 더 있는지 확인하고 다음노드로 이동하거나 종료한다.



2주차 실습 구현 함수 - newRank ()

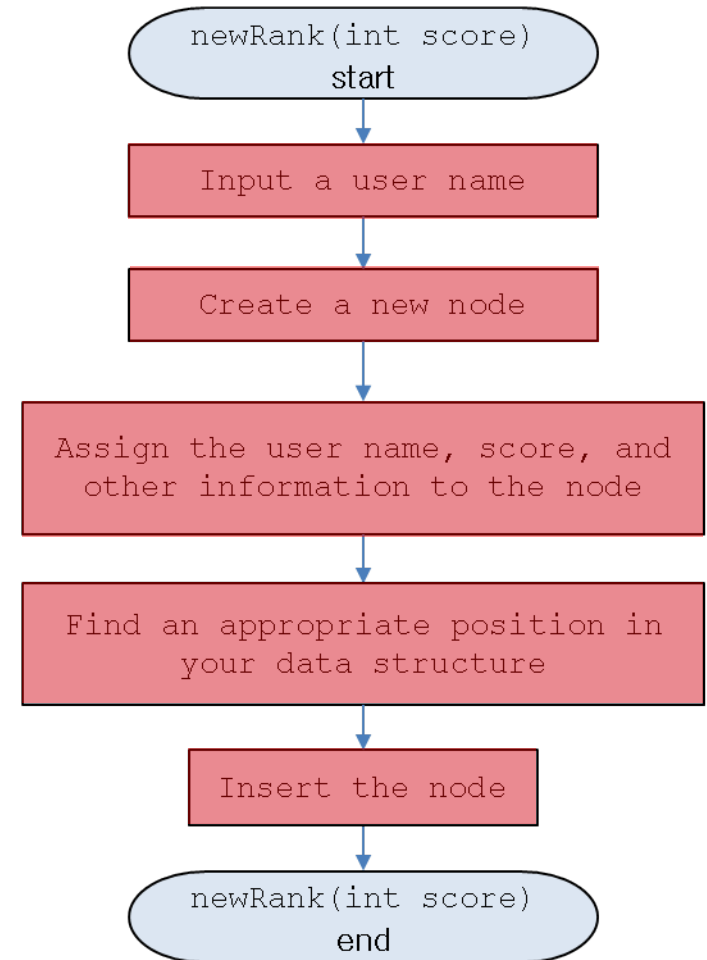
□ newRank ()

- Input

- int score: gameover시,
얻은 score

- Gameover시 호출되는 함수로,
사용자 이름을 입력받고, score
와 함께 랭킹 정보를 구성하여 선
택된 자료구조로 구성된 랭킹 목
록의 적절한 위치에 해당 노드를
삽입한다.

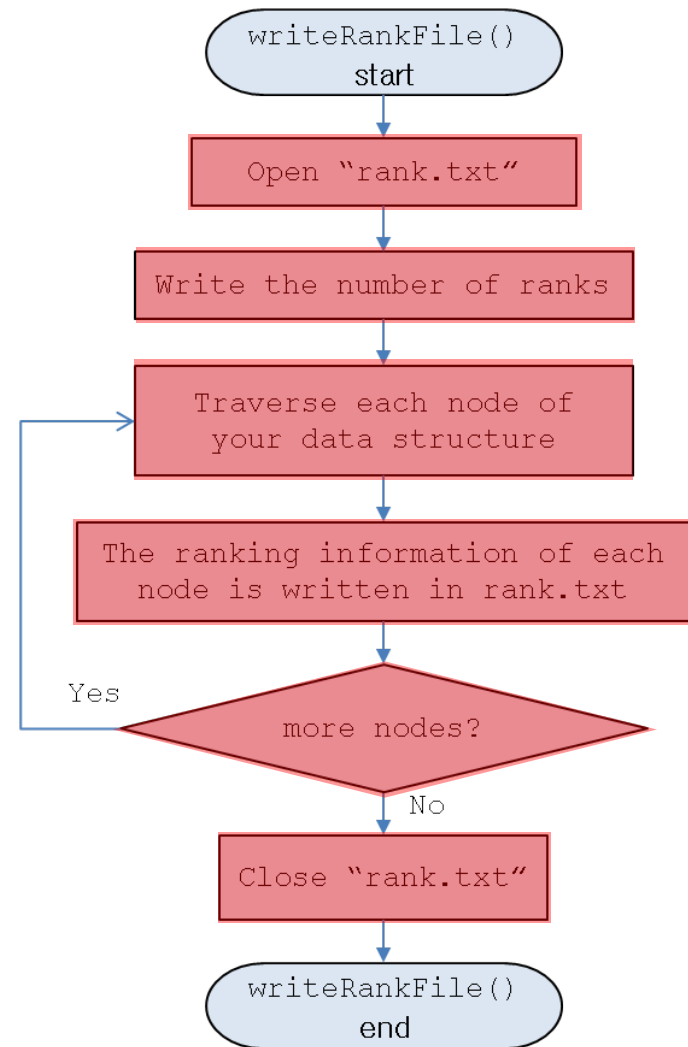
- 사용자 이름을 입력받는다.
- 사용자 이름과 플레이 후 얻은
score를 이용해서 새로운 노드를
생성한다.
- 랭킹 정보를 저장하는 자료구조
에서 적절한 위치를 찾고, 생성된
새로운 노드를 삽입한다.



2주차 실습 구현 함수 - writeRankFile()

□ writeRankFile()

- 추가된 랭킹 정보가 있으면, 새로운 정보를 “rank.txt”에 기록하고, 추가된 정보가 없으면 그대로 종료한다.
- Input 파일인 “rank.txt”를 연다.
- 랭킹 정보들의 수를 “rank.txt”에 기록한다.
- 랭킹정보들을 저장한 각 정보를 탐색하여, “rank.txt”파일에 랭킹 정보를 기록한다.
- 탐색할 노드가 더 있는지 체크하고, 탐색할 노드가 있다면 다음 노드로 이동하고, 그렇지 않으면, 파일을 닫고 종료한다.



