<컴파일러_과제2_1415040오수민_1415001무슬리마트>

1. Lex 프로그램 파일 (scanner_prac.l)

```
%{
/*
 * hw2_scanner.l - lexical analyzer for hw#2
 *
 * Programmer - Soomin, Muslimat
 *
 * date - 4/7/2017
 *
 * modification history
 *
*/

#include <stdio.h>
#include <stdlib.h>
#include "tn.h"
#include "glob.h"
#include "SymbolTable.h"
#include "ReportError.h"

%}

%%
"const"                 return(TCONST);
"else"                  return(TELSE);
"if"                    return(TIF);
"int"                   return(TINT);
"return"                return(TRETURN);
"void"                  return(TVOID);
"while"                 return(TWHILE);
"+"                     return(TPLUS);
"-"                     return(TMINUS);
"*"                     return(TSTAR);
"/"                     return(TSLASH);
"%"                     return(TMOD);
"="                     return(TASSIGN);
"+="                    return(TADDASSIGN);
"-="                    return(TSUBASSIGN);
"*="                    return(TMULASSIGN);
"/="                    return(TDIVASSIGN);
"%="                    return(TMODASSIGN);
"!"                     return(TNOT);
"&&"                    return(TAND);
"||"                    return(TOR);
"=="                    return(TEQUAL);
"!="                    return(TNOTEQU);
"<"                     return(TGREAT);
```

```
">"                                     return(TLESS);
"<="                                    return(TGREATE);
">="                                    return(TLESSE);
"++"                                    return(TINC);
"--"                                    return(TDEC);
"("                                     return(TOPEN);
")"                                     return(TCLOSE);
","                                     return(TCOMMA);
"{"                                     return(TBIGOPEN);
"}"                                     return(TBIGCLOSE);
"["                                     return(TOPENBRACKET);
"]"                                     return(TCLOSEBRACKET);
";"                                     return(TSEMICOLON);

[A-Za-z_][A-Za-z0-9_]*                  { SymbolTable(); }

[1-9][0-9]*                             return(TNUMBER);
[0-9]+"."[0-9]+(e[+-]?[0-9]+)?          return(TREALNUMBER);
₩"[^"]*₩"                               return(TSTRING);
"/*"([^*]|₩*+[^*/])*₩**"*/"             ;
"//".*                                  ;
[ ₩t]                                   ;
₩n                                      LineNumber++;
.                                       { ReportError(2); }
%%

void printtoken(enum tokennumber tn){
 switch(tn){
        case TCONST                     : printf("%d      TCONST                  const₩n", LineNumber); break;
        case TELSE                      : printf("%d      TELSE                   else₩n", LineNumber); break;
        case TIF                : printf("%d      TIF                     if₩n", LineNumber); break;
        case TINT               : printf("%d      TINT                    int₩n", LineNumber); break;
        case TRETURN                    : printf("%d      TRETURN                 return₩n", LineNumber); break;
        case TVOID                      : printf("%d      TVOID                   void₩n", LineNumber); break;
        case TWHILE                     : printf("%d      TWHILE                  while₩n", LineNumber); break;
        case TPLUS                      : printf("%d      TPLUS                   +₩n", LineNumber); break;
        case TMINUS                     : printf("%d      TMINUS                  -₩n", LineNumber); break;
        case TSTAR                      : printf("%d      TSTAR                   *₩n", LineNumber); break;
        case TSLASH                     : printf("%d      TSLASH                  /₩n", LineNumber); break;
        case TMOD                       : printf("%d      TMOD                    %₩n", LineNumber); break;
        case TASSIGN                    : printf("%d      TASSIGN                 =₩n", LineNumber); break;
        case TADDASSIGN                 : printf("%d      TADDASSIGN              +=₩n", LineNumber); break;
        case TSUBASSIGN                 : printf("%d      TSUBASSIGN              -=₩n", LineNumber); break;
        case TMULASSIGN                 : printf("%d      TMULASSIGN              *=₩n", LineNumber); break;
        case TDIVASSIGN                 : printf("%d      TDIVASSIGN              /=₩n", LineNumber); break;
        case TMODASSIGN                 : printf("%d      TMODASSIGN              %=₩n", LineNumber); break;
        case TNOT                       : printf("%d      TNOT                    !₩n", LineNumber); break;
        case TAND                       : printf("%d      TAND                    &&₩n", LineNumber); break;
        case TOR                : printf("%d      TOR                     ||₩n", LineNumber); break;
        case TEQUAL                     : printf("%d      TEQUAL                  ==₩n", LineNumber); break;
```

```
            case TNOTEQU                    : printf("%d        TNOTEQU                    !=₩n",  LineNumber);
break;
            case TGREAT                     : printf("%d        TGREAT                     <₩n", LineNumber); break;
            case TLESS                      : printf("%d        TLESS                      >₩n", LineNumber); break;
            case TGREATE                    : printf("%d        TGREATE                    <=₩n", LineNumber); break;
            case TLESSE                     : printf("%d        TLESSE                     >=₩n", LineNumber); break;
            case TINC                       : printf("%d        TINC                       ++₩n", LineNumber); break;
            case TDEC                       : printf("%d        TDEC                       --₩n", LineNumber); break;
            case TOPEN                      : printf("%d        TOPEN                      (₩n", LineNumber); break;
            case TCLOSE                     : printf("%d        TCLOSE                     )₩n", LineNumber); break;
            case TCOMMA                     : printf("%d        TCOMMA                     ,₩n", LineNumber); break;
            case TBIGOPEN                   : printf("%d        TBIGOPEN                   {₩n", LineNumber); break;
            case TBIGCLOSE                  : printf("%d        TBIGCLOSE                  }₩n", LineNumber); break;
            case TOPENBRACKET               : printf("%d        TOPENBRACKET               [₩n", LineNumber); break;
            case TCLOSEBRACKET              : printf("%d        TCLOSEBRACKET              ]₩n", LineNumber); break;
            case TSEMICOLON                 : printf("%d        TSEMICOLON                 ;₩n", LineNumber); break;
            case TNUMBER                    : printf("%d        TNUMBER                    %s₩n",  LineNumber,
yytext); break;
            case TREALNUMBER                : printf("%d        TREALNUMBER                %s₩n",  LineNumber,  yytext);
break;
            case TSTRING                    : printf("%d        TSTRING                    %s₩n",  LineNumber,  yytext);
break;
            case TIDENT                     : break;
            case TERROR                     : break;
 }
}

void main(){
 enum tokennumber tn;
 printf("₩nStart of LEX!!!₩n₩n");

 printf("Line number          Token type          ST-index  Token₩n");

 while((tn = yylex()) != TEOF){
  printtoken(tn);
 }

 if(cErrors == 0){
    printf("No errors detected.₩n");
 }
 else{
    printf("%d errors detected.₩n", cErrors);
 }
}

int yywrap(){
 printf("₩n₩nEnd of LEX!!!₩n");
 return 1;
}
```

