9주차 결과보고서

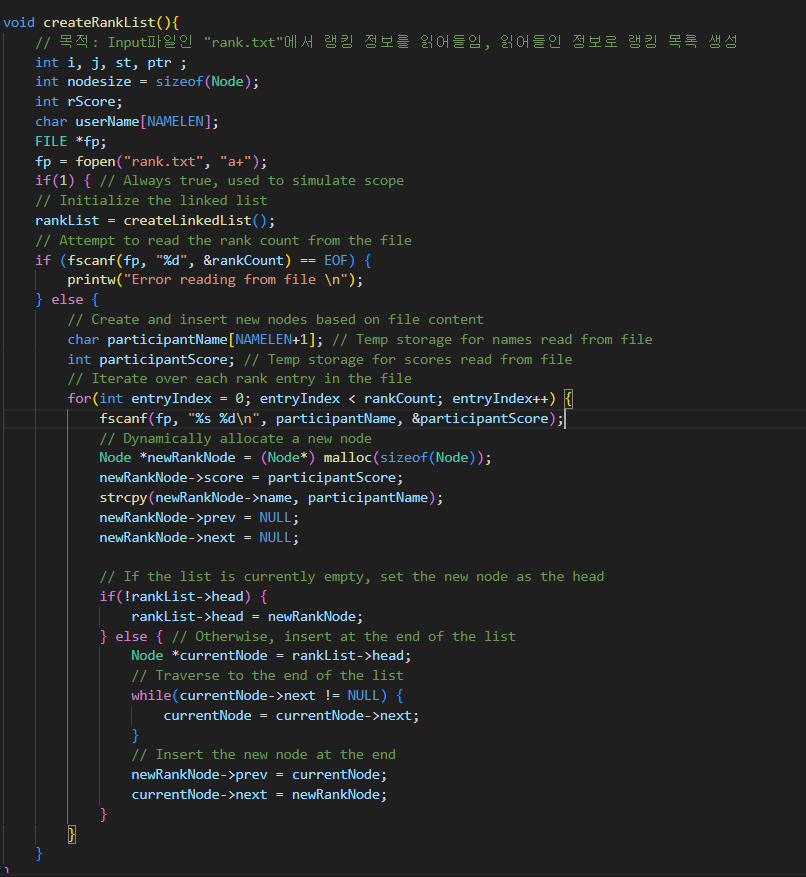
전공: 컴퓨터공학과 학년: 4학년 학번: 20191612 이름: 윤기웅

**1. (문제 및 제목은 작성하지 말 것. 답안만 작성하면 됩니다.)**

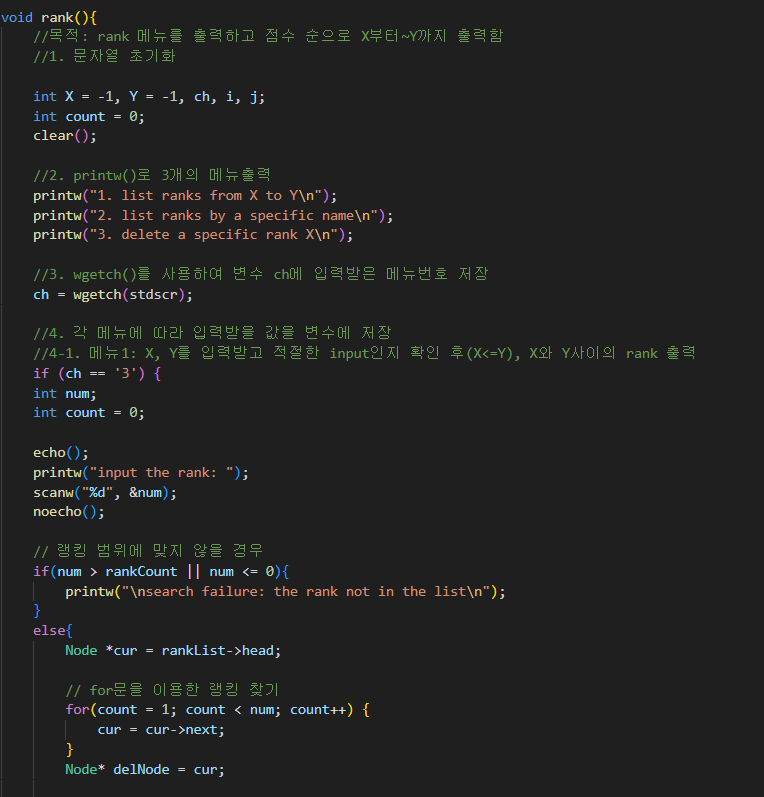
.................