랭킹 정보 저장을 위한 structure

```
□ Typedef struct _Node {  
    char name[NAMELEN];  
    int score;  
    ...  
} Node;
```

□ 랭킹 정보를 저장하기 위한 structure

- **char name[NAMELEN]**
 - 사용자의 이름(name)을 저장하기 위한 character array
- **int score**
 - 사용자의 점수를 저장하기 위한 정수형 score 변수
- “...”은 랭킹 정보들을 자료구조에 저장하기 위해서 다른 변수들을 추가할 수 있다는 것을 의미한다.
- NAMELEN은 상수로 16으로 tetris.h에 define되어 있음(수정가능).

구현내용(1/2)

- 메뉴 2번 rank가 랭킹 시스템으로써 동작할 수 있도록 프로그램을 구현한다.

- Flag `MENU_RANK`를 새롭게 define한다.
 - `#define MENU_RANK 2`

구현내용(2/2)

- 테트리스 프로젝트 2주차 구현에는 랭킹 시스템을 구축하기 위한 자료구조의 선택에 자유를 부여한다.
 - 고려할 수 있는 자료구조 : 지금까지 배운 모든 자료구조에서 택 1.
- 자료구조 선택 시 고려사항
 - Game over가 된 후새로운 사용자 이름을 입력 받고, 사용자 이름과 테트리스 게임을 play한 점수를 **기존에 존재하는 랭킹 리스트에 삽입 시 발생하는 시간 복잡도.**
 - **원하는 랭킹을 삭제 시 발생하는 시간 복잡도(고려사항).**
 - 구축된 자료구조에서 랭킹 정보 화면에 출력하기 위해, **원하는 범위의 순위를 추출 할 때, 정렬된 상태로 추출 가능한가의 여부와 시간 복잡도.**
- 결정된 자료구조를 사용해서 위 flow chart의 4가지 함수, **`createRankList()`, `rank()`, `newRank()`, `writeRankFile()`**을 구현한다.

테트리스 프로젝트 2주차 실습 평가

- 선택한 자료구조의 효율성에 대해서 평가한다.

- 자료구조의 효율성은 테트리스의 랭킹 시스템에서 자료구조에 저장된 랭킹 정보 이용하는 동작의 시간 및 공간 복잡도로 평가된다.

- 1st 평가 기준
 - Game over가 되고, 새로운 랭킹 정보(사용자 이름, 점수)가 등록될 때, 시간 및 공간 복잡도

- 2nd 평가 기준
 - 원하는 랭킹 범위를 입력 받고, 랭킹을 추출하기 위한 과정에서 자료구조를 탐색 및 랭킹 추출에서의 시간 및 공간 복잡도

테트리스 프로젝트 2주차 숙제

- 메뉴 2번의 rank에서 새로운 모드 2개를 추가한다.
- 모드 1: 사용자의 이름을 입력 받아, 입력 받은 사용자 이름에 해당하는 모든 랭킹 정보를 찾고, 해당 사용자의 랭킹 정보를 화면에 출력한다.
 - 구축된 자료구조를 유지 및 확장하거나, 새로운 자료구조를 사용할 수 있다.
 - 일치하는 사용자가 있을 때, 해당 사용자의 모든 랭킹 정보를 기존 랭킹 시스템의 출력 방식과 같이, 사용자 이름과 score 순으로 화면에 출력한다.
 - 일치하는 사용자가 없을 때는 “search failure: no name in the list”를 출력한다.
- 모드 2: 원하는 랭킹 정보를 삭제한다.
 - 구축된 자료구조나 새로운 자료구조를 사용할 수 있다.
 - 입력은 삭제하길 원하는 랭킹(정수)를 입력 받는다.
 - 일치하는 랭킹이 있을 때는 랭킹 정보를 삭제하고, “result: the rank deleted”를 출력한다.
 - 일치하는 랭킹이 없을 때는 “search failure: the rank not in the list”를 출력한다.

테트리스 프로젝트 2주차 숙제 - 모드 1(1/3)