2. 모든 헤더파일(glob.h / tn.h / ReportError.h / SymbolTable.h)

-glob.h-

```
/*모든 c프로그램에서 공통으로 사용될 전역변수이다.*/

int LineNumber; //yytext가 입력파일의 몇 번째 줄에 있는지 출력하기 위한 라인번호 변수
int cErrors; //해당 입력파일에 에러개수가 몇개 인지 출력하기 위한 변수
```

-tn.h-

```
enum tokennumber { TEOF, TCONST, TELSE, TIF, TINT, TRETURN, TVOID, TWHILE, TPLUS, TMINUS, TSTAR, TSLASH, TMOD, TASSIGN, TADDASSIGN, TSUBASSIGN, TMULASSIGN, TDIVASSIGN, TMODASSIGN, TNOT, TAND, TOR, TEQUAL, TNOTEQU, TGREAT, TLESS, TGREATE, TLESSE, TINC, TDEC, TOPEN, TCLOSE, TCOMMA, TBIGOPEN, TBIGCLOSE, TOPENBRACKET, TCLOSEBRACKET, TSEMICOLON, TIDENT, TNUMBER, TREALNUMBER, TSTRING, TERROR };
```

-ReportError.h-

```
/*ReportError.c에서 사용할 함수 정의*/

#include <stdio.h>
#include <stdlib.h>

void ReportError(int error);
```

