

MARG Python Study Assignment 5

Question 1. Text Analyzer

주어진 텍스트 파일을 분석하는 프로그램을 작성하세요. 분석해야 할 내용은 글자 수, 단어 수, 문장당 평균 글자 수, 그리고 등장하는 단어 별 빈도수입니다.

글자 수와 문장당 평균 글자 수를 셀 때는 공백과 문장 기호(마침표, 쉼표, 콜론, 세미콜론)를 무시해야 합니다. 예를 들어 다음과 같은 텍스트가 있다고 하면:

```
Hello, world; I am Kim. I am twenty years old.
```

글자 수는 총 33 개이고, 총 2 문장이 있으므로 문장 당 글자 수는 16.5 개이며, 단어 수는 10 개입니다. 나아가 단어 별 빈도수는 대소문자를 구분하지 않고 계산해야 하며 알파벳 순으로 출력해야 합니다. 따라서 종합적으로 다음과 같이 출력해야 합니다:

```
Character Count: 33
Word Count: 10
Average Sentence Length: 16.5 characters
Word Frequency:
am: 2
hello: 1
i: 2
kim: 1
old: 1
twenty: 1
world: 1
years: 1
```

주어진 세 개의 텍스트 파일(1.txt, 2.txt, 3.txt)를 활용해 프로그램을 완성하세요. 입력으로는 파일 이름이 주어지며, 텍스트의 분석 결과를 출력해야 합니다. 아래 예시를 참고하세요.

입력 파일 내용 (3.txt):

A well organized paragraph supports or develops a single controlling idea, which is expressed in a sentence called the topic sentence. A topic sentence has several important functions: it substantiates or supports the thesis statement; it unifies the content of a paragraph and directs the order of the sentences; and it advises the reader of the subject to be discussed and how the paragraph will discuss it. Readers generally look to the first few sentences in a paragraph to determine the subject and perspective of the paragraph. That is why it is often best to put the topic sentence at the very beginning of the paragraph. In some cases, however, it is more effective to place another sentence before the topic sentence, for example, a sentence linking the current paragraph to the previous one, or one providing background information.

입력 예시:

3.txt

출력 예시:

Character Count: 690

Word Count: 139

Average Sentence Length: 138.0 characters

Word Frequency:

a: 7

advises: 1

and: 4

another: 1

.

.

.

the: 17

thesis: 1

to: 6

topic: 4

unifies: 1

very: 1

well: 1

which: 1

why: 1

will: 1

Question 2. Python Interpreter

파이썬 코드를 이해하는 프로그램을 작성하세요. 프로그램은 유효한 파이썬 코드를 한 줄씩 입력 받고 적절한 출력을 해야 합니다. 이 프로그램이 해석할 수 있는 파이썬 함수는 print()와 quit()이며 string 변수와 '+' 연산자를 사용할 수 있습니다.

공백은 코드의 실행에 영향을 주지 않아야 합니다. 즉,

```
print()  
print    (  )  
    print (  )
```

위의 코드는 전부 동일하게 해석되어야 합니다. 나아가 print 함수는 변수만 입력 받을 수 있고, string 을 직접적으로 받을 수는 없습니다.

```
a = 'hello, world!'  
print(a)
```

위의 코드는 정상적으로 작동하나,

```
print('hello, world!')
```

위의 코드는 문법 에러입니다. 변수는 오로지 string 값만 저장할 수 있으며 string 은 작은따옴표 안에 들어가야 합니다.

```
random_variable = 'hello, world!'
```

위의 코드는 정상적인 변수 할당이지만,

```
random_variable = 4  
random_variable = "hello world"  
random_variable = 'hello'world
```

위의 코드는 에러입니다.

마지막으로 print() 함수는 변수 사이의 + 연산이 가능합니다. 더 자세한 입출력은 아래 예시를 참고하세요. 설명되지 않았거나 아래 예시에 나오지 않는 예외 사항은 고려하지 않아도 됩니다.

입출력 예시: (입력은 파란색, 출력은 검은색)

```
>> greeting = 'Hi, how are you'
>> my_name = 'Kim'
Error: undefined syntax
>> my_name = Kim'
Error: undefined syntax
>> my_name = + 'Kim'
Error: undefined syntax
>> my_name = 'Kim'
>> white_space = ' '
>> print ( greeting + white_space + my_name )
Hi, how are you Kim
>> your_name = 'Lee'____
Error: undefined syntax
>> your_name = 'Lee'
>> print(greeting + white_space + your_name)
Hi, how are you Lee
>> =verb = 'likes'
Error: undefined syntax
>> verb = 'likes'
>> print(your_name + whitespace + verb ++ whitespace + my_name)
Error: undefined syntax
>> print(+ whitespace + verb + whitespace + my_name)
Error: undefined syntax
>> print(your_name + whitespace + verb + whitespace + my_name)
Lee likes Kim
>> print(my_name)
Kim
```

```
>> print(your_name)
Lee
>> qiut()
Error: undefined syntax
>> quit(
Error: undefined syntax
>> quit
Error: undefined syntax
>> quit)
Error: undefined syntax
>> quit('hello, world!')
Error: undefined syntax
>> quit()
```