□ 입력 파일

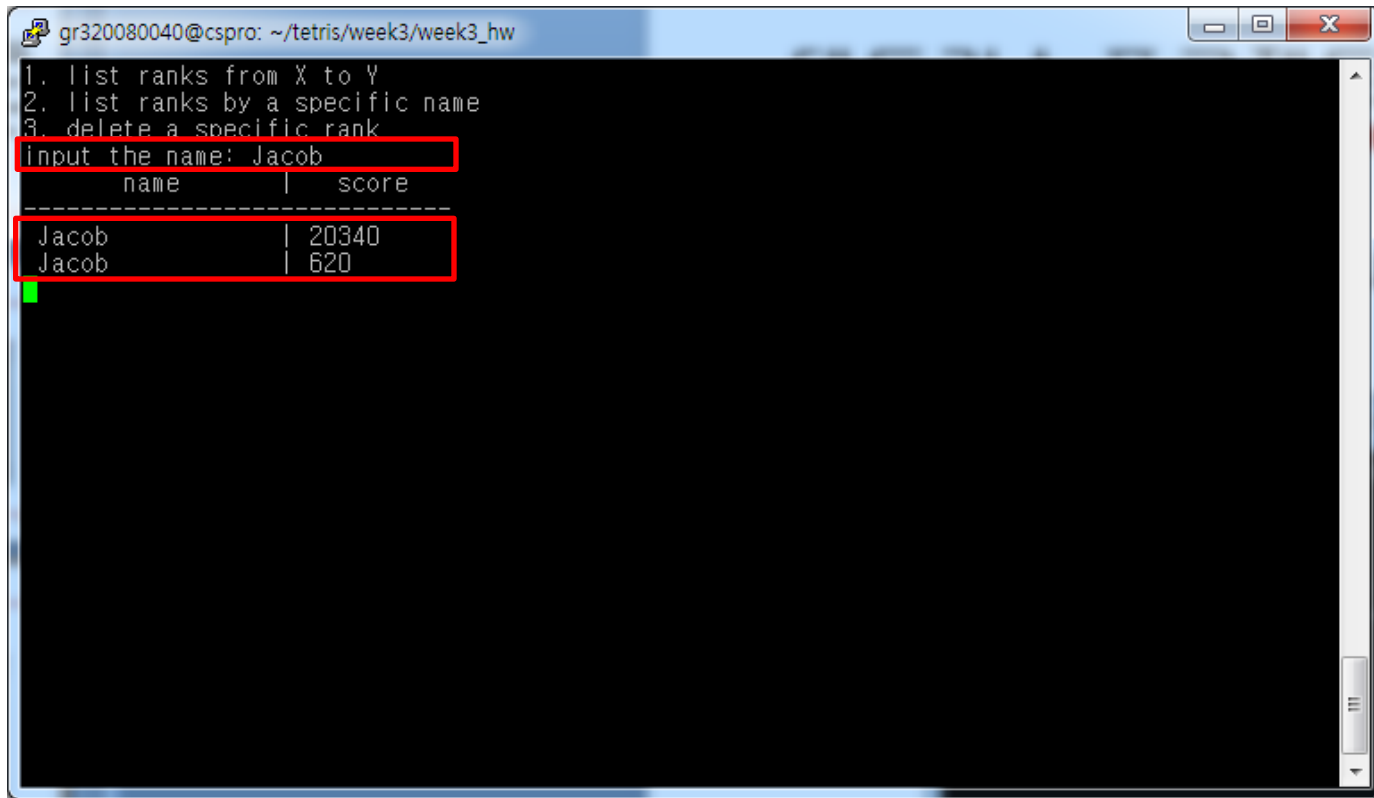
- rank.txt

[illegible]

테트리스 프로젝트 2주차 숙제 – 모드 1(2/3)

□ 출력 예제 1

- Jacob을 입력하여 Jacob의 모든 랭킹 정보들을 찾는 경우



A terminal window titled 'gr320080040@cspro: ~/tetris/week3/week3_hw' displays the following text:

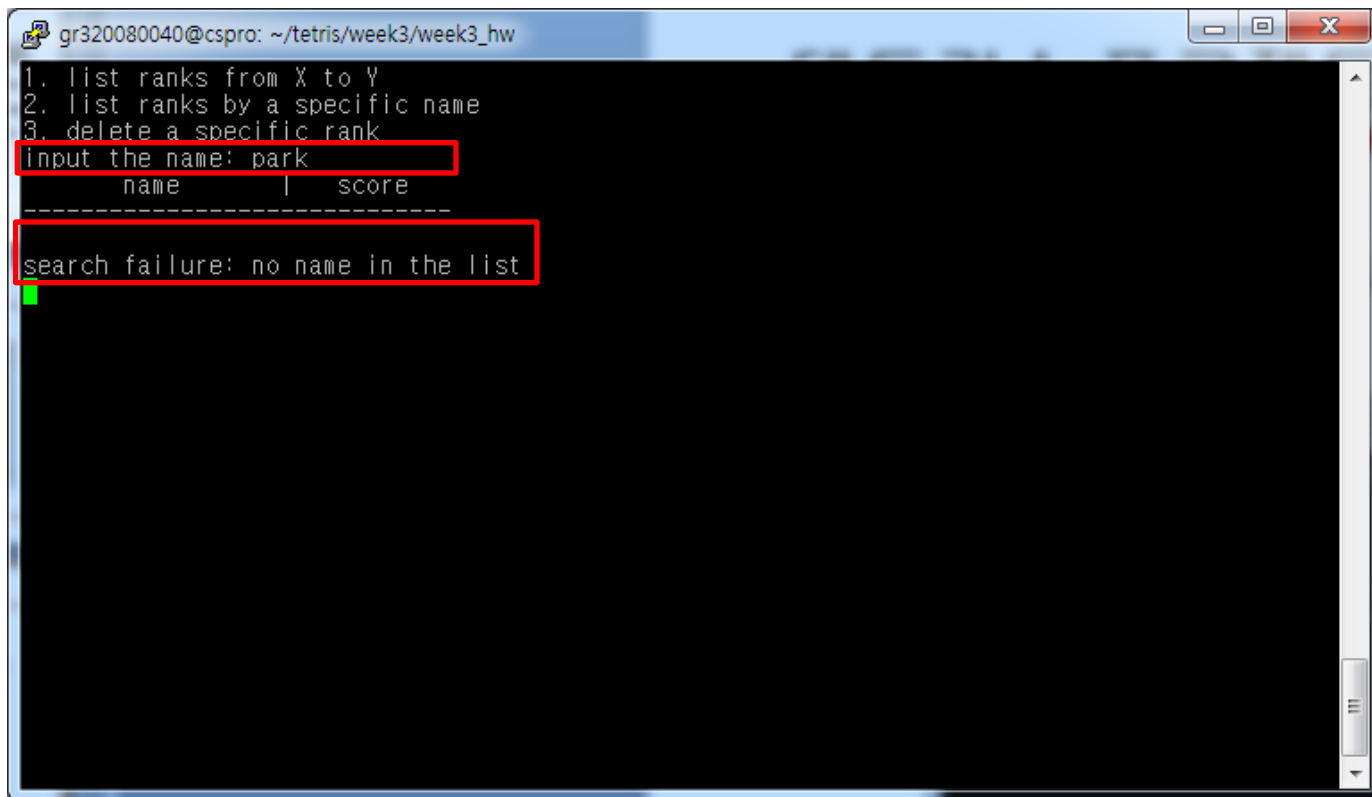
```
1. list ranks from X to Y
2. list ranks by a specific name
3. delete a specific rank
input the name: Jacob
  name      | score
-----
Jacob      | 20340
Jacob      | 620
```

The input 'Jacob' and the resulting table are highlighted with red boxes. A green cursor is visible on the line following the table.

테트리스 프로젝트 2주차 숙제 – 모드 1(3/3)

□ 출력 예제 2

- park을 입력하여 park의 랭킹 정보를 찾지만, 그 정보들이 존재하지 않는 경우



```
gr320080040@cspro: ~/tetris/week3/week3_hw
1. list ranks from X to Y
2. list ranks by a specific name
3. delete a specific rank
input the name: park
  name | score
-----
search failure: no name in the list
```

테트리스 프로젝트 2주차 숙제 - 모드 2(1/5)

□ 입력 파일

- rank.txt
- Kim의 정보를 삭제할 예정.

