과제 해결 방법

- 주어진 과제를 해결하기 위해 고민한 내용과 방법을 적는다.  
\* 소스 코드를 보고서에 적을 필요는 없다.  
- 과제 1, 2의 화면 출력 결과를 보고서에 적어라.  
- 과제 3의 결과는 조건에 맞는 객체의 수와 파일 크기를 보고서에 적는다.  
- 과제 4의 결과는 실행 결과 화면 일부를 캡처하여 보고서에 추가하라.

파일을 일단 읽어야 하겠지?  
그래서 파일을 읽어 봤다  
정말로 200만개만큼 읽는지 테스트 해보기  
테스트해봤는데 200만개 이상 자꾸 읽어진다  
ofstream이 유효한지 테스트해보자. 어라? 유효한데 왜 안 읽어질까  
폴더안에 보니까 읽어야할 파일인 2022 STL 과제 파일이 제대로 된 경로에  
안 들어가있다.  
폴더안에 집어넣고 다시 돌려  
다시 돌리니까 boundary exception이 일어난다  
메모리의 개수인 num을 읽고 그 메모리를 읽는데 문제가 있었음  
num만큼 byte가 써져있는데, num이 몇백 바이트짜리는 변수에다가 그냥 집어넣을수가 없어서 동적 할당으로 잡아줬음 (근데 메모리 해제가 없어서 visual studio diagnostic tool에서 확인해보면 메모리가 1GB가 넘게 산으로 올라가는 것을 볼수 있다 그래서 나는 어차피 읽어낸 num의 메모리 가지고 하는 것도 없어서 매 반복마다 삭제해주었다.)  
돌려보니까 200만 1개까지 돌아가네?  
이건 또 무슨일?  
ofstream로 선언된 변수는 파일의 커서를 가리키고 있으니까 while 문 조건으로 한번 넣어보자  
그래도 똑같다 이유는 파일을 size만큼 읽어내기 때문에 다 읽고나서 파일의 마지막에 써져있는 무언가를 ofstream 변수가 가리키고 있기 때문.  
교수님께 질문해보았다.  
교수님께서 in.read를 조건으로 체크할 수 있다는 것을 상기시켜주셨다  
이렇게하니까 200만개가 잘 reading 된다.  
확인하려고 읽은 player 정보를 count 로 세어서 출력해봤는데 200만개 정확하게 읽어지는 것을 확인했다.  
출력한거랑 안한거랑 속도차이가 엄청나게 나는 것도 확인했음  
벡터에서 어레이로 고침  
파일을 main 함수 while 문 안에서 코드 손수 써서 해결 냈었는데  
Read 함수를 player 클래스 내부에 작성해서 해결하는 방법으로 개선  
2024개로 고친 과정

4번 문제 풀 때 처음에는 선형 알고리즘인 find를 사용했지만, 후에 binary\_search와 같은 방법의 알고리즘으로 변경 이 과정에서 sort를 id순으로 하기 때문에 2번째 정렬부터는 sort의 시간복잡도가 n^2이 된다는 문제점. 이를 해결하기 위한 bool 변수 sortedById를 추가하여 재정렬을 하지 않게 함

제출한 문제해결 방법이 효율적이라고 주장하는 내용/개선할 부분

과제를 하면서 느낀 점