```
VARNAME [A-Za-z]([A-Za-z0-9])*
DIGIT [0-9]*
INT "int"
CHAR "char"
QUOTE ["""]
DOUBLE "double"
BOOL "bool"
TRUE "true"
FALSE "false"
CHARACTER ["'"].["'"]
FLOATVAL [0-9]*(["."][0-9]*)?
AND "and"
OR "or"
NOT "!"|"not"
SHOW "write"
READ "read"
INCLUDE "include"

%{
    #include "abc.tab.h"
    #include<bits/stdc++.h>
    #include<string>
    using namespace std;
    extern int yyparse();
    char *strcp(const char *str) {
            size_t len = strlen(str);
            char *x = (char *)malloc(len+1);
            if(!x) return NULL;
            memcpy(x,str,len+1);
            return x;
    }
    char *mystrcp(string str) {
            size_t len = str.length();
            char *x = (char *)malloc(len+3);
            strcpy(x, str.c_str());
            return x;
    }
%}

%%
"/n"    { return (NEWLINE); }
"#" { return (HASH); }
["""]   { return (QUOTE); }
{INCLUDE}       { return(INCLUDE); }
[A-Za-z]+".h" { yylval.str = strcp(yytext); return(HEADFILENAME); }
">"         { return(GREATER); }
"<"         { return(LESS); }
">="            { return(GREATEREQ); }
"<="            { return(LESSEQ); }
"func_main()"   { return(MAIN); }
"(" { return(OPENBR1); }
")" { return(CLOSEBR1); }
"[" { return(OPENBR3); }
"]" { return(CLOSEBR3); }
"." { return(DOT); }
{READ} { return(READ); }
{DIGIT} { yylval.ival = atoi(yytext); return(DIGIT); }
{FLOATVAL} { yylval.fval = atof(yytext); return(FLOATVAL); }
{INT} { yylval.type = strcp(yytext); return(INT); }
{CHAR} { yylval.type = strcp(yytext); return(CHAR); }
{DOUBLE} { yylval.type = strcp(yytext); return(DOUBLE); }
{BOOL} { yylval.type = strcp(yytext); return(BOOL); }
{CHARACTER} { yylval.cval = yytext[1];  return(CHARACTER); }
"+" { return(PLUS); }
"-" { return(MINUS); }
"*" { return(MUL); }
```

```
"/" { return(DIV); }
"=" { return(EQUAL); }
";" { return(SEMI); }
"," { return(COMM); }
":" { return(COLON); }
"gcd" {return (GCD);}
"lcm" {return (LCM);}
"sqrt" {return(SQRT);}
"square" {return(SQ);}
"qube" {return (QUBE);}
"prime" {return(PRIME);}
"isprime" {return(ISPRM);}
"fibo" {return(FIBO);}
"bigmod" {return (BIGM);}
"power" {return(POW);}
"sin" {return (SIN);}
"cos" {return(COS);}
"tan" {return(TAN);}
"sin_inv" {return (ASIN);}
"cos_inv" {return (ACOS);}
"tan_inv" {return(ATAN);}
"mod" {return (MOD);}
"if" { return(IF); }
"elif" { return(ELIF); }
"else" { return(ELSE); }
"end_if" { return(ENDIF); }
"end_main" { return(END); }
"loop" { return(LOOP); }
"end_loop" { return(ENDLOOP); }
"in" { return(IN); }
"by"   { return(BY);}
"while" { return(WHILE); }
"end_while" {return(ENDWHILE);}
{AND} { return(AND); }
{OR} { return(OR); }
{NOT}  {return(NOT);}
{TRUE} { yylval.bval = true; return(BOOLVAL); }
{FALSE} { yylval.bval = false; return(BOOLVAL); }
{SHOW} {return(SHOW);}
"func" { return(FUNC); }
"send" { return(RETURN); }
"end_func" { return(FUNCEND); }
["/"]["/"].* { return(SCMNT); }
["%"]["c"|"d"|"b"|"f"|"i"]    { yylval.c = yytext[1]; return(PERCENT); }
["|"][^|]*["|"]   {
  string x = yytext;
  string y = x.substr(1,x.length() - 2);
  yylval.str = mystrcp(y);
  return(STRING);
}
["/"]["*"]("/n")*[^*/]*("/n")*["*"]["/"] { return(MCMNT);; }
{VARNAME} {yylval.name = strcp(yytext); return(VARNAME); }
[ \n\t]*

%%

int yywrap(){
return 1;
}
```