A screenshot of a Windows command prompt window. The title bar shows the user 'gr320080040@cspro' and the current directory '~/tetris/week3/week3_hw'. The terminal output displays the contents of a file named 'rank.txt':

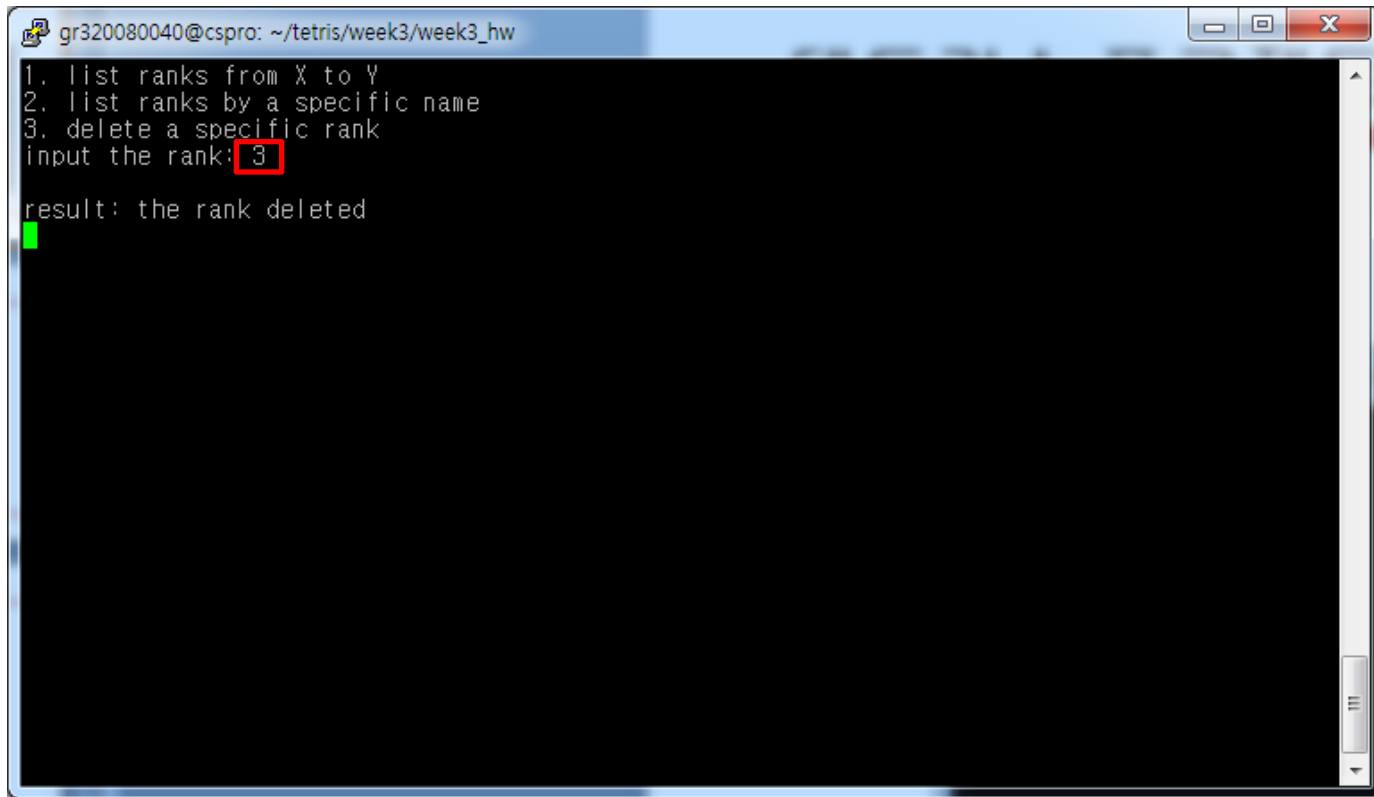
```
Jacob 20340  
Park 17990  
Kim 7430  
Jacob 620  
Chris 450  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```


The line 'Kim 7430' is highlighted with a red rectangular box. At the bottom of the window, a status bar indicates '"rank.txt" 6L, 54C' on the left, '1,1' in the center, and '모두' (All) on the right.

테트리스 프로젝트 2주차 숙제 – 모드 2(2/5)

□ 출력 예제 1

- Kim의 랭킹 정보를 삭제하기 위해, 3을 입력

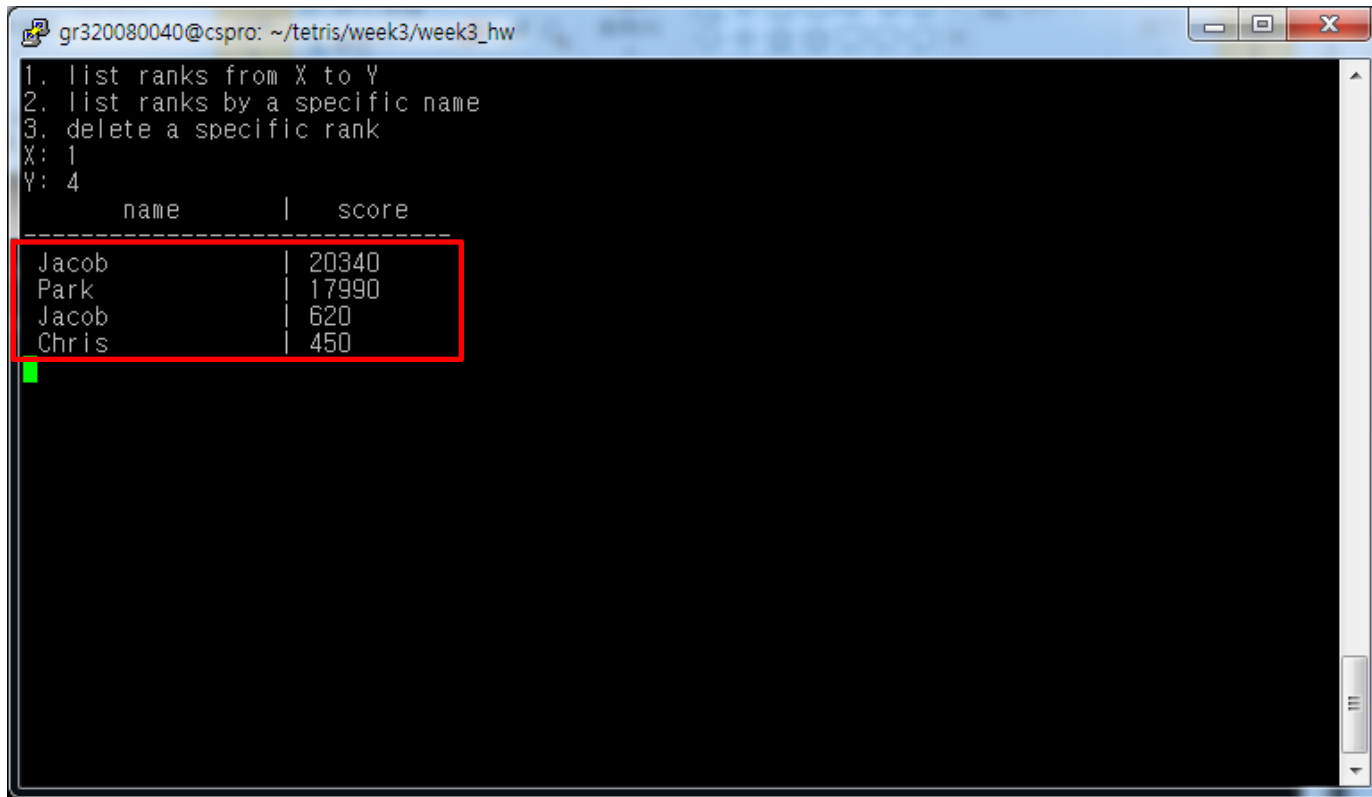


