

코드 & 설명

```
%{
#include <stdio.h>
%}

%%
```

```
"#define"      printf("DEFINE ");
"int"          printf("INT ");
"void"         printf("VOID ");
"if"           printf("IF ");
"else"         printf("ELSE ");
"for"          printf("FOR ");
```

#define, int, void, if, else, for라는 token을 각각 받으면 그에 맞게 대문자로 출력하도록 직접 printf

```
"=" | "+="      printf("OP_ASSIGN ");
```

= 또는 +=을 받으면 OP_ASSIGN으로 출력하도록 직접 printf

```
"++"          printf("OP_INC ");
"+"           printf("OP_ADD ");
"-"           printf("OP_SUB ");
"*"           printf("OP_MUL ");
"/"           printf("OP_DIV ");
```

각각의 기호에 맞춰 직접 출력하도록 printf

```
"&&" | "||"     printf("OP_LOGIC ");
```

&& 또는 ||을 받으면 OP_LOGIC으로 출력하도록 직접 printf

```
"!=" | "==" | "<" | ">" | "<=" | ">="      printf("OP_REL ");
```

!=, ==, <, >, <=, >=을 받으면 OP_REL으로 출력하도록 직접 printf

```
[_a-zA-Z][_a-zA-Z0-9]*      printf("ID ");
```

ID는 알파벳(a-zA-Z), 숫자(0-9), 언더바(_)로 이루어질 수 있다.

그러나 숫자로 시작할 수는 없기 때문에 시작 문자는 언더바 또는 알파벳, 그 이후부터는 알파벳, 숫자, 언더바가 있을 수도 없을 수도 있기에 *를 이용했다.

```
[0-9]+          printf("NUM ");
```

숫자가 하나 이상 있다면 NUM으로 출력하도록 직접 printf

```
"//".* | "/*".**"/"      printf("comment ");
```

// 주석 또는 /* 주석 */일 경우 comment로 출력하도록 직접 printf

// 다음에 오거나 /* */ 사이에 오는 모든 문자는 comment로 처리하기 위해 .*을 사용

```
"["|"]|"("|")|"{"|"}|";|",|"    printf("%s ", yytext);
```

[], (), { }, ; , ,는 입력이 되면 그대로 출력하기 위해 yytext를 이용. 각각의 token이 yytext에 저장 되기 때문에 yytext를 출력하면 그대로 나온다.

```
[ \t ]+
```

공백(whitespace)또는 탭(wt)이 하나 이상 있다면 무시.

```
%%

int main(argc, argv)
int argc;
char **argv;
{
    ++argv, --argc;
    if( argc > 0 )
        yyin = fopen(argv[0], "r");
    else
        yyin = stdin;
    yylex();
    return 0;
}
```

실행 할 때 뒤에 파일 이름을 적으면 그 파일을 scan, 아무것도 적지 않으면 콘솔에서 input을 직접 받는다.