

```

%option noyywrap
IDENTIFIER [a-z][A-Za-z0-9]*
STRING_CONST [\"][ a-zA-Z0-9_!\,\\s]*\[\"
NUMBER_CONST 0|[\+]*[\-]*[1-9]+[0-9]*
%%
"if" {printf(" if \n");}
"then" {printf(" then \n");}
"else" {printf(" else \n");}
"while" {printf(" while \n");}
"execute" {printf(" execute \n");}
"array" {printf(" array \n");}
"int" {printf(" int \n");}
"string" {printf(" string \n");}
"read" {printf(" read \n");}
"write" {printf(" write \n");}
"," {printf(" coma \n");}
";" {printf(" semicolon \n");}
"[" {printf(" leftSquareBracket \n");}
"]" {printf(" rightSquareBracket \n");}
"(" {printf(" leftRoundBracket \n");}
")" {printf(" rightRoundBracket \n");}
"{" {printf(" startCurlyBracket \n");}
"}" {printf(" endCurlyBracket \n");}
"+" {printf(" plus \n");}
"-" {printf(" minus \n");}
"*" {printf(" multiply \n");}
"/" {printf(" division \n");}
"=" {printf(" equal \n");}
"<" {printf(" lessThan \n");}
"<=" {printf(" lessThanOrEqual \n");}
"==" {printf(" equalTo \n");}
">" {printf(" greaterThan \n");}
">=" {printf(" greaterThanOrEqual \n");}
"<>" {printf(" different \n");}
"%" {printf(" modulo \n");}
{IDENTIFIER} {printf(" identifier \n");}
{STRING_CONST} {printf(" string_const \n");}
{NUMBER_CONST} {printf(" number_const \n");}

[ \t\n]+ {}
[\+][\-][0]+[1-9]+[0-9]* {printf("Illegal number constant \n");}
([\"][ a-zA-Z0-9_!\,\\s]* | [a-zA-Z0-9_!\,\\s]*\[\" ) {printf("Illegal string constant \n");}
[0-9!.,_][A-Za-z0-9] {printf("Wrong name for identifier \n");}
. {printf("Illegal symbol\n"); return -1;}

```

```
%%  
int main(int argc, char **argv)  
{  
    ++argv;  
    --argc;  
    if ( argc > 0 )  
        yyin = fopen( argv[0], "r" );  
    else  
        yyin = stdin;  
    int ntoken, vtoken;  
    ntoken = yylex();  
    while(ntoken)  
    {  
        printf("%d\n", ntoken);  
        if(ntoken == -1)  
            return 0;  
        ntoken = yylex();  
    }  
    return 0;  
}
```