```
gr320080040@cspro: ~/tetris/week3/week3_hw
1. list ranks from X to Y
2. list ranks by a specific name
3. delete a specific rank
input the rank: 3
result: the rank deleted
```

테트리스 프로젝트 2주차 숙제 – 모드 2(3/5)

□ 출력 예제 2

- Kim의 랭킹 정보가 삭제된 것을 확인.



```
gr320080040@cspro: ~/tetris/week3/week3_hw
1. list ranks from X to Y
2. list ranks by a specific name
3. delete a specific rank
X: 1
Y: 4
name | score
-----
Jacob | 20340
Park  | 17990
Jacob | 620
Chris | 450
```

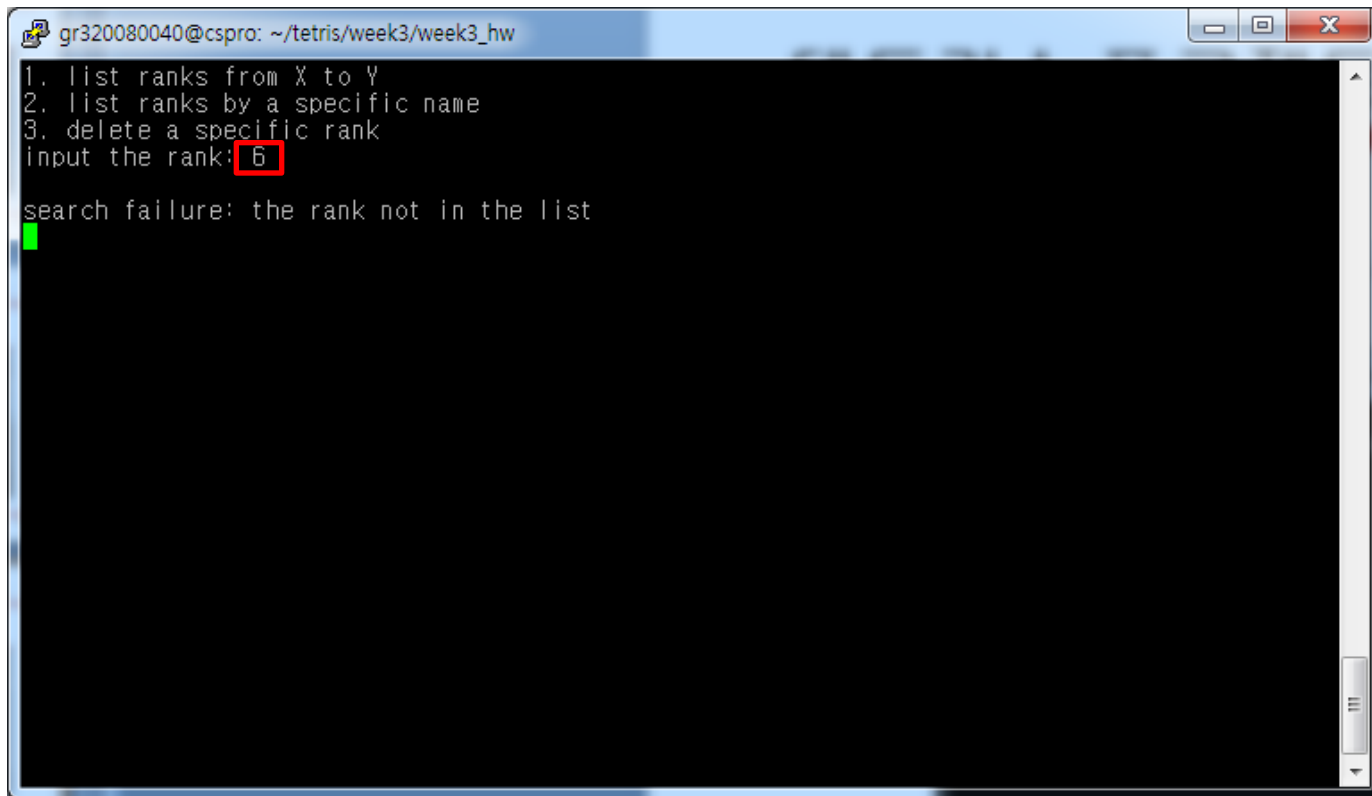
□ 출력 예제 3

-