-SymbolTable.h-

```
/*SymbolTable.c에서 사용할 함수 및 변수 정의*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define STsize 1000
//#define STsize 30
#define HTsize 100

#define isLetter(x) ( ((x) >= 'a' && (x) <= 'z') || ((x) >= 'A' && (x) <= 'Z') || (x) == '_')
#define isDigit(x) ( (x) >= '0' && (x) <= '9' )

typedef struct HTentry *HTpointer;
typedef struct HTentry {
        int index;
        HTpointer next;
}HTentry;

HTpointer HT[HTsize];
char ST[STsize];

static int nextid = 0; //현재 identifier 위치
static int nextfree = 0; //다음 identifier가 올 위치 인덱스
static int hashcode; //identifier가 가질 해시코드
static int sameid; //이미 있는 identifier일 경우 첫 identifier의 인덱스
```

```
static int found = 0; //이미 있는지 없는지 판단할 때 쓰이는 변수(있다면 1, 처음 온 것이라면 0)
static int error = 0; //각각의 에러를 판단할 때 쓰이는 변수



//SymbolTable.c 함수들
void SkipSeperators();
void ReadID();
void ComputeHS(int nid, int nfree);
void LookupHS(int nid, int hscode);
void ADDHT(int hscode);
void PrintHStable();
void SymbolTable();
```

3. 모든 c프로그램 파일(main.c / ReportError.c / SymbolTable/c)

-main.c-

```
#include <stdio.h>
#include <stdlib.h>
#include "tn.h"
#include "glob.h"
#include "SymbolTable.h"
#include "ReportError.h"

extern int yylex();
extern char *yytext;

//tn의 케이스에 맞게 출력해주는 함수
void printtoken(enum tokennumber tn){
 switch(tn){ //전역변수 LineNumber는 0으로 초기화 되어있기 때문에 1씩 더해준 후 출력해준다.
        case TCONST                     : printf("%d                TCONST                          const₩n",
LineNumber+1); break;
        case TELSE                      : printf("%d                TELSE                           else₩n",
LineNumber+1); break;
        case TIF                : printf("%d                TIF                             if₩n",
LineNumber+1); break;
        case TINT               : printf("%d                TINT                            int₩n",
LineNumber+1); break;
        case TRETURN            : printf("%d                TRETURN                         return₩n",
LineNumber+1); break;
        case TVOID                      : printf("%d                TVOID                           void₩n",
LineNumber+1); break;
        case TWHILE                     : printf("%d                TWHILE                          while₩n",
LineNumber+1); break;
        case TPLUS                      : printf("%d                TPLUS                           +₩n",
LineNumber+1); break;
        case TMINUS                     : printf("%d                TMINUS                          -₩n",
LineNumber+1); break;
        case TSTAR                      : printf("%d                TSTAR                           *₩n",
LineNumber+1); break;
```

```
        case TSLASH             : printf("%d    TSLASH          /₩n",
LineNumber+1); break;
        case TMOD               : printf("%d    TMOD            %₩n",
LineNumber+1); break;
        case TASSIGN        : printf("%d    TASSIGN         =₩n",
LineNumber+1); break;
        case TADDASSIGN     : printf("%d    TADDASSIGN      +=₩n",
LineNumber+1); break;
        case TSUBASSIGN     : printf("%d    TSUBASSIGN      -=₩n",
LineNumber+1); break;
        case TMULASSIGN     : printf("%d    TMULASSIGN      *=₩n",
LineNumber+1); break;