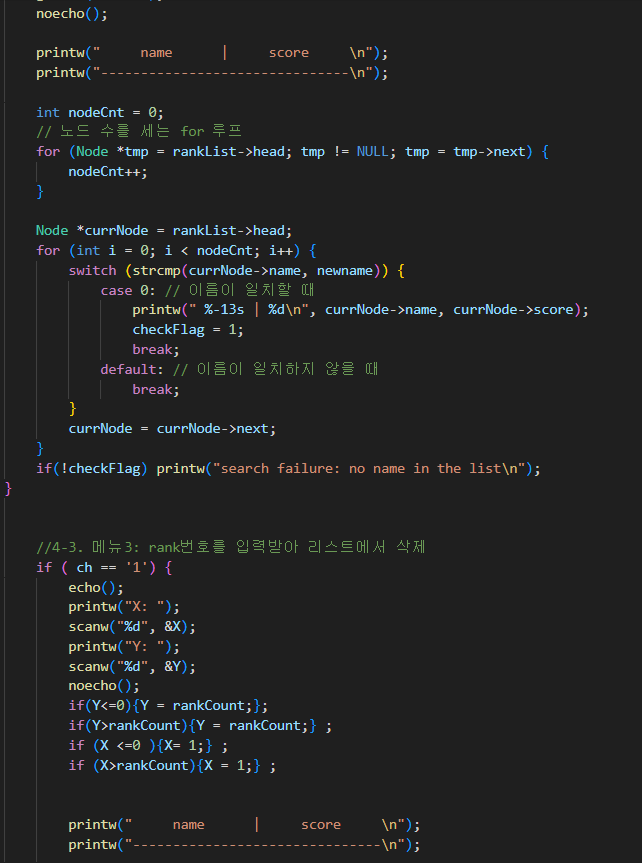
연결리스트를 만들고 초기화 하는 함수이다.

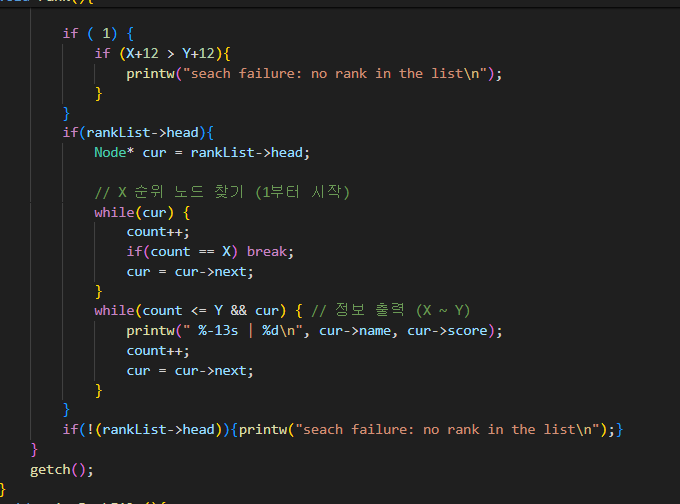


랭크 리스트를 만들어서 점수를 기록한다. Rank.txt를 읽어서 점수 정보를 받아오고 연결리스트로 만든다. 첫 줄의 등록된 인원 수를 읽고 그만큼 연결리스트 노드를 생성한다. 함수 시간 복잡도는 O(n)이고 공간 복잡도는 O(n)이 된다.