- A screenshot of a terminal window titled "gr320080040@cspro: ~/tetris/week3/week3_hw". The terminal displays the contents of a file named "rank.txt":
-
- ```
Jacob 20340
Park 17990
Jacob 620
Chris 450
```
- 
- The first four lines are enclosed in a red rectangular box. Below these, there are several lines of tilde (~) characters. At the bottom left, it says '"rank.txt" 5L, 45C'. At the bottom right, it shows "1,1" and "모두" (All).

# 테트리스 프로젝트 2주차 숙제 – 모드 2(5/5)

## □ 출력 예제 4

- 존재하지 않는 랭킹을 삭제하려고 할 때는 다음 메시지를 출력한다.

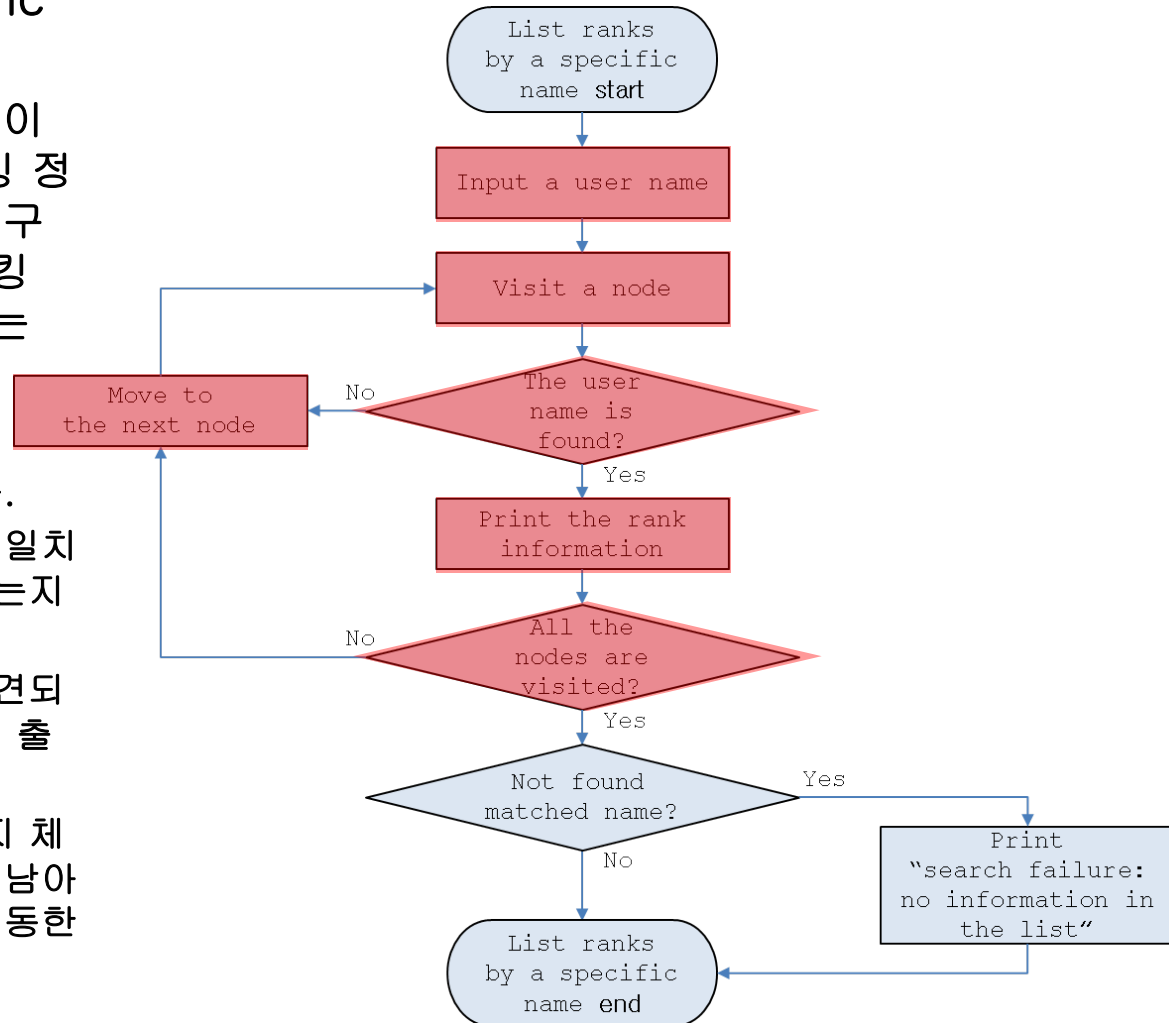