        case TDIVASSIGN     : printf("%d    TDIVASSIGN      /=₩n",
LineNumber+1); break;
        case TMODASSIGN     : printf("%d    TMODASSIGN      %=₩n",
LineNumber+1); break;
        case TNOT               : printf("%d    TNOT            !₩n",
LineNumber+1); break;
        case TAND               : printf("%d    TAND            &&₩n",
LineNumber+1); break;
        case TOR            : printf("%d    TOR             ||₩n",
LineNumber+1); break;
        case TEQUAL             : printf("%d    TEQUAL          ==₩n",
LineNumber+1); break;
        case TNOTEQU        : printf("%d    TNOTEQU         !=₩n",
LineNumber+1); break;
        case TGREAT             : printf("%d    TGREAT          <₩n",
LineNumber+1); break;
        case TLESS              : printf("%d    TLESS           >₩n",
LineNumber+1); break;
        case TGREATE        : printf("%d    TGREATE         <=₩n",
LineNumber+1); break;
        case TLESSE             : printf("%d    TLESSE          >=₩n",
LineNumber+1); break;
        case TINC               : printf("%d    TINC            ++₩n",
LineNumber+1); break;
        case TDEC               : printf("%d    TDEC            --₩n",
LineNumber+1); break;
        case TOPEN              : printf("%d    TOPEN           (₩n",
LineNumber+1); break;
        case TCLOSE             : printf("%d    TCLOSE          )₩n",
LineNumber+1); break;
        case TCOMMA             : printf("%d    TCOMMA          ,₩n",
LineNumber+1); break;
        case TBIGOPEN       : printf("%d    TBIGOPEN        {₩n",
LineNumber+1); break;
        case TBIGCLOSE      : printf("%d    TBIGCLOSE       }₩n",
LineNumber+1); break;
        case TOPENBRACKET   : printf("%d    TOPENBRACKET    [₩n", LineNumber+1); break;
        case TCLOSEBRACKET  : printf("%d    TCLOSEBRACKET       ]₩n",
```

```
LineNumber+1); break;
        case TSEMICOLON     : printf("%d     TSEMICOLON                ;₩n",
LineNumber+1); break;
        case TNUMBER        : printf("%d     TNUMBER                        %s₩n",
LineNumber+1, yytext); break;
        case TREALNUMBER    : printf("%d     TREALNUMBER               %s₩n",
LineNumber+1, yytext); break;
        case TSTRING        : printf("%d     TSTRING                   %s₩n",
LineNumber+1, yytext); break;
        //TIDENT와 TERROR는 SymbolTable.c와 ReportError.c에서 따로 작업해주기 때문에 아무것도 출력해주지 않았다.
        case TIDENT                 : break;
        case TERROR                 : break;
 }
}

void main(){
        enum tokennumber tn; //tn.h에 정의되어있는 tokennumber 값들을 tn변수명으로 재정의
        printf("₩n<<<Start of LEX!!!>>>₩n₩n");

        printf("Line numberToken type       ST-index  Token₩n");
        printf("-------------------------------------------------------₩n");

        /*yylex()함수를 호출하면 lex.yy.c로 작업이 넘어감
        yylex()로 값을 받아온 후 해당되는 tn에 맞게 출력해주는 함수 printtoken() 호출*/
        while((tn = yylex()) != TEOF){
                printtoken(tn);
        }

        //모든 작업이 종료된 후 전역변수로 설정된 cError값을 출력
        if(cErrors == 0){ //에러가 없을 경우
                printf("No errors detected.₩n");
        }
        else{ //cError값이 0이 아닐 경우
                printf("%d errors detected.₩n", cErrors);
        }
}
```

