자료구조 HW#3

2016706018 박건희

**0.개발환경**



해당 과제의 모든 과정은 Visual Studio 2019에서 진행되었다.

**1. 개요**

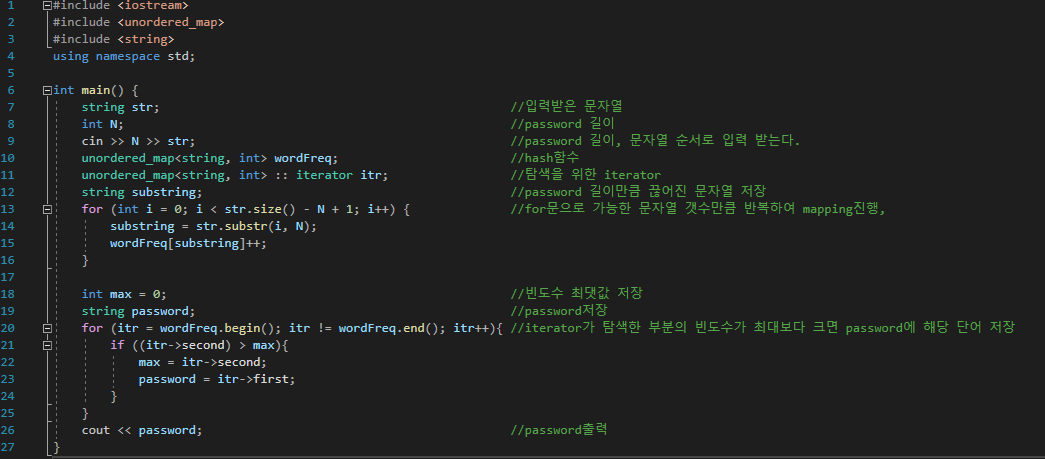
길이가 N인 password를 찾는 과제이다.

입력받은 문자열에서 빈도수가 가장 높은 문자열을 찾아내어 출력한다.

**2. 분석**

hash\_map을 사용하여 password 길이만큼 문자열들을 mapping시키며 빈도수를 카운트하여 저장한 뒤 빈도수를 비교하여 빈도수가 가장 높은 문자열을 출력하면 될 것이다.

**3. 구현**

****

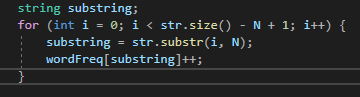
소스코드는 위와 같다.



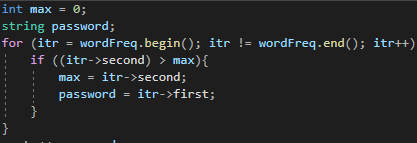
먼저 password 길이, 문자열 순서대로 입력 받기 때문에 각 변수 선언 후에 입력을 받는다.



Password 길이만큼 문자열들을 mapping시키기 위한 hash\_map과 탐색을 위해 iterator를 선언하였다.



이 후 문자열에서 password 길이만큼 substring을 추출한 후 mapping시키며 빈도수를 누적시켰다.



문자열들을 전부 mapping한 이후에는 빈도수가 가장 큰 password를 iterator를 이용하여 탐색하였다.

탐색하면서 만약 빈도수가 최대값이 나오면 password에 해당 substring을 저장하는 방식으로 진행되었다.

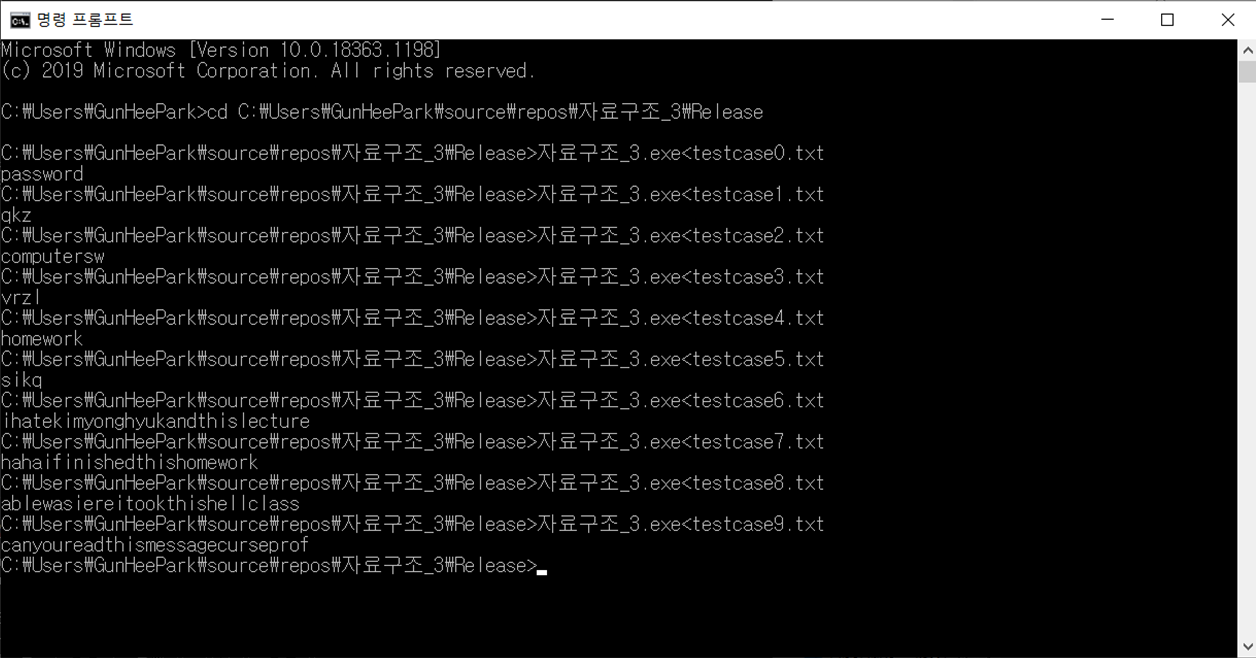


최종적으로 저장된 password를 출력한다.

**4. 테스트**

테스트케이스를 복사하여 붙여 넣으면서 테스트를 진행하였으나 테스트케이스의 크기가 매우 크기 때문에 테스트케이스가 전부 복사가 되지 않았다.

이러한 이유로 cmd창을 이용하여 컴파일 후 생성된 exe실행파일에 cmd창을 이용하여 테스트케이스를 집어넣었다.



실행 결과

Testcase0 : password

Testcase1 : qkz

Testcase2 : computersw

Testcase3 : vrzl

Testcase4 : homework

Testcase5 : sikq

Testcase6 : ihatekimyonghyuckandthislecture

Testcase7 : hahaifinishedthishomework

Testcase8 : ablewasiereitookthishellclass

Testcase9 : canyoureadthismessagecurseprof

위와 같은 password를 찾아냈다.

**5. 고찰**

코딩을 진행하면서 hash함수를 이용하여 간단하게 구현이 가능한 것을 확인하였다. 처음에는 링크드리스트 같은 자료구조에 문자열을 집어넣고 탐색하려 했으나 탐색시 비효율 적이라고 생각하여 hash함수를 배우고나서 진행하였다. 함수를 통해 index를 만들어서 탐색이 단번에 되고, password를 찾는데 코드를 짧게 구현 가능하였다. 다음에 시간이 있다면 다른 자료구조를 이용하여 password를 찾는 코드를 구현하여 실행시간을 비교 해보는것도 괜찮을 것 같다.