```
gr320080040@cspro: ~/tetris/week3/week3_hw
1. list ranks from X to Y
2. list ranks by a specific name
3. delete a specific rank
input the rank: 6
search failure: the rank not in the list
```

## 2주차 숙제 구현 함수 - rank () (1/2)

### □ List ranks by a specific name

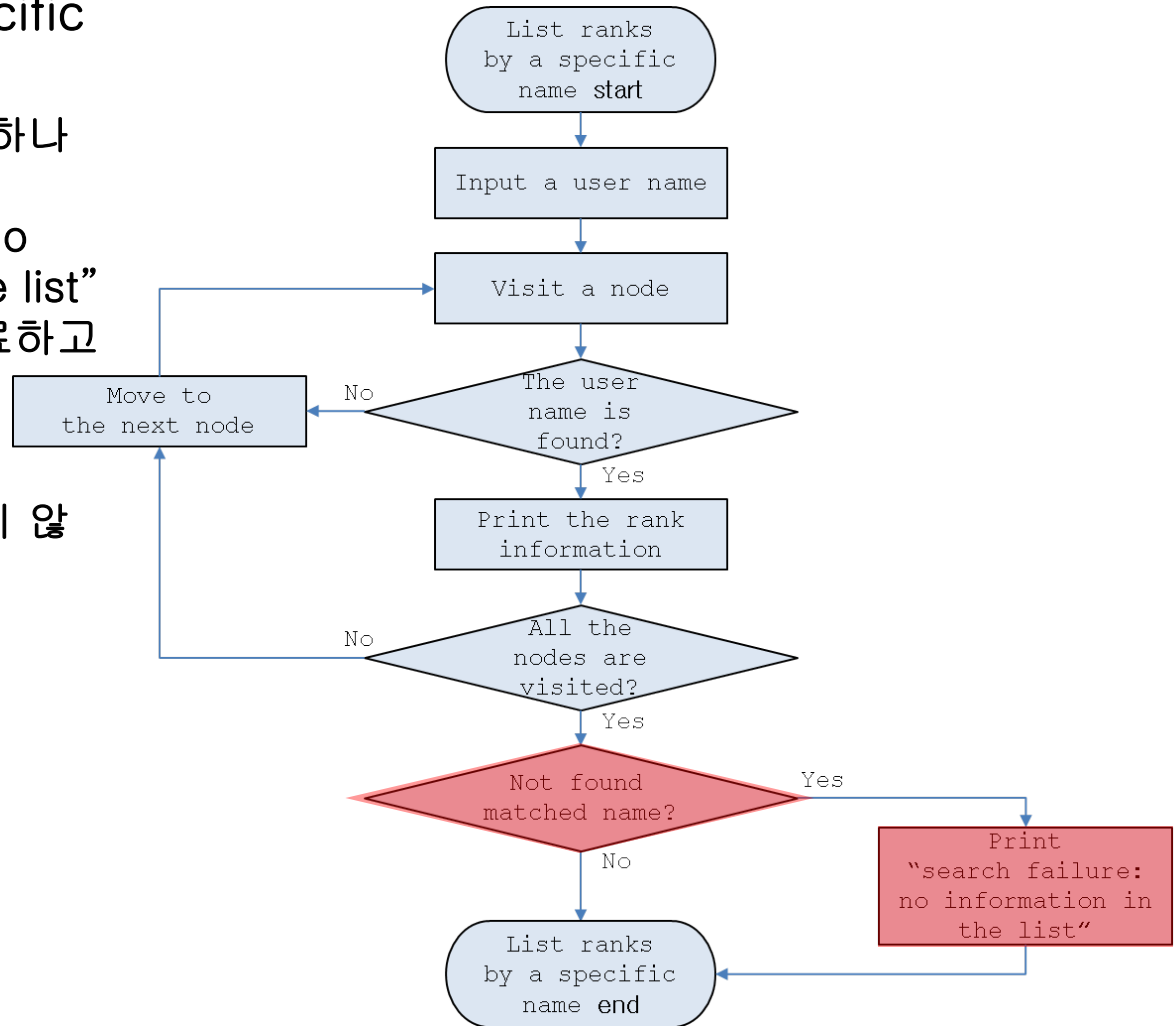
- 찾고자 하는 사용자 이름을 입력 받고, 랭킹 정보를 갖고 있는 자료구조에서 일치하는 랭킹 정보를 찾고 출력하는 함수.
- 사용자이름을 입력한다.
- 각 노드를 방문하면서, 일치하는 사용자 이름이 있는지 확인한다.
- 찾는 사용자 이름이 발견되면, 랭킹 정보를 화면에 출력한다.
- 모든 노드를 방문했는지 체크해서, 방문할 노드가 남아있으면, 다음 노드로 이동한다.



## 2주차 숙제 구현 함수 - rank () (2/2)

### □ List ranks by a specific name

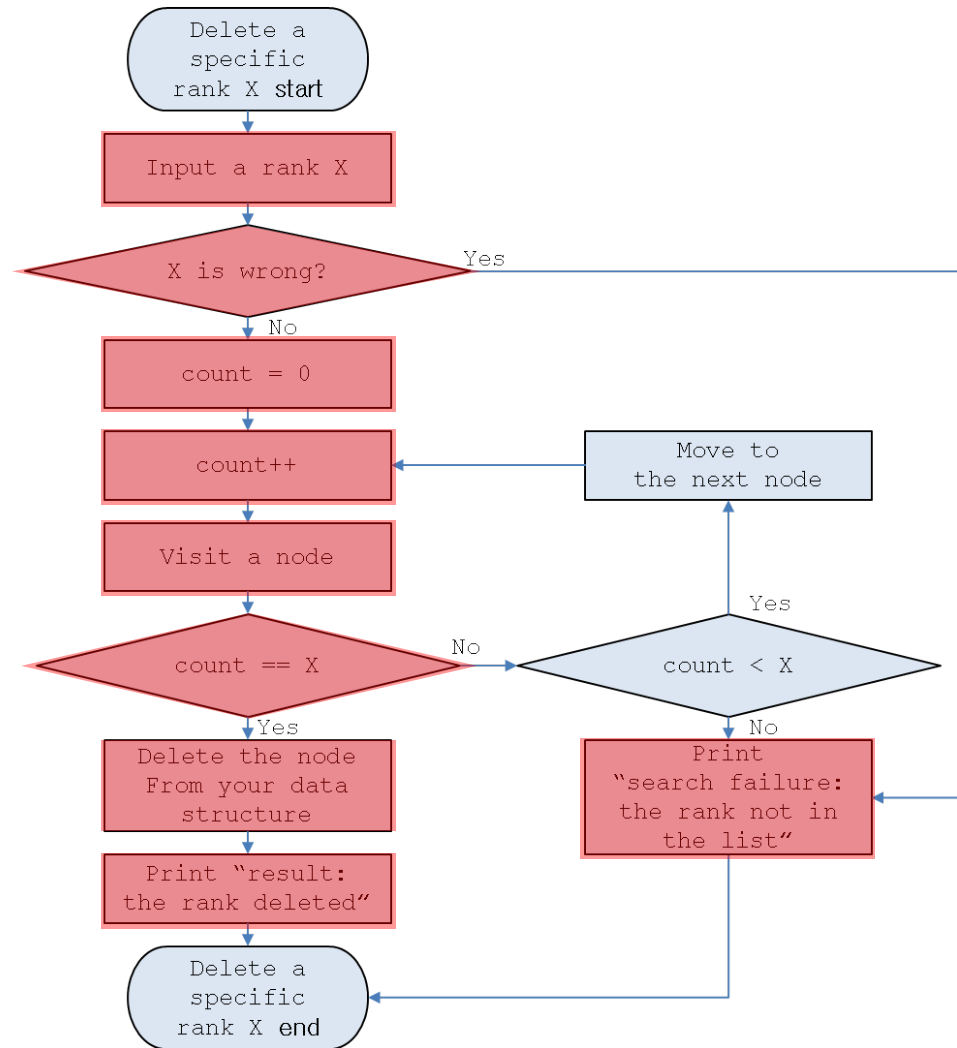
- 일치하는 이름을 하나도 찾지 못했다면, “search failure: no information in the list”를 출력한 후, 종료하고
- 일치하는 이름을 찾았다면, 아무런 메시지를 출력하지 않고 종료한다.



## 2주차 숙제 구현 함수 - rank () (1/2)

### □ Delete a specific rank X flow chart

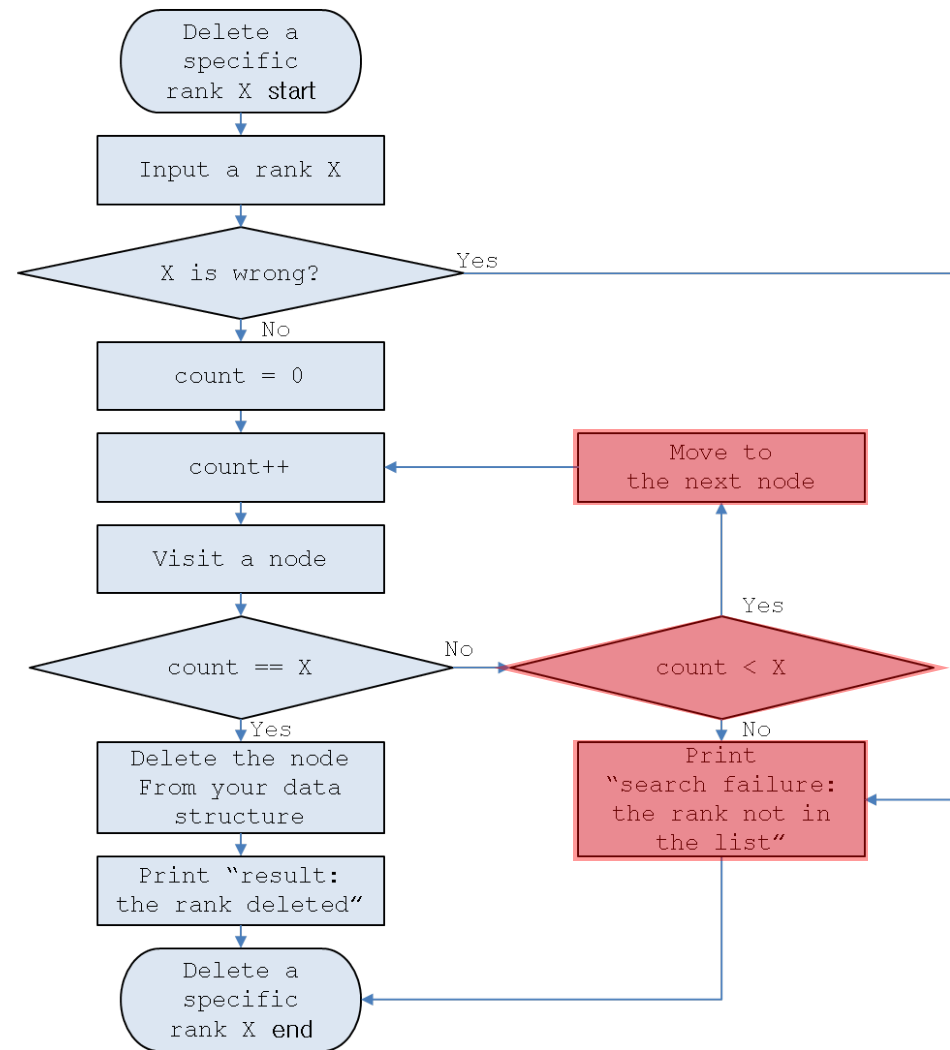
- 삭제하길 원하는 랭킹(정수)을 입력 받고, 랭킹 정보를 저장하고 있는 자료 구조에서 해당 랭킹에 해당하는 랭킹 정보(노드)를 삭제하는 함수.
- 삭제하길 원하는 랭킹(정수)를 입력한다.
- 입력받은 랭킹이 잘못되었는지 체크해서 잘못되었다면, 메시지를 출력하고 종료한다.
- 변수 count를 0으로 초기화한다.
- 변수 count를 1증가한다.
- 노드를 방문해서 X번째인지 확인해서, 그렇다면 노드를 삭제한다.