-ReportError.c-

```
#include "ReportError.h"
#include "SymbolTable.h"
#include "glob.h"

extern int yylex();
extern char *yytext;

/*SymbolTable.c에서 에러를 인식하면 호출되는 c프로그램
error번호에 따라 switch문 이용해서 출력되는 문이 다르다*/
void ReportError(int error){
        switch(error){
        case 1: //오버플로우가 발생했을 경우
```

```
                printf("%d **Error** OVERFLOW ₩n", LineNumber+1);
                cErrors++;
                exit(0);
                break;
        case 2: //지정된 문자가 아닌 경우
                printf("%d **Error** %s Illegal Character ₩n",
LineNumber+1, yytext);
                cErrors++;
                break;
        case 3: //첫 문자가 숫자로 시작하는 경우
                printf("%d **Error** %s Illegal IDENT ₩n",
LineNumber+1, yytext);
                cErrors++;
                break;
        case 4: //이미 존재하는 identifier와 겹치는 경우
                printf("%d **Error** %s Already Existed ₩n",
LineNumber+1, yytext);
                cErrors++;
                break;
        }
}
```

-SymbolTable.c-

```
#include "SymbolTable.h"
#include "ReportError.h"
#include "glob.h"

/*yytext로 문자열 형태로 들어오게 된다. 따라서 문자열을 문자마다 처리해주어야 하기 때문에
입력받는 변수는 char input으로 설정하고, 인덱스를 int a로 설정하였다.*/
char input; //입력될 문자에 관한 변수
int a = 0; //yytext에 사용될 인덱스 변수

extern int yylex();
extern char *yytext;

/*지정된 문자, 숫자, delimiter이외의 character를 구분하는 함수이다.*/
void SkipSeperators() {
        while ( input != EOF && !(isLetter(input) || isDigit(input)) ) {
                if(input != ' ' && input != '₩t' && input != ',' && input != ';' && input != '?' && input != '!'
                        && input != '.' && input != ':' && input != '₩n'){
                        ReportError(2); //ReportError 함수 호출
                }
                strcpy(&input, &yytext[a++]); //yytext에서 다음 문자를 input변수에 복사
        }
}

/*한 글자(character)씩 읽어서 처리하는 함수이다.*/
void ReadID() {
        nextid = nextfree;
```

```
	//첫 character가 숫자로 시작하면 에러
	if (isDigit(input)) {
		ReportError(3);
	}

	//첫 character가 숫자로 시작하지 않는 경우
	else {
		error = 0; //error번호를 0으로 만들어서 위의 error=3과 다른 경우라는 것을 명시해줌
		while (input != EOF && (isLetter(input) || isDigit(input))) {
			if (nextfree == STsize) { //nextfree가 STsize와 같다면 오버플로우 발생
				ReportError(1);
				exit(0);
			}
			//그 외의 정상적인 경우
			ST[nextfree++] = input;
			input = yytext[a++];
		}
	}
}

/*해시테이블의 해시코드 계산하는 함수이다.
delimiter로 구분되는 스트링의 정수값을 모두 더한 후 해시테이블 사이즈로 나눈다.*/
void ComputeHS(int nid, int nfree) {
	int code, i;
	code = 0;
	for (i = nid; i < nfree - 1; i++) {
		code += (int)ST[i];
	}
	hashcode = code % HTsize;
}

/*이미 해시테이블에 존재하는 identifier인지 아닌지 구별하는 함수이다.*/
void LookupHS(int nid, int hscode) {
	HTpointer here;
	int i, j, k;

	found = 0;

	if(HT[hscode] != NULL) {
		here = HT[hscode];
		if(here != NULL && found == 0) {
			found = 1; //기본값은 1이라고 설정
			i = here->index;
			j = nid;
			sameid = i;

			if(ST[i] != '₩0' && ST[j] != '₩0' && found == 1) {
				if (ST[i] != ST[j]) { //만약 같지 않다면 처음 들어온 것이라 판단
					found = 0;
				}
```

```
                                        else { //인덱스를 늘려나가며 계속 비교
                                                i++;
                                                j++;
                                        }
                                }
                                here = here->next;
                        }
                }

                //처음 들어온 identifier
                if (found == 0) {
                        printf("%d                      TIDENT          %d                      ", LineNumber+1, nextid);
                        for (i = nextid+1; i < nextfree - 1; i++) {
                                printf("%c", ST[i]);
                        }
                        printf("₩n");
                        ADDHT(hashcode);
                        nextfree -= 1;
                }

                //이미 들어온 identifier와 겹치는 경우(found == 1)
                else {
                        ReportError(4);
                        nextfree = nextid;
                }
}

/*해시테이블에 추가하는 함수이다.*/
void ADDHT(int hscode) {
        HTpointer ptr;

        ptr = (HTpointer)malloc(sizeof(ptr));
        ptr->index = nextid;
        ptr->next = HT[hscode];
        HT[hscode] = ptr;
}

/*SymbolTable.c프로그램에서 가장 먼저 수행될 함수이다.
lex.yy.c에서 identifier가 인식되면 호출된다.*/
void SymbolTable() {

        //소문자 변환(대소문자 구분 없애기 위해)
        yytext = strlwr(yytext);

        //yytext 길이가 10보다 크면 길이가 10인 character까지 잘라서 yytext에 다시 저장,
        if(strlen(yytext) > 10){
                strncpy(yytext, yytext ,10);
                yytext[10] = 0;
        }
```

```
        //yytext는 문자열이므로 첫 문자부터 마지막 문자까지 for문을 수행
        for(a = 0; a<strlen(yytext); a++) {
                input = yytext[a]; //input에 해당 character 대입

                SkipSeperators();
                ReadID();

                if(input != EOF && error != 3){
                        if (nextfree == STsize) {//오버플로우 발생 조건
                                ReportError(1);
                                exit(0);
                        }
                        ST[nextfree++] = '₩0';

                        ComputeHS(nextid, nextfree);
                        LookupHS(nextid, hashcode);
                }
        }
}
```

4. Inputfile I/O 캡처

-noerrordata1.dat-

```
My math test //test1
5+3=8
great!
6-4=3
wrong!
```

C:₩WINDOWS₩system32₩cmd.exe

```
<<<Start of LEX!!!>>>

Line number     Token type      ST-index        Token
-----------------------------------------------------
1               TIDENT          0               my
1               TIDENT          3               math
1               TIDENT          8               test
2               TNUMBER                         5
2               TPLUS                           +
2               TNUMBER                         3
2               TASSIGN                         =
2               TNUMBER                         8
3               TIDENT          13              great
3               TNOT                            !
4               TNUMBER                         6
4               TMINUS                          -
4               TNUMBER                         4
4               TASSIGN                         =
4               TNUMBER                         3
5               TIDENT          19              wrong
5               TNOT                            !

<<<End of LEX!!!>>>


No errors detected.
계속하려면 아무 키나 누르십시오 . . .
```

