STL 과제 보고서

폰트, 텍스트, 그래픽, 로고이(가) 표시된 사진

자동 생성된 설명

**2020182006 김경민**

**[문제 1번]**

**[문제1] 마지막 Player 정보**

**이름: hqlmueqrz, 아이디: 883042, 점수: 227999479, 자원수: 348**

**저장된 글자:**

**Mzxtvgdiujkbszixxmvxocmnonawnahxzojtlscorefdapccpmmnyuraztnhopkdhmfbamqlvpbnvffhghydibrypftcogrdmzixdvxdbfebradnbxdfkakegxxdhijcwrsrbdsijledbbqgqizkkgwtgkmhxseidvmtqtfncoaxggrheciacgswkocorzwshzmfwtkskmugusvnyceebtktkcbuhutapgpzjhmqajcninzzfaoetqxdrdqbqggbgsdpddsnmgujfxbzxnmokjlvrswksdzsgtfefxlwlwogfsmiufgiciedeynvfnphpnomkptmziudlhlodbqakehfmyix**

문제 1번을 해결하기 위해서 Player class에 read함수를 선언하여 파일을 불러오면 파일을 읽게 하였습니다.

그 전에 vector로 컨테이너를 만들어서 정보를 저장시켜 두었습니다. 그리고 players.rbegin()->show(); 를 작성하여 마지막 Player정보를 출력하게 하였습니다.

**[문제 2번]**

**[문제2] 가장 점수가 높은 Player 정보**

**이름: waxddgbjaffqv, 아이디: 1021053, 점수: 423456084, 자원수: 281**

**저장된 글자:**

**Rohvirasmibbhpkygumtibeugfnwtbiwdbyiblvkwcgzixpuzwajmuzxzuwrlnwlzwzroictoefuiqeybdzoakabwvktsrgbalpoamyhxrlkqwvdofrmejqkckivopphjnwheuliszljuzisffwspcnbihrwgfpqtiiavexqfnnfvjunxobpepnkhywbyawwrlzqmrmolbzhtvdpvdvwprcwebdqnvnubmanciekqgyxkeudojgrezcfqxvegjpbdnubeleiaohcayeqqljnvgghu**

**평균 점수: 123484768**

문제 2번을 해결하기 위해 for문을 사용하여 최댓값을 찾을수 있지만 그건 하나하나 2’000’000개의 자료를 다 비교해야 하는 작업을 해야기에 매우 비 효율적이다. 그래서 효과적이라고 수업 시간에 배운 max\_element를 사용하여 최대값을 찾게 하였습니다.

평균 점수를 구하기 위해 accumulate를 사용하여 Player 점수들을 다 더하게 하였습니다. 그 더한 값을 player.size() 로 나누어 구하였습니다. 그리고 점수들을 다 더하면 생각보다 값이 클 수 있기 때문에 int보다 더 큰 값을 저장할 수 있는 long long으로 설정하였습니다.

**[문제 3번]**

**[문제3] id가 같은 객체의 개수: 1146626**

일단 파일을 저장할 때 보기 편하게 하기 위해서 id를 오름차순으로 정렬해 주었습니다. 그런 다음 순차적으로 해당id랑 그 다음 id랑 같은 지 검사하고 해당 id랑 그 전의 id랑 같은 지 검사하였습니다.

그리고 같은 id들을 ‘같은아이디.txt’파일로 아이디와 이름만 보내게 하였습니다. 그리고 그 과정을 하나하나 카운트하여 같은 아이디의 객체가 몇 개인지 판별하였습니다.

**[문제 4번]**

**[문제4] 'a'부터 'z'까지 모든 문자가 메모리에 있는 Player의 개수: 1631058**

이 문제는 일단 p가 소유한 메모리로 읽어온 char를 오름차순으로 정렬해 주어야 합니다. sort 함수를 사용하여 오름차순으로 정렬해 주었습니다.

