**2025 STL 과제 보고서**

2023180007 김도윤

문제 1

고민한 점 : player 클래스 내부에는 unique\_ptr<char[]>를 통해서 동적 메모리를 관리하고 있어서 read()로는 충분 하지 않았다.

해결 방법 : player::read(std::istream&) 함수를 정의하여서, player 구조체 본체를 읽은 후, num 만큼 p가 가르키는 메모리를 새로 할당하고 내용을 읽었다.

컨테이너 선택 : std::array<Player,250’000>로 고정 크기 배열을 사용하였다. 문제에서 파일에는 250'0000(2백5십만) 개의 Player 객체가 기록되어 있는 것을 알 수 있었기 때문에 선택 하였다.

읽은 방식 : 범위 기반 for문을 사용해서 모든 player 객체에 접근해 player 클래스에 추가 한 멤버 함수 read를 사용해 읽었다.

문제 2

**2. 점수가 가장 큰 Player를 찾아 화면에 출력하라.**

**Player의 평균 점수를 계산하여 화면에 출력하라.**

**- 어떻게 계산하였는지 보고서에 설명하라.**

고민한 점 : 전체를 탐색해 가장 큰 점수와 평균 점수를 효율적으로 구할 수 있는 stl 알알고리즘 어떤 것이 있을지 고민하였다.

해결 방법 : std::max\_element()를 사용하여서 최고 점수 보유자를 찾아내였다.

std::accumulate()를 통해 총합을 구하고, 플레이어의 size로 나눠 평균을 계산하였다.

문제 3

**3. id가 서로 같은 객체를 찾아 "같은아이디.txt"에 기록하라.**

**id가 같은 객체는 모두 몇 개인지 화면에 출력하라.**

**파일에는 id가 같은 Player 객체의 이름과 아이디를 한 줄 씩 기록한다.**

**- 어떻게 같은 id를 찾았는지 보고서에 설명하라.**

고민한 점 : id 중복 여부 탐색 및 그룹화

**해결 방법:**

* std::unordered\_map<size\_t, std::vector<const Player\*>>를 사용하여 id별로 Player들을 묶음. 이를 사용하면 id별로 그룹화를 해서 해당 아이디마다 player를 추가할 수 있기 때문에 해당 자료구조를 선택 하였다.
* 이 자료구조를 모두 순회 하면서 해당 아이디의 valude 값의 크기가 2 이상인 그룹만 "같은아이디.txt"에 기록 후 중복된 id를 새는 변수의 값을 추가
* 중복된 Player 수를 세어 출력.

**4. Player의 멤버 p가 가리키는 메모리에는 파일에서 읽은 num개의 char가**

**저장되어 있어야 한다**

**메모리에 저장된 char를 오름차순으로 정렬하라.**

**'a'가 10글자 이상인 Player의 개수를 화면에 출력하라.**

**- 어떻게 정렬하였고 어떻게 개수를 셌는지 보고서에 설명하라.**

 **고민한 점:** 포인터가 가리키는 메모리 영역에 대해 안전하게 정렬 및 탐색.

 **해결 방법:**

* 모든 객체를 순회하면서, 각 객체의 p가 가리키는 char[] 영역을 std::sort()를 사용해 정렬하였다.  
  p는 C 스타일 문자열을 가리키고 있으므로, 별도의 비교 함수 없이도 알파벳 순으로 정렬되었다.
* 정렬을 함과 동시에 std::count()로 'a'의 개수를 세고, 10개 이상인 경우만 카운트하였다.

5**. [ LOOP ] id를 입력받아 존재하는 id라면 다음 내용을 한 번에 화면 출력하라.**

- **Player를 id 기준 오름차순으로 정렬하였을 때**

해당 id 포함 앞과 뒤 Player의 정보를 출력한다.

id가 같은 Player가 둘 이상이면 이들의 정보를 모두 출력하여야 한다.

- **Player를 name 기준 오름차순으로 정렬하였을 때**

해당 name 포함 앞과 뒤 Player의 정보를 출력한다.

같은 name이 여럿일 경우 바로 앞뒤 한명의 Player 정보만 출력하면 된다.

- **Player를 score 기준 오름차순으로 정렬하였을 때**

해당 score 포함 앞과 뒤 Player의 정보를 출력한다.

같은 score가 여럿일 경우 바로 앞뒤 한명의 Player 정보만 출력하면 된다.

\* **프로그램은 5번 과제를 무한히 반복할 수 있어야 한다.**

- **5번 문제를 어떻게 해결하였는지 보고서에 설명하라.**

 **고민한 점:** 다양한 기준으로 정렬된 데이터를 사용하여 빠르게 근처 Player를 탐색할 방법.

 **해결 방법:**

* 정렬된 포인터 배열 3개(sortedById, sortedByName, sortedByScore)를 구성.
* 각 기준별로 std::lower\_bound, std::upper\_bound를 사용하여 같은 값의 범위를 찾고, 앞뒤 요소를 함께 출력.
* 무한 루프를 통해 사용자 입력을 반복 처리.