

과목명	컴파일러
담당교수	강승식
보고서 ID	HW03

학과	컴퓨터공학부
학번	20123368
이름	박희상

1. 아래 정규표현식들을 인식하는 lex 입력을 작성하시오. 액션코드는 아래 설명을 출력하시오.

```
kmucs@localhost:~$ ./lex1.exe
Computer
대문자로 시작해서 소문자로 이루어진 단어

[EXAMPLE]
[와 ] 사이에 대문자로만 구성된 단어

1011000
부호없는 이진수로 4의 배수인 수

0B00101
0B로 시작되는 2진수
```

```
%%
[A-Z][a-z]+    { puts("대문자로 시작해서 소문자로 이루어진 단어\n"); }
\[[A-Z]+\]     { puts("[와 ] 사이에 대문자로만 구성된 단어\n"); }
[0-1]+00       { puts("부호없는 이진수로 4의 배수인 수\n"); }
0B[0-1]+       { puts("0B로 시작되는 2진수\n"); }
```

- Computer 와 같이 대문자로 시작해서 소문자로 이루어진 단어
- [와] 사이에 대문자로만 구성된 단어
- 부호없는 이진수로 4 의 배수인 수
- 0B 로 시작되는 2 진수

```
0001234567  
00로 시작되는 8진수
```

```
0X01234ABCD56789EF  
0X로 시작되는 16진수
```

```
" C language string constant! "  
c언어의 스트링 상수
```

```
// random comment!!@#$  
//로 시작하여 라인끝까지 코멘트 문장
```

```
/* one line comment ! */  
/*와 */ 사이에 있는 코멘트 문장
```

```
00[0-7]+      { puts("00로 시작되는 8진수\n"); }  
0X[A-F0-9]+   { puts("0X로 시작되는 16진수\n"); }  
\".*\" { puts("c언어의 스트링 상수\n"); }  
\\/\\/.* { puts("//로 시작하여 라인끝까지 코멘트 문장\n"); }  
\\/\\*.*\\*\\/   { puts("/*와 */ 사이에 있는 코멘트 문장\n"); }  
.;  
%%
```

- 00 로 시작되는 8 진수
- 0X 로 시작되는 16 진수
- C 언어의 스트링 상수
- //로 시작하여 라인끝까지 코멘트 문장
- /*와 */ 사이에 있는 코멘트 문장

2. 아래 내용의 lex 입력을 작성하시오.

```
kmucs@localhost:~$ ./lex2.exe
AAA  AA
AAA AA

A      A
A A

64
십진법 64
이진법 1000000
```

- 여러 개의 공백문자를 1 개로 축약
- tab 문자는 공백문자로 치환
- 10 진수 또는 16 진수를 2 진수로 변환하여 출력 -- 함수 dtob() 사용

```
%%
[ ]+          { printf(" "); }
[\\tab]+      { printf(" "); }
[0-9]+        { printf("십진법 %d\\n이진법 %d\\n",atoi(yytext),dtob(atoi(yytext))); }
```

```
abcd
abCD

abcabc
ABCABC
```

- abcd 의 cd 를 CD 로 치환
- 소문자를 모두 대문자로 치환

```
abcd          { puts("abCD"); }
[a-z]+        { for(int i=0;i<strlen(yytext);i++){
                  *(yytext+i) = toupper(*(yytext+i));
                }
                printf("%s",yytext); }
```

3. 아래 내용의 lex 입력을 작성하시오.

```
kmucs@localhost:~$ ./lex3.exe
Count Example One Two Three

/* example abc /* kk */
example abc kk

{      a b c d }
begin  a b c d end
word_count = 5
/*    = 2
*/    = 1
Count is not same
```

- 첫문자가 대문자이고 나머지는 모두 소문자인 단어의 개수를 count 하여 출력
- /*의 개수와 */의 개수를 count 하고 개수가 같은지 비교 결과를 출력
- {는 begin 으로, }는 end 로 치환하여 출력
- 줄바꿈 문자를 모두 공백문자로 치환

```
%%
[A-Z][a-z]+      word_count++;
\\/*             open_count++;
\\*/             close_count++;
"{"              { printf("begin"); }
"}"              { printf("end"); }
"\n"             { printf(" "); }
%%

main()
{

    yylex();
    printf("word_count = %d\n", word_count);
    printf(" /*    = %d\n", open_count);
    printf(" */    = %d\n", close_count);

    if(open_count == close_count)
        printf("Count is same \n");
    else
        printf("Count is not same \n");

    return 0;
}
```

Ctrl+D 를 입력시 입력이 종료되고 카운트한 숫자들을 화면에 출력한다.