다음으로는 ‘a’부터 ~ ‘z’까지 모든 알파벳이 있으면 하나씩 카운트하여 모든 문자가 있는 Player의 개수를 출력하는 거였습니다. 해결하기 위해 먼저 26개의 bool값이 저장된 vector를 만들고 그 값들을 초기에 false로 만들게 하였습니다. 이렇게 하면 각 알파벳이 나타날 때마다 true로 변경하여 해당 알파벳이 나타났음을 확인할 수 있습니다. 그리고 이제 count\_if를 사용하여 앞에서 만든 vector가 아닌 players vector의 각요소에 대하여 조건을 확인하였습니다. 그리고 true로 바뀌면 found vector에 표시를 합니다. 그럼 found vector에는 값 들은 true/false이렇게 두가지 경우로 정렬이 되어 있을 것 입니다. 이 값을 all\_of를 사용하여 모든 값 들이 true인지 확인하여 모든 알파벳이 있는가 없는가 판별하게 하였습니다. 그리고 모든 값이 true이면 cont값을 하나 하나 늘렸습니다.

**[문제 5번]**

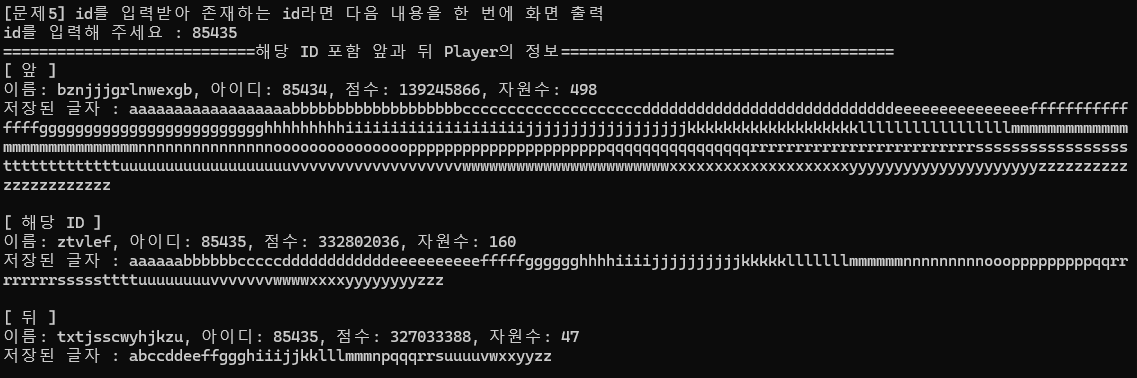
이 문제를 해결하기 위해서 컨테이너 3개를 준비하였습니다.

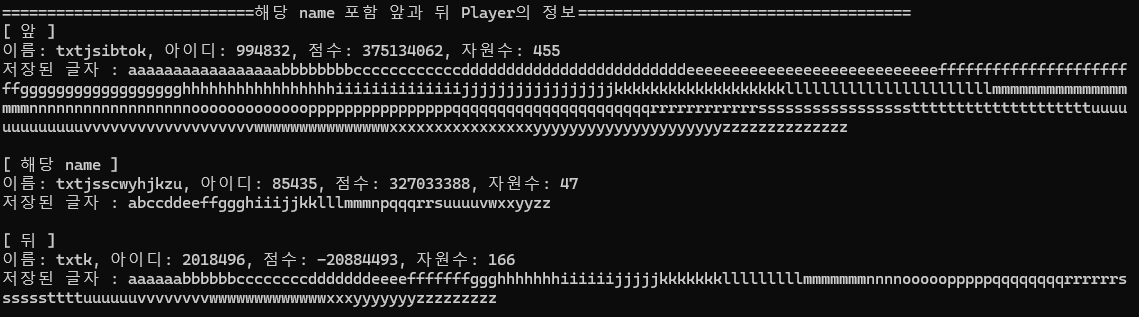
처음에는 vector를 사용하였지만 이번 문제를 해결하기 위해서는 array를 사용하였습니다. array를 사용해도 이 문제는 2’000’000개라는 개수가 정해져 있어서 array를 쓸 때 개수를 딱 맞게 선언하면 공간이 남는 그런 데이터 낭비가 되는 부분이 없다고 생각하였습니다. 따라서 좀더 편리한 array를 사용하였습니다.

