

# 1주차 문제 발표

-1316 그룹 단어 체커-

소프트웨어학과 202021006 김예진

# 1316 - 그룹 단어 체커

그룹 단어? -> 단어에 존재하는 모든 문자에 대해서, 각 문자가 연속해서 나타남

Ex ) ccaazzzb -> c, a, z, b 가 모두 연속해서 나타남 ( Group Word o )

yejinida -> i 가 떨어져서 나타남 ( Group Word x )

입력 :

- 단어의 개수 N (  $1 \leq N \leq 100$  )
- 단어 입력 ( 알파벳 소문자, 중복 x, 최대 100자 )

출력 :

- 그룹 단어의 개수

# 접근 방법

- 1) 알파벳 수 만큼 26개의 배열을 만들어 모두 0으로 초기화 ( 특정한 알파벳이 나왔었는지 확인하기 위함 )
- 2) 탐색할 각 단어의 길이를 구해준다.
- 3) 그룹 단어인지 확인하기 위해 -> `int isGroup = 1`
- 4) 반복문 안에서 단어의 한 글자씩 탐색하면서, 만약에 그 알파벳 번호에 해당하는 자리가 0 이라면, 그 자리를 1로 바꾸어 준다.
- 5) 하지만, 만약 0이 아니라면, 전에 있는 자리의 문자와 비교하여, 다르다면, 그룹 단어가 아니므로 `isGroup` 변수를 0으로 바꾸어 주고 `break`로 반복문을 탈출한다.
- 6) 반복문이 다 돌고 나서, `isGroup`이 1 이라면, 그룹 단어의 개수를 세는 `cnt` 변수를 1 증가시켜 준다.

# 코드 설명

```
#include <stdio.h>
#include <string.h>
```

-> 사용한 헤더파일들

```
int main(void){
    int N;
    char word[100][101];
    int cnt = 0;

    scanf("%d", &N);
```

-> N : 입력 받을 단어의 개수  
word : 단어  
cnt : 그룹 단어의 개수

N을 입력

```
for ( int i = 0; i < N; i++ ) {
    scanf("%s", word[i]);
    getchar();
}
```

-> N 만큼 단어를 입력 받음

```

for ( int i = 0; i < N; i++ ) {
    char check[26] = {0, };
    int len = strlen(word[i]);
    int isGroup = 1;
    for ( int j = 0; j < len; j++ ) {
        if ( check[word[i][j] - 'a'] == 0 )
            check[word[i][j] - 'a'] = 1;
        else {
            if ( word[i][j - 1] != word[i][j] ){
                isGroup = 0;
                break;
            }
        }
    }

    if ( isGroup == 1 )
        cnt++;
}

```

1) 26개의 배열을 만들어 모두 0으로 초기화

2) 각 단어들의 길이를 구해줌

3) 그룹 단어인지 확인하기 위한 변수

4)

-단어의 j번째 칸에 있는 문자가 이 전에 존재한 적이 없다면, 그 자리에 1을 넣어준다.

-존재한적이 있는데, 앞의 문자와 다르다면 그룹문자가 될 수 없음!

5) 만약 isGroup이 1이면 그룹 단어의 개수를 늘려준다.

```
printf("%d\n", cnt);  
return 0;
```

-> 결과 출력

감삼당 😎