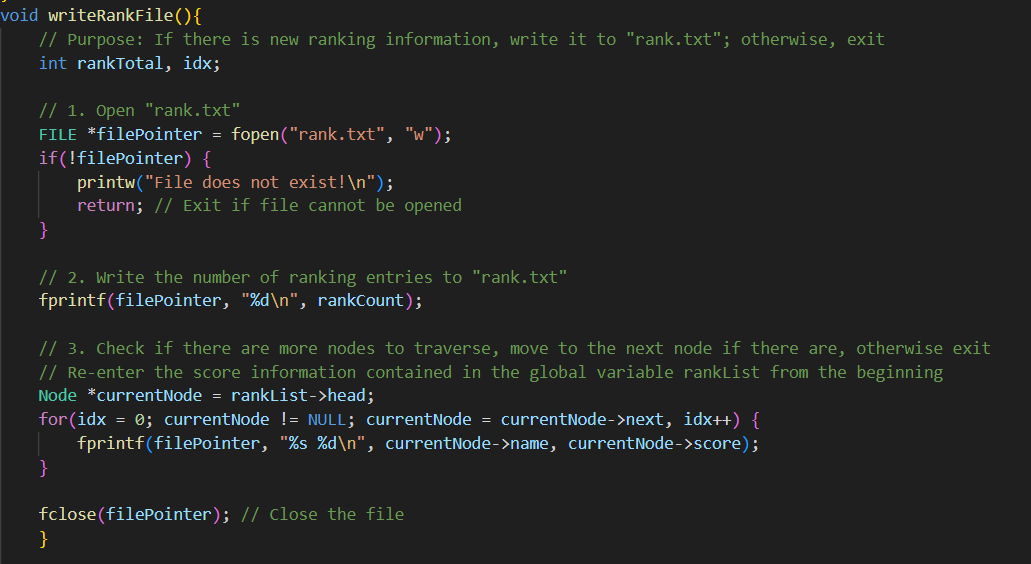




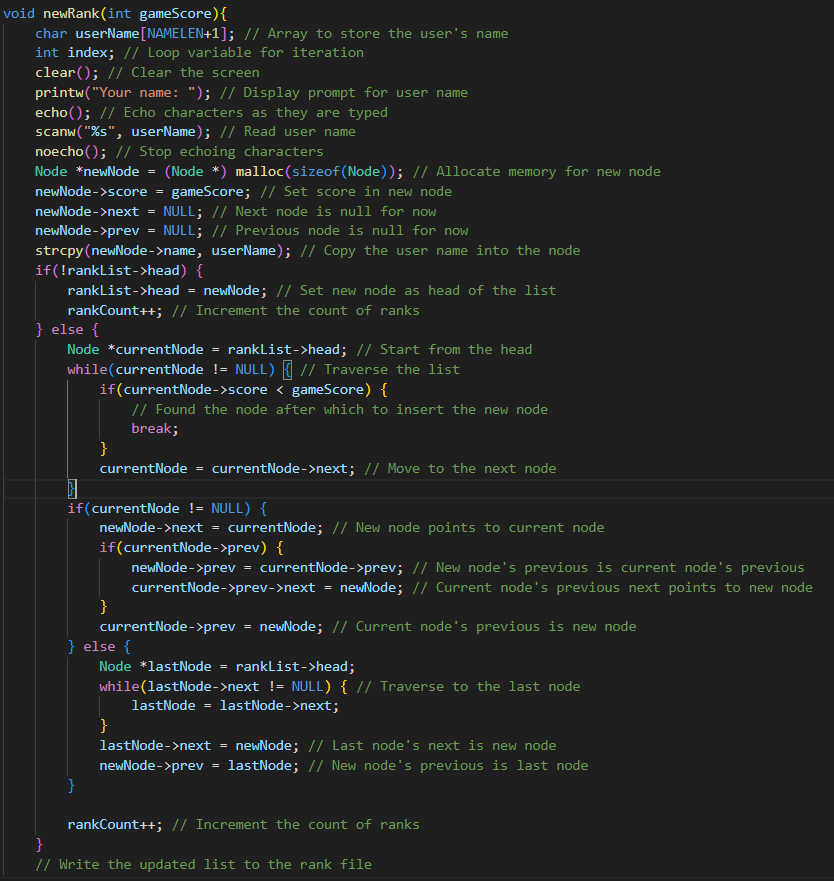


Rank의 함수에서는 2를 입력하고 나서 나온 창에서 랭크를 출력할 숫자 1,2,3 중에

하나를 입력 받는다. 특정 범위에 있는 등수의 사람들을 출력하거나, 특정 이름의 사람의 점수를 출력하거나 아니면 제거를 하면 된다. 연결리스트의 노드 제거할 때 노드의 위치가 몇 번인지에 따라서 다르게 처리해준다. 만약 범위를 잘못 입력하게 되면 전체 등수를 출력하게 만든다. 이 함수 시간 복잡도는 연결리스트 노드의 개수가 n이라 O(n)이고 공간 복잡도는 또한 O(n)이 된다.



**새로운 랭킹 정보로 연결리스트가 최신화 되면 rank.txt도 최신화 하는 것이다. 이전과는 반대로 연결리스트로부터 노드 정보를 갖고와서 파일에 다시 적으면 된다. 시간 복잡도와 공간 복잡도는 둘다 O(n)이 된다.**

