```
%{
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

char st_id[1000] = "";
char st_const[1000] = "";
char pif[1000] = "";
char digit_check[100][10];
int arr = 0;
int id_count = 0;
int const_count = 0;
%}

%option noyywrap

ID              [a-z][a-zA-Z0-9_]*
LETTER                  [A-Za-z]
ZERODIGIT       [1-9]
DIGIT           [0-9]
CHAR            ^'[a-zA-Z0-9]'$
STRING                  ^'[a-zA-Z0-9]+'$

%%
"+"|"-"|"/"|"*"|"="|"and"|"or"|"<"|">"|"<="|">="|"=="|"<>"|"%" {strcat(pif, "("); strcat(pif, yytext); strcat(pif, ", 0)\n");}
"("|")"|"["|"]"|":"|";"|"," {strcat(pif, "("); strcat(pif, yytext); strcat(pif, ", 0)\n");}
"function"|"list"|"char"|"int"|"string"|"go"|"from"|"to"|"length"|"check"|"otherwise"|"return"|"write"|"read" {strcat(pif, "("); strcat(pif, yytext); strcat(pif, ", 0)\n");}

{DIGIT}+ {
        bool found = false;
        int i=0;
        int pos = 0;
        for (i=0; i<arr; i++)
        {
                char *r = *(digit_check + i);
                int result = strcmp(r, yytext);
                if (result == 0)
                {
                        pos = i;
                        found = true;
                }
```

```
    }
    if (found == true)
    {
        char count_str[10];
        itoa(pos, count_str, 10);

        strcat(pif, "(const, ");
        strcat(pif, count_str);
        strcat(pif, ")\n");
    } else
    {
        strcpy(digit_check[arr], yytext);
        ++arr;

        ++const_count;
        char count_str[10];
        itoa(const_count, count_str, 10);

        strcat(pif, "(const, ");
        strcat(pif, count_str);
        strcat(pif, ")\n");

        strcat(st_const, yytext);
        strcat(st_const, " ");
        strcat(st_const, count_str);
        strcat(st_const, "\n");
    }
     }
{CHAR} {
    char *found = strstr(st_const, yytext);
    if (found)
    {
        int pos;
        if (found-st_id != 0)
            pos = found-st_const + strlen(yytext) + 1;
        else
            pos = found-st_const + strlen(yytext) + 2;
        char r = st_const[pos];

        strcat(pif, "(const, ");
        strcat(pif, &r);
        strcat(pif, ")\n");
    } else
    {
        ++const_count;
        char count_str[10];
        itoa(const_count, count_str, 10);

        strcat(pif, "(const, ");
        strcat(pif, count_str);
        strcat(pif, ")\n");

        strcat(st_const, yytext);
        strcat(st_const, " ");
        strcat(st_const, count_str);
        strcat(st_const, "\n");
    }
  }
{STRING} {
    char *found = strstr(st_const, yytext);
    if (found)
    {
        int pos;
        if (found-st_id != 0)
            pos = found-st_const + strlen(yytext) + 1;
        else
            pos = found-st_const + strlen(yytext) + 2;
        char r = st_const[pos];

        strcat(pif, "(const, ");
        strcat(pif, &r);
        strcat(pif, ")\n");
    } else
    {
        ++const_count;
        char count_str[10];
        itoa(const_count, count_str, 10);

        strcat(pif, "(const, ");
        strcat(pif, count_str);
        strcat(pif, ")\n");

        strcat(st_const, yytext);
        strcat(st_const, " ");
        strcat(st_const, count_str);
        strcat(st_const, "\n");
    }
     }
```

```
false {
        char *found = strstr(st_const, yytext);
        if (found)
        {
                int pos;
                if (found-st_id != 0)
                        pos = found-st_const + strlen(yytext) + 1;
                else
                        pos = found-st_const + strlen(yytext) + 2;
                char r = st_const[pos];

                strcat(pif, "(const, ");
                strcat(pif, &r);
                strcat(pif, ")\n");
        } else
        {
                ++const_count;
                char count_str[10];
                itoa(const_count, count_str, 10);

                strcat(pif, "(const, ");
                strcat(pif, count_str);
                strcat(pif, ")\n");

                strcat(st_const, "false ");
                strcat(st_const, count_str);
                strcat(st_const, "\n");
        }
    }
true {
        char *found = strstr(st_const, yytext);
        if (found)
        {
                int pos;
                if (found-st_id != 0)
                        pos = found-st_const + strlen(yytext) + 1;
                else
                        pos = found-st_const + strlen(yytext) + 2;
                char r = st_const[pos];

                strcat(pif, "(const, ");
                strcat(pif, &r);
                strcat(pif, ")\n");
        } else
        {
                ++const_count;
                char count_str[10];
                itoa(const_count, count_str, 10);

                strcat(pif, "(const, ");
                strcat(pif, count_str);
                strcat(pif, ")\n");

                strcat(st_const, "false ");
                strcat(st_const, count_str);
                strcat(st_const, "\n");
        }
      }

{ID} {
        char *found = strstr(st_id, yytext);
        if (found && found-st_id != 0)
        {
                int pos = found-st_id + strlen(yytext) + 1;
                char r = st_id[pos];

                strcat(pif, "(id, ");
                strcat(pif, &r);
                strcat(pif, ")\n");
        }
        else
        {
                ++id_count;
                char count_str[10] = "";
                itoa(id_count, count_str, 10);

                strcat(pif, "(id, ");
                strcat(pif, count_str);
                strcat(pif, ")\n");

                strcat(st_id, yytext);
                strcat(st_id, " ");
                strcat(st_id, count_str);
                strcat(st_id, "\n");
        }
      }
```

```
%%
main( argc, argv )
int argc;
char **argv;
{
    ++argv, --argc; /* skip over program name */
    if ( argc > 0 )
    yyin = fopen( argv[0], "r" );