LEXICAL ANALYSIS USING LEX

MOHAMMED FARHAN K N
S7 CSE B


```
%{
%}
%%
[main|int]                  {printf("\n%s\t---> Keyword",yytext);}
([a-z]|[A-Z]|[0-9])+                {if((strcmp(yytext,"int")==0)||(strcmp(yytext,"main")==0)||
(strcmp(yytext,"return")==0)||(strcmp(yytext,"if")==0)||(strcmp(yytext,"else")==0)) printf("\n%s\t---> Keyword",yytext);
else printf("\n%s\t---> Identifier",yytext);}
[(|)|;|{|}|,|&]             { printf("\n%s\t---> Special Character",yytext);}
([0-9])*(.)([0-9])+          { printf("\n%s\t---> Number",yytext);}
[+]                         { printf("\n%s\t---> Addition Operator",yytext);}
[-]                         { printf("\n%s\t---> Subtraction Operator",yytext);}
[*]                         { printf("\n%s\t---> Multiplication Operator",yytext);}
[/]                         { printf("\n%s\t---> Division Operator",yytext);}
(<)                         { printf("\n%s\t---> <RELOP,LT>",yytext);}
(>)                         { printf("\n%s\t---> <RELOP,GT>",yytext);}
(<=)                        { printf("\n%s\t---> <RELOP,LE>",yytext);}
(>=)                        { printf("\n%s\t---> <RELOP,GE>",yytext);}
(==)                        { printf("\n%s\t---> <RELOP,EE>",yytext);}
(!=)                        { printf("\n%s\t---> <RELOP,NE>",yytext);}
(=)                         { printf("\n%s\t---> Assignment Operator",yytext);}
\n                          {;}

%%
main()
{
        yyin=fopen("lexpgminput.c","r");
        yylex();
        fclose(yyin);
}
```

INPUT FILE

```
main ()
{
        int a5  , b , c;
        a = 25;
        b = 13;
        c = a - b;
        if(a<=b)
                b = 5;
        else
                c = a + b;
}
```

OUTPUT

```
mnt/42813/s7comp/lexfile$ flex lexpgm.l
42813@user:/mnt/42813/s7comp/lexfile$ gcc lex.yy.c -ll
42813@user:/mnt/42813/s7comp/lexfile$ ./a.out

main    ---> Keyword
(       ---> Special Character
)       ---> Special Character
{       ---> Special Character
int     ---> Keyword
a5      ---> Identifier
,       ---> Special Character
b       ---> Identifier
,       ---> Special Character
c       ---> Identifier
;       ---> Special Character
a       ---> Keyword
=       ---> Assignment Operator
 25     ---> Number
;       ---> Special Character
b       ---> Identifier
=       ---> Assignment Operator
 13     ---> Number
;       ---> Special Character
c       ---> Identifier
=       ---> Assignment Operator
a       ---> Keyword
-       ---> Subtraction Operator
b       ---> Identifier
;       ---> Special Character
if      ---> Keyword
(       ---> Special Character
a       ---> Keyword
<=      ---> <RELOP,LE>
b       ---> Identifier
)       ---> Special Character
b       ---> Identifier
=       ---> Assignment Operator
 5      ---> Number
;       ---> Special Character
printf  ---> Identifier
(       ---> Special Character"%
d       ---> Identifier\
n       ---> Keyword "
,       ---> Special Character
c       ---> Identifier
)       ---> Special Character
;       ---> Special Character
}       ---> Special Character
```