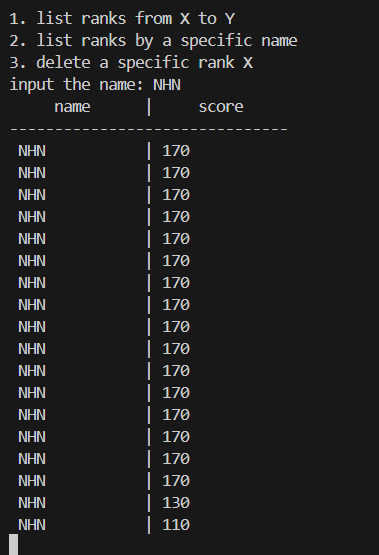
****

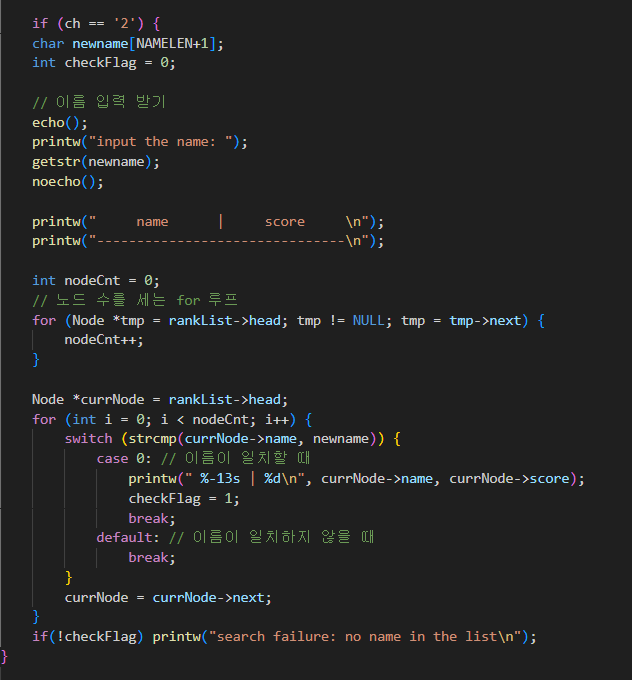
**Newrank : 게임이 끝나면 이름을 입력받고 이를 연결리스트에 넣어주면 된다.**

**연결리스트가 텍스트에서 받을 때 정렬되어 있으니까 거기에 맞는 점수 위치의 노드에 삽입하면 된다. 이후, writerankfile을 실행해서 rank.txt 파일을 최신화 한다. 함수의 시간, 공간 복잡도는 이전처럼 O(n)이 된다.**

**연결리스트는 다른 자료구조보다 장점이 몇 가지 있다. 먼저 동적인 크기를 갖기 때문에 메모리 효율을 높이고 국한된 데이터 크기에 제한되지 않는다. 그리고 노드 삽입과 삭제가 쉽게 할 수 있고 시간 복잡도가 낮다. 대부분 연산이 노드 개수에 비례하여 시간 복잡도와 공간 복잡도가 모두 O(n)이 된다.**

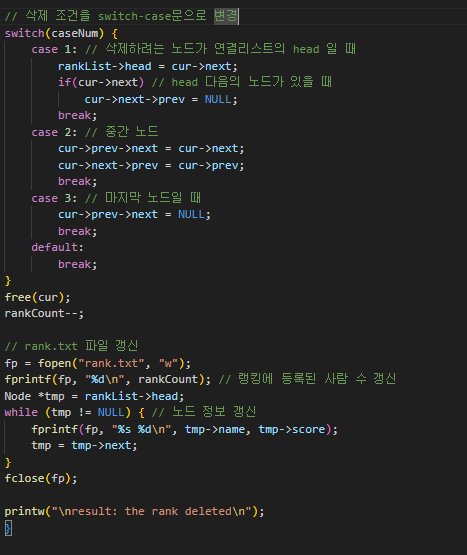
**2. 과제**





같은 이름인 NHN으로 기록된 랭크 점수들을 모두 출력이 가능하다. 코드에서는 switch문을 사용해서 이름이 일치하는 경우에 모두 찾아서 차례대로 출력하면 된다. 만약 찾고자 하는 이름이 하나도 없는 경우는 검색 실패 했음을 알려주면 된다. 연결리스트 모두를 찾는 과정이 있어서 시간,공간 복잡도 모두 O(n)이 된다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

기존의 코드에서는 제거할 대상을 입력받고 만약 노드에 같은 이름이 있다면 노드에서만 제거하는 것인데 이번에는 rank.txt파일에도 해당 이름을 가진 랭크 정보를 지우는 것이다. txt파일을 열어서 다시 등록된 랭커 수를 입력하고 그 다음 줄부터 정보를 갱신하는 코드를 추가했다. 만약 찾고자 하는 이름이 없으면 탐색에 실패했다고 출력한다. 해당 함수 또한 연결리스트 모두 찾는 과정으로 인해 시간 공간 복잡도가 O(n)이 된다.

연결리스트의 사용은 동적인 크기 조정이 가능하고 삽입 삭제가 편하다는 장점이 있다. 필요할 때마다 메모리를 할당하면 되니 메모리의 관리에도 편하다. 또한 양방향 연결리스트의 경우는 이전과 다음 노드 모두 접근 가능해서 유연한 관리가 가능하다.