-noerrordata2.dat-

```
TODAY = FRIDAY!!!!!
I LOVE YOU



3 > 2
3 IS GREATER THAN 2
my mother LIKEs [YOUR STYLE]
```

C:₩WINDOWS₩system32₩cmd.exe

```
<<<Start of LEX!!!>>>

Line number     Token type      ST-index        Token
-----------------------------------------------------
1               TIDENT          0               today
1               TASSIGN                         =
1               TIDENT          6               friday
1               TNOT                            !
1               TNOT                            !
1               TNOT                            !
1               TNOT                            !
1               TNOT                            !
2               TIDENT          13              i
2               TIDENT          15              love
2               TIDENT          20              you
5               TNUMBER                         3
5               TLESS                           >
5               TNUMBER                         2
6               TNUMBER                         3
6               TIDENT          24              is
6               TIDENT          27              greater
6               TIDENT          35              than
6               TNUMBER                         2
7               TIDENT          40              my
7               TIDENT          43              mother
7               TIDENT          50              likes
7               TOPENBRACKET                    [
7               TIDENT          56              your
7               TIDENT          61              style
7               TCLOSEBRACKET                   ]


<<<End of LEX!!!>>>


No errors detected.
계속하려면 아무 키나 누르십시오 . . .
```

-noerrordata3.dat-

```
IF i cannot go there
"please send me" //an email


while(1){
  x++;
}
```

C:₩WINDOWS₩system32₩cmd.exe

```
<<<Start of LEX!!!>>>

Line number     Token type      ST-index        Token
-----------------------------------------------------
1               TIDENT          0               if
1               TIDENT          3               i
1               TIDENT          5               cannot
1               TIDENT          12              go
1               TIDENT          15              there
2               TSTRING                         "please send me"
4               TWHILE                          while
4               TOPEN                           (
4               TNUMBER                         1
4               TCLOSE                          )
4               TBIGOPEN                        {
5               TIDENT          21              x
5               TINC                            ++
5               TSEMICOLON                      ;
6               TBIGCLOSE                       }


<<<End of LEX!!!>>>


No errors detected.
계속하려면 아무 키나 누르십시오 . . .
```

-errordata1.dat-

```
HEY!


Z=10;
while(Z < 0){
  Z--;
}
printf("Z = %d₩n", Z);

i'm hungry~
```

C:₩WINDOWS₩system32₩cmd.exe

```
<<<Start of LEX!!!>>>

Line number     Token type      ST-index        Token
-----------------------------------------------------
1               TIDENT          0               hey
1               TNOT                            !
3               TIDENT          4               z
3               TASSIGN                         =
3               TNUMBER                         10
3               TSEMICOLON                      ;
4               TWHILE                          while
4               TOPEN                           (
4               **Error**                       z               Already Existed
4               TGREAT                          <
4               **Error**                       0               Illegal Character
4               TCLOSE                          )
4               TBIGOPEN                        {
5               **Error**                       z               Already Existed
5               TDEC                            --
5               TSEMICOLON                      ;
6               TBIGCLOSE                       }
7               TIDENT          6               printf
7               TOPEN                           (
7               TSTRING                         "Z = %d₩n"
7               TCOMMA                          ,
7               **Error**                       z               Already Existed
7               TCLOSE                          )
7               TSEMICOLON                      ;
9               TIDENT          13              i
9               **Error**                       '               Illegal Character
9               TIDENT          15              m
9               TIDENT          17              hungry
9               **Error**                       ~               Illegal Character


<<<End of LEX!!!>>>


6 errors detected.
계속하려면 아무 키나 누르십시오 . . .
```

-errordata2.dat-

```
happy birthday to you~~

goodMORNING

GIVE ME a 3color pen!
return it to me><
```

