

# 3주차 문제 발표

-1181 단어 정렬-

소프트웨어학과 202021006 김예진

# 1181 – 단어 정렬

알파벳 소문자로 이루어진 N개의 단어가 들어오면 아래와 같은 조건에 따라 정렬

- 1) 길이가 짧은 것부터
- 2) 길이가 같으면 사전 순으로

입력

```
13
but
i
wont
hesitate
no
more
no
more
it
cannot
wait
im
yours
```

출력

```
i
im
it
no
but
more
wait
wont
yours
cannot
hesitate
```

# 접근 방법

- 1) 가장 먼저, 입력하고자 하는 단어의 개수와 단어들을 입력해준다.
- 2) String 타입의 벡터를 선언해서 한 단어씩 벡터에 `push_back()`을 해준다.
- 3) `Sort()`를 이용하여 정렬을 해준다.
- 4) 3)번의 과정을 하기 위해 `bool` 타입의 `compare` 함수를 미리 만들어준다.
- 5) 중복을 없애기 위해 `unique()`와 `erase()`를 사용한다.
- 6) 결과 출력.

# 코드 설명

```
2  #include <iostream>
3  #include <vector>
4  #include <algorithm>
```

-> 사용한 헤더파일들

```
7  bool Compare( string a, string b ){
8      if ( a.size() == b.size() )
9          return a < b;
10     else
11         return a.size() < b.size();
12
13 }
```

-> compare 함수

- 만약 두 단어의 길이가 같으면 사전순으로 반환
- 아니라면 짧은 단어 반환

```
15 int main( void ) {
16     vector<string> words;
17     int wordNum; // Number of Testcase
18
19     cin >> wordNum;
20 }
```

-> 입력받을 단어 개수 입력

```
20
21 for ( int i = 0; i < wordNum; i++ ) { wordNum만큼 단어들을 입력받음
22     string temp;
23     cin >> temp;
24     words.push_back(temp);
25 }
```

```
26
27 sort(words.begin(), words.end(), Compare); Compare함수 이용하여 정렬
28 words.erase(unique(words.begin(), words.end()), words.end());
29     -> Unique()를 이용해 중복된 것들은 뒤로 빼고, 정렬 후, erase()로 삭제하여 준다.
```

```
30 int len = words.size();
```

```
31
32 for ( int i = 0; i < len; i++ ){ 결과 출력!
33     cout << words[i] << '\n';
34 }
```

```
35
36 return 0;
```

```
37
38 }
```

감삼당 😎