



파일 내부 이름 규칙 통일시키기

- 텍스트 파일들의 이름 규칙은
- "T_이름부위_용도.확장자" 로 구현되어 있다. 파일 탐색기 내부의 모든 파일들을 순회하면서 해당 규칙에 어긋나는 파일들의 이름을 출력하는 프로그램을 작성해보세요.

예시)

 T_이름1_AO.txt
 이름1_Diffuse.txt

2024-11-06 오후 6:45 텍스트 문서
2024-11-06 오후 6:45 텍스트 문서

1. 테스트 폴더 안에 있는 T_이름1_AO는 규칙에 맞게 정리된 내용이다.
2. 아래 규칙의 경우 "이름1" 앞에 T_가 추가되어야 한다.

주의 사항

- 공백 문자도 문자로 인식한다. 이를 어떻게 해결할 것인가?
- 위 알고리즘은 운영체제의 파일 탐색기에서 실행되어야 한다. IO System으로 파일 탐색기에 파일 추가 하는 방법을 복습 또는 공부해볼 것
- 어렵게 알고리즘을 작성하는 대신 더 어렵지만 간단한 **정규식(Regular Expression)**이 존재한다. 어렵지만 한번 배우면 정말 강력한 기능을 하기 때문에 프로그래머라면 시도 해보는 것을 추천한다.

$^{\wedge}(.*)$: 첫번째 구분자 이전의 모든 문자열 캡처
T1 : 파일 이름 맨 앞에 추가될 텍스트는 T이다.

예시 코드

```
#include <iostream>
#include <regex>
#include <string>

std::string regexReplace(const std::string& input, const std::string& pattern,
const std::string& replacement) {
    std::regex regexPattern(pattern); // 정규식 패턴을 사용해 정규식 객체 생성
    return std::regex_replace(input, regexPattern, replacement); // 정규식을 이
용한 치환
}
```

```
int main() {
    std::string input = "Claws_AlbedoTransparency";
    std::string pattern = R"^(.*?)_"; // 정규식 패턴
    std::string replacement = "T_$1_"; // 치환할 텍스트 (프리픽스 T를 추가)

    std::string result = regexReplace(input, pattern, replacement);
    std::cout << "Result: " << result << std::endl;

    return 0;
}
```