- 화면에 “result: the rank deleted”를 출력한다.



## 2주차 숙제 구현 함수 - rank () (2/2)

### □ Delete a specific rank X flow chart

- 점수에 대해서 내림차순으로 정렬된 리스트 상에서 현재 방문한 노드가 X 번째 노드인지 체크해서, X번째 노드가 아니라면, count값이 X보다 작은지 확인한다.
- count값이 X보다 작다면, 다음 노드로 이동한다.
- count값이 X보다 크다면, “search failure: the rank not in the list”를 출력하고 종료한다(앞에서 X가 잘못된 입력인지 체크하므로 이 경우는 발생하지 않으나, 조건문이 모든 경우 고려한다는 것을 보여주기 위해 플로우 차트 상에서만 명시).





# 테트리스 프로젝트 2주차 숙제 평가

---

- 테트리스 프로젝트 2주차 숙제에 대한 평가는 실습에서와 마찬가지로 선택한 자료구조의 효율성에 대해서 평가한다.
  
- 자료구조의 효율성은 테트리스의 랭킹 시스템에서 자료구조에 저장된 랭킹 정보 이용하는 동작의 시간 및 공간 복잡도로 평가된다.
  
- 1<sup>st</sup> 평가 기준
  - 사용자 이름으로 원하는 사용자의 이름을 검색할 때, 시간 및 공간 복잡도
  
- 2<sup>nd</sup> 평가 기준
  - 원하는 랭킹 정보(사용자 이름, 점수)를 삭제할 때, 시간 및 공간 복잡도

# 테트리스 2주차 실습 결과 보고서

---

- 실험시간에 작성한 랭킹 시스템의 자료구조와 랭킹 시스템의 각 기능에 대한 알고리즘을 요약하여 기술하시오. 본인이 선택한 랭킹 시스템을 구현하기 위한 자료구조가 왜 효율적인지 시간 및 공간복잡도를 통해 보이고, 설명하시오.
- 본 실험 및 숙제를 통해 습득한 내용을 한 내용을 기술하시오.

# 테트리스 프로젝트 3주차 실습 예비 보고서

---

- 교재를 참조하여 테트리스 프로젝트 3주차에 구현하는 추천 기능은 어떤 원리로 작동되는지 설명하시오. 그리고 추천 기능을 구현하는 tree 구조의 장, 단점을 기술하시오.
- Tree 구조의 단점을 해결할 방법에 대해서 2가지 이상 생각하고, 그 idea에 대해 기술하시오.