C:₩WINDOWS₩system32₩cmd.exe

```
<<<Start of LEX!!!>>>

Line number     Token type      ST-index        Token
-------------------------------------------------------
1               TIDENT          0               happy
1               TIDENT          6               birthday
1               TIDENT          15              to
1               TIDENT          18              you
1               **Error**                       ~               Illegal Character
1               **Error**                       ~               Illegal Character
3               TIDENT          22              goodmornin
5               TIDENT          33              give
5               TIDENT          38              me
5               TIDENT          41              a
5               TNUMBER                         3
5               TIDENT          43              color
5               TIDENT          49              pen
5               TNOT                            !
6               TRETURN                         return
6               TIDENT          53              it
6               **Error**                       to              Already Existed
6               **Error**                       me              Already Existed
6               TLESS                           >
6               TGREAT                          <


<<<End of LEX!!!>>>


4 errors detected.
계속하려면 아무 키나 누르십시오 . . .
```

-errordata3.dat-

```
[3+{2/(5-10)}*8] ?   i don't know~!

x = 10; y = 24;
x > y!
x += 3;
y /= 8;

what is x and y?
x = 13;
y = 3;
x < y@

"muslimat"&&"soomin"
```

C:₩WINDOWS₩system32₩cmd.exe

```
<<<Start of LEX!!!>>>

Line number     Token type      ST-index        Token
-----------------------------------------------------
1               TOPENBRACKET                    [
1               TNUMBER                         3
1               TPLUS                           +
1               TBIGOPEN                        {
1               TNUMBER                         2
1               TSLASH                          /
1               TOPEN                           (
1               TNUMBER                         5
1               TMINUS                          -
1               TNUMBER                         10
1               TCLOSE                          )
1               TBIGCLOSE                       }
1               TSTAR                           *
1               TNUMBER                         8
1               TCLOSEBRACKET                   ]
1               **Error**                       ?               Illegal Character
1               TIDENT          0               i
1               TIDENT          2               don
1               **Error**                       '               Illegal Character
1               TIDENT          6               t
1               TIDENT          8               know
1               **Error**                       ~               Illegal Character
1               TNOT                            !
3               TIDENT          13              x
3               TASSIGN                         =
3               TNUMBER                         10
3               TSEMICOLON                      ;
3               TIDENT          15              y
3               TASSIGN                         =
3               TNUMBER                         24
3               TSEMICOLON                      ;
4               **Error**                       x               Already Existed
4               TLESS                           >
4               **Error**                       y               Already Existed
4               TNOT                            !
5               **Error**                       x               Already Existed
5               TADDASSIGN                      +=
5               TNUMBER                         3
5               TSEMICOLON                      ;
6               **Error**                       y               Already Existed
6               TDIVASSIGN                      /=
6               TNUMBER                         8
6               TSEMICOLON                      ;
8               TIDENT          17              what
8               TIDENT          22              is
8               **Error**                       x               Already Existed
8               TIDENT          25              and
8               **Error**                       y               Already Existed
8               **Error**                       ?               Illegal Character
9               **Error**                       x               Already Existed
9               TASSIGN                         =
9               TNUMBER                         13
9               TSEMICOLON                      ;
10              **Error**                       y               Already Existed
10              TASSIGN                         =
10              TNUMBER                         3
10              TSEMICOLON                      ;
11              **Error**                       x               Already Existed
11              TGREAT                          <
11              **Error**                       y               Already Existed
11              **Error**                       @               Illegal Character
13              TSTRING                         "muslimat"
13              TAND                            &&
13              TSTRING                         "soomin"


<<<End of LEX!!!>>>


15 errors detected.
계속하려면 아무 키나 누르십시오 . . .
```

-data given by a teacher_testdata1.dat-

C:₩WINDOWS₩system32₩cmd.exe

```
<<<Start of LEX!!!>>>

Line number     Token type      ST-index        Token
-----------------------------------------------------
1               TIDENT          0               const
1               TIDENT          6               else
1               TCONST                          const
1               TELSE                           else
1               TIF                             if
1               TVOID                           void
1               TINT                            int
1               TIDENT          11              real
2               TIF                             if
2               TRETURN                         return
2               TNUMBER                         123
2               TIDENT          16              abc
2               TIDENT          20              acd
2               **Error**                       acd             Already Existed
2               TREALNUMBER                     5.43
2               **Error**                       abc             Already Existed
3               TPLUS                           +
3               TMINUS                          -
3               TSTAR                           *
3               TSLASH                          /
3               TASSIGN                         =
3               TNOT                            !
3               TNOTEQU                         !=
3               TOPEN                           (
3               TBIGOPEN                        {
3               TCLOSEBRACKET                   ]
3               TSEMICOLON                      ;


<<<End of LEX!!!>>>


2 errors detected.
계속하려면 아무 키나 누르십시오 . . .
```

