# Code:

```
%option noyywrap
%{
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

double _strtoexp(const char *str)
{
double mantissa = 0;
long exponent = 0;
int sign = 1;
int exp_sign = 1;

if (*str == '-' || *str == '+')
sign = (*str++ == '-') ? -1 : 1;

mantissa = strtof(str, (char **)&str);

if (*str == 'e' || *str == 'E')
{
str++;
if (*str == '-' || *str == '+')
{
exp_sign = (*str++ == '-') ? -1 : 1;
exponent = strtol(str, (char **)&str, 10);
}
else
{
exponent = strtol(str, (char **)&str, 10);
}
}
else
{
exponent = 0;
}

return sign * mantissa * pow(10, exponent * exp_sign);
}

long long _strtoo(char *str)
{
long long num = 0;
int sign = 1;

if (*str == '-' || *str == '+')
sign = (*str++ == '-') ? -1 : 1;

while (*str != '\0')
{
num = num * 8 + (*str - '0');
str++;
}
return num * sign;
}
```

```
long long _strtod(char *str)
{
long long num = 0;
int sign = 1;
if (*str == '-' || *str == '+')
sign = (*str++ == '-') ? -1 : 1;
while (*str != '\0')
{
num = num * 10 + (*str - '0');
str++;
}
return num * sign;
}
long long _strtoh(char *str)
{
long long num = 0;
int sign = 1;
if (*str == '-' || *str == '+')
sign = (*str++ == '-') ? -1 : 1;
if (*str == '0' && (*(str + 1) == 'x' || *(str + 1) == 'X'))
str += 2;
while (*str != '\0')
{
if (*str >= '0' && *str <= '9')
num = num * 16 + (*str++ - '0');
else if (*str >= 'a' && *str <= 'f')
num = num * 16 + (*str++ - 'a' + 10);
else if (*str >= 'A' && *str <= 'F')
num = num * 16 + (*str++ - 'A' + 10);
}
return num * sign;
}
%}
SIGN [+-]?
DIGIT [0-9]
ALPHA_ [a-zA-Z_]
ALNUM_ [a-zA-Z_0-9]
HEXDIGIT [0-9a-fA-F]
%%
{SIGN}[1-9]{DIGIT}*|{SIGN}0 {printf("INTEGER: %lld", _strtod(yytext));}
{SIGN}0[0-7]+ {printf("OCTAL: %lld", _strtoo(yytext));}
{SIGN}0[xX]{HEXDIGIT}+ {printf("HEXADECIMAL: %lld", _strtoh(yytext));}
{SIGN}{DIGIT}*\.?{DIGIT}+ {printf("FLOAT: %lf", _strtoexp(yytext));}
{SIGN}{DIGIT}*\.?{DIGIT}+([eE]{SIGN}+{DIGIT}+)+ {printf("EXPONENT: %.10e",
_strtoexp(yytext));}
{ALPHA_}{ALNUM_}* {printf("IDENTIFIER: %s", yytext);}
%%

int main(int argc ,char *argv[]) {
extern FILE *yyin;
yyin = fopen(argv[1], "r");
yylex();
return 0;
}
```