

과제

장유선

2023.09.01

1. 문제 정의

전달해 드리는 코드에서 소,중,대 괄호가 정상인지 확인하고, 정상이 아닌 경우에는 쌍이 맞지 않는 괄호가 시작하는 위치에 대한 Line 및 Position을 출력하세요

입력값

첨부파일로 추가된 cpp 파일 3개

결과값

1. Error (line 3, col 16)
2. Error (line 22, col 28)
3. Pass

2. 입력값

2-1. test1.cpp

```
int main() {
    const int arraySize = 20;
    int numbers[arraySize];

    // 난수를 생성하여 배열에 저장
    std::srand(static_cast<unsigned>(std::time(nullptr)));
    for (int i = 0; i < arraySize; ++i) {
        numbers[i] = std::rand() % 100;
    }

    // 배열 요소 출력
    std::cout << "배열 요소: ";
    for (int i = 0; i < arraySize; ++i) {
        std::cout << numbers[i] << " ";
    }
    std::cout << std::endl;

    // 최대값 찾기
    int maxNumber = findMax(numbers, arraySize);
```

```

    std::cout << "최대값: " << maxNumber << std::endl;

    return 0;
}

```

2-2. test2.cpp

```

#include <iostream>
#include <fstream>
#include <sstream>
#include <map>
#include <string>

int main() {
    std::string filename = "sample.txt"; // 분석할 파일 이름
    std::ifstream inputFile(filename);

    if (!inputFile.is_open()) {
        std::cerr << "파일을 열 수 없습니다." << std::endl;
        return 1;
    }

    std::map<std::string, int> wordFrequency;

    std::string line;
    while (std::getline(inputFile, line)) {
        std::istringstream iss(line);
        std::string word;
        while (iss >> word) { // 여기서 괄호 위치 오류
            // 단어 처리: 구두점 제거 등
            // 여기서는 단순히 모든 문자를 소문자로 변환
            for (char &c : word) {
                c = std::tolower(c);
            }
            wordFrequency[word]++;
        }
    }

    inputFile.close();

    std::cout << "파일 내 단어 빈도:" << std::endl;
    for (const auto &pair : wordFrequency) {
        std::cout << pair.first << ": " << pair.second << std::endl;
    }

    return 0;
}

```

2-3. test3.cpp

```
#include <iostream>
#include <string>
#include <vector>

class Student {
public:
    Student(const std::string& name, int age) : name(name), age(age) {}

    void displayInfo() const {
        std::cout << "이름: " << name << ", 나이: " << age << std::endl;
    }

private:
    std::string name;
    int age;
};

int main() {
    std::vector<Student> students;

    while (true) {
        std::cout << "1. 학생 정보 입력, 2. 전체 학생 목록 출력, 3. 종료: ";
        int choice;
        std::cin >> choice;

        if (choice == 1) {
            std::string name;
            int age;

            std::cout << "이름 입력: ";
            std::cin >> name;
            std::cout << "나이 입력: ";
            std::cin >> age;

            students.emplace_back(name, age);
            std::cout << "학생 정보가 입력되었습니다." << std::endl;
        } else if (choice == 2) {
            std::cout << "전체 학생 목록:" << std::endl;
            for (const Student& student : students) {
                student.displayInfo();
            }
        } else if (choice == 3) {
            std::cout << "프로그램을 종료합니다." << std::endl;
            break;
        } else {
            std::cout << "잘못된 선택입니다. 다시 선택하세요." << std::endl;
        }
    }
}
```

```

    }
}

return 0;
}

```

3. Python Code Hard Copy

```

def check(file_path):
    opening = "([{"
    closing = ")]}"
    stack = []
    line_number = 0
    col_number = 0

    try:
        with open(file_path, 'r', encoding='utf-8') as file:
            for line in file:
                line_number += 1
                col_number = 0
                for char in line:
                    col_number += 1
                    if char in opening:
                        stack.append((char, line_number, col_number))
                    elif char in closing:
                        if not stack:
                            print(f"Error (line {line_number}, col
{col_number}))")
                            return
                        top = stack.pop()
                        if top[0] == '(' and char != ')':
                            print(f"Error (line {top[1]}, col {top[2]})")
                            return
                        if top[0] == '[' and char != ']':
                            print(f"Error (line {top[1]}, col {top[2]})")
                            return
                        if top[0] == '{' and char != '}':
                            print(f"Error (line {top[1]}, col {top[2]})")
                            return

                        top = []

    except FileNotFoundError:
        print(f"File not found: {file_path}")

```

```

    while stack:
        top = stack.pop()
        print(f"Error (line {top[1]}, col {top[2]}): Unmatched opening
bracket '{top[0]}')

    if not stack:
        print("Pass")

test1_path = "C:\\Users\\inavi\\Desktop\\장유선\\개발교육\\알고리즘\\괄호
검사\\test1.cpp"
test2_path = "C:\\Users\\inavi\\Desktop\\장유선\\개발교육\\알고리즘\\괄호
검사\\test2.cpp"
test3_path = "C:\\Users\\inavi\\Desktop\\장유선\\개발교육\\알고리즘\\괄호
검사\\test3.cpp"

print("Test 1:")
check(test1_path)

print("\nTest 2:")
check(test2_path)

print("\nTest 3:")
check(test3_path)

```

4. Code 설명

4-1. 변수 설정

```

opening = "([{"
closing = ")]}"
stack = []
line_number = 0
col_number = 0

```

- 여는 괄호, 닫는 괄호를 문자열로 지정해준다.
- 괄호를 저장하기 위한 스택을 설정한다.
- 위치를 확인하기 위해 라인과 열 번호를 0으로 초기화한다.

4-2. 파일 읽기, 여는 괄호를 읽을 경우

```
with open(file_path, 'r', encoding='utf-8') as file:
    for line in file:
        line_number += 1
        col_number = 0
        for char in line:
            col_number += 1
            if char in opening:
                stack.append((char, line_number, col_number))
            elif char in closing:
                if not stack:
                    print(f"Error (line {line_number}, col {col_number})")
                    return
                top = stack.pop()
```

- 라인, 열 순서로 char을 읽는다.
- 여는 괄호일 경우 스택에 위치와 함께 추가한다.

4-3. 닫는 괄호를 읽을 경우

```
if char in closing:
    if not stack:
        print(f"Error (line {line_number}, col {col_number})")
        return
    top = stack.pop()

    if top[0] == '(' and char != ')':
        print(f"Error (line {top[1]}, col {top[2]})")
        return
    if top[0] == '[' and char != ']':
        print(f"Error (line {top[1]}, col {top[2]})")
        return
    if top[0] == '{' and char != '}':
        print(f"Error (line {top[1]}, col {top[2]})")
        return

    top = []
```

- 여는 괄호가 없는데 닫는 괄호가 나온 경우 -> 에러
- 괄호 쌍이 일치하지 않는 경우 -> 에러

5. 결과

Test 1:

Error (line 3, col 16)

Test 2:

Error (line 19, col 43)

Test 3:

Pass

6. 결과 화면

```
Test 1:  
Error (line 3, col 16)  
  
Test 2:  
Error (line 19, col 43)  
  
Test 3:  
Pass
```