프로그래밍 언어 Lex

<과제 2\_3>

1. 문제  
   지환이는 프로그래밍 언어론 시간에 lex 프로그램에 대해 배웠다. Lex 프로그램은 자체적으로 원하는 부분의 언어를 분석하여 읽는 기능을 가지고 있으며 컴퓨터공학과인 지환이는 이 기능을 이용하여 어떤 부분을 읽어 들었는지를 파악하고, 읽어 들인 부분을 카운트 하여 출력하는 프로그램을 만들어 친구들에게 자랑하고자 한다.
2. 프로그램 작동 순서
   1. Lex 프로그램을 실행 시 C 코드를 읽어오게 된다.  
      ./hw2\_3 < test.c
   2. 코드의 내용을 분석한다.
   3. 각 코드에 나오는 특성에 따라서 카운트 한다.
      1. Preprocessor : 전처리문의 개수 ( #include, #define 만)
      2. octal number : 8진법 숫자 개수
      3. negative decimal number : 10진법 숫자 중 음수의 개수
      4. positive decimal number : 10진법 숫자 중 양수의 개수
      5. operator : 아래에 명시된 연산자의 개수
         1. 연산자의 종류
            1. 산술 연산자 +, -, \*, /, %
            2. 논리 연산자 <, <=, ==, >, !=
            3. 관계 연산자 ||, &&, !
            4. 증감 연산자 ++, --
            5. 콤마 연산자 ,
            6. 참조 연산자 . , ->
            7. 포인터 연산자 &, \*
      6. comment : 주석문의 개수
         1. 주석문의 종류
            1. //\_
            2. /\*\_\*/
            3. 주석문 안의 값은 count 되지 않는다.
            4. A,B의 \_는 임의의 단어들의 반복 (0회 이상).
      7. ‘=’ : = 기호의 개수 ( 대입 연산자 )
      8. ‘{‘ : { 기호 개수
      9. ‘}’ : } 기호 개수
      10. wordcase1 : p가 두 번만: 들어간 단어의 개수 apple
      11. wordcase2 : e로 시작하고 마지막 글자가 m인 단어의 개수
      12. word : 그 외 단어의 개수
      13. mark : 위에서 count 되지 않은 문자의 개수
   4. 카운트 된 값을 화면에 출력한다.
3. 입출력 예시
   1. 입력 : test.c ( 제공된 파일 )
   2. 출력 내용  
      텍스트이(가) 표시된 사진

      자동 생성된 설명
4. 보고서  
   구현한 부분에 대한 코드 분석 및 사용 이유 등의 설명을 자세히 문서화 한다.
   1. Lex에 대한 설명 ( 동작 방식 )
   2. 구현 내용에 대한 정리 및 자세한 코드 설명(Subsection으로 구분)
   3. Latex로 작성하며, tex 파일과 pdf 파일 모두 제출
5. Q & A
   1. 채점 기준
      1. Lex에 대해 제대로 이해하였는가? (Latex 중요)
      2. 주의사항
         1. 주석문 내부 값은 count하지 않음.
         2. %d의 경우 %는 연산자, d는 word로 count
         3. 테스트 케이스의 무결성은 보장됨.
         4. Output이 제시된 Output과 다르면 0점.
   2. 감점 사항  
      Cheating 발견 시 관련 학생 모두 0점 처리.
   3. 질문  
      이메일을 통한 질문 : [pemta806@gmail.com](mailto:pemta806@gmail.com)(메일제목 앞에 [Lex] 추가)
   4. 유의사항

Late 제출, 메일 제출은 받지 않습니다.

Cheating 시 무조건 F