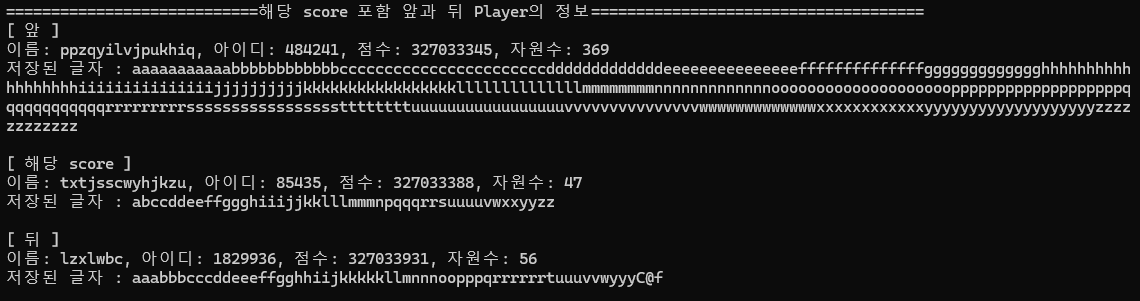
3개를 준비한 이유는 이 문제가 id별로 정렬, name별로 정렬, score별로 정렬 총 3가지로 정렬을 해주고 비교하는 것이기 때문이었습니다. 물론 그때 그때 한 번씩 비교하게 해서 하는 방법이 있지만 그러면 코드가 계속 돌아서 효율적이지 않다고 생각하여(while문이 계속 돌아가기 때문) 각각 따로 정렬해 놓고 그걸 컨테이너에 저장시켜 놓고 그걸 빼서 쓰자 라는 결론을 가졌습니다. 그리고 이제 id를 입력하면 그 입력된 값을 찾는 것은 find\_if를 사용하였습니다. find\_if를 사용하여 나온 값을 하나의 변수(auto infor라 설정하였다.)로 설정해 두고 그 변수를 빼고 더하면서 앞의 값과 뒤의 값을 구하게 하였습니다.

문제 4번을 해결하면서 메모리 p값을 읽어온 char를 오름차순으로 정렬되어 출력됩니다. 다시 처음처럼 출력되게 바꾸고자 하였으나 문제 4번의 오름차순 정렬을 했다는 결과물을 보여주기 위해 그대로 두었습니다.

[문제 5번] <id를 기준으로 하였을 때>



[문제 5번] <name을 기준으로 하였을 때>

[문제 5번] <score를 기준으로 하였을 때>

**[제출한 과제해결 방법이 효율적이라고 주장하는 내용/ 개선할 부분]**

- unique\_ptr을 사용하면 사용자가 복사할당자를 이동 생성할당자들을 따로 만들어 주어야 하는데 이 과정이 수업시간에 배운 내용들을 한 번 더 공부할 수 있어서 좋았지만 귀찮았습니다. 다음부턴 char\*로 하면 좋을 것 같습니다.

- 과제 5번을 할 때 저는 이제 문제를 읽고 문제 그대로 코드를 작성하였습니다. 그런데 이제 좀더 깊게 생각해보면 만약에 name일 때 예시를 들자면 중복id가 있을 때는 그 중복된 id들이 같은 id지만 name이 다르기에 해당name일 때의 앞, 뒤 정보 출력을 따로 다 출력을 해야 한다는 것을 알게 되었습니다. 이 문제를 전 과제 제출 당일 날 수업을 같이 듣는 학생이랑 이야기를 하면서 알게 되었습니다. 만약 이게 맞다 면 다음 문제를 출제하실 때 “중복id가 있을 때는 중복 아이디들에 대한 해당name, score를 다 따로 각각 출력하게 하라” 라는 문구를 하나 추가해 주셨으면 좋겠습니다.

**[과제를 하면서 느낀 점]**

이번 과제를 하면서 수업시간에 배운 내용들을 복습을 하는 느낌을 받아 공부를 더 할 수 있게 되어서 좋았습니다. 저의 부족한 부분이 어떤 곳인지 알게 되었습니다. 과제를 해결하기 위한 여러가지 방법들을 보고 더 무엇이 더 효율적인지 더 완성된 코드인지 비교하면서 좀더 효율적인 코드를 짜는 법을 알게 되었습니다.