-data given by a teacher_testdata2.dat-

C:₩WINDOWS₩system32₩cmd.exe

```
<<<Start of LEX!!!>>>

Line number     Token type      ST-index        Token
-----------------------------------------------------
1               TGREAT                          <
1               TNUMBER                         2
1               TIDENT          0               nd
1               TIDENT          3               homework
1               TLESS                           >
2               TMINUS                          -
2               TIDENT          12              scanner
2               TIDENT          20              for
2               TIDENT          24              minic
3               TMINUS                          -
3               TIDENT          30              due
3               **Error**                       :               Illegal Character
3               TNUMBER                         4
3               TSLASH                          /
3               TNUMBER                         13
3               TOPEN                           (
3               TNUMBER                         10
3               TIDENT          34              am
3               TCLOSE                          )
5               TGREAT                          <
5               TIDENT          37              test
5               TIDENT          42              data
5               TLESS                           >
6               TNOT                            !
6               **Error**                       @               Illegal Character
6               **Error**                       &               Illegal Character
6               **Error**                       $               Illegal Character
6               **Error**                       ^               Illegal Character
6               TMOD
6               TPLUS                           +
6               TOPEN                           (
6               **Error**                       |               Illegal Character
6               **Error**                       ~               Illegal Character
7               **Error**                       0               Illegal Character
7               TIDENT          47              xff
7               **Error**                       ?               Illegal Character
```

```
<<<End of LEX!!!>>>


9 errors detected.
계속하려면 아무 키나 누르십시오 . . .
```

-data given by a teacher_testdata3.dat-

C:₩WINDOWS₩system32₩cmd.exe

```
<<<Start of LEX!!!>>>

Line number     Token type      ST-index        Token
-------------------------------------------------------
1               TIDENT          0               everydayha
1               **Error**                       everydayha              Already Existed
4               **Error**                       #               Illegal Character
4               TIDENT          11              include
4               TGREAT                          <
4               TIDENT          19              stdio
4               **Error**                       .               Illegal Character
4               TIDENT          25              h
4               TLESS                           >
5               **Error**                       #               Illegal Character
5               **Error**                       include         Already Existed
5               TSTRING                         "tn.h"
7               TINT                            int
7               TIDENT          27              main
7               TOPEN                           (
7               TVOID                           void
7               TCLOSE                          )
7               TBIGOPEN                        {
8               TINT                            int
8               TIDENT          32              i
8               TCOMMA                          ,
8               TIDENT          34              j
8               TASSIGN                         =
8               **Error**                       0               Illegal Character
8               TSEMICOLON                      ;
9               TWHILE                          while
9               TOPEN                           (
9               **Error**                       i               Already Existed
9               TGREAT                          <
9               TNUMBER                         10
9               TCLOSE                          )
10              TIDENT          36              printf
10              TOPEN                           (
10              TSTRING                         "%d₩n"
10              TCOMMA                          ,
10              **Error**                       i               Already Existed
10              TINC                            ++
10              TCLOSE                          )
10              TSEMICOLON                      ;
11              **Error**                       j               Already Existed
11              TASSIGN                         =
11              TIDENT          43              factorial
11              TOPEN                           (
11              **Error**                       i               Already Existed
11              TCLOSE                          )
11              TSEMICOLON                      ;
12              TRETURN                         return
12              **Error**                       0               Illegal Character
12              TSEMICOLON                      ;
12              TBIGCLOSE                       }
14              TINT                            int
14              **Error**                       factorial               Already Existed
14              TOPEN                           (
14              TINT                            int
14              TIDENT          53              n
14              TCLOSE                          )
14              TBIGOPEN                        {
```

```
15              TIF                             if
15              TOPEN                           (
15              **Error**                       n               Already Existed
15              TEQUAL                          ==
15              TNUMBER                         1
15              TCLOSE                          )
15              TRETURN                         return
15              TNUMBER                         1
15              TSEMICOLON                      ;
16              TELSE                           else
16              TRETURN                         return
16              **Error**                       n               Already Existed
16              TSTAR                           *
16              **Error**                       factorial               Already Existed
16              TOPEN                           (
16              **Error**                       n               Already Existed
16              TMINUS                          -
16              TNUMBER                         1
16              TCLOSE                          )
16              TSEMICOLON                      ;
16              TBIGCLOSE                       }


<<<End of LEX!!!>>>


16 errors detected.
계속하려면 아무 키나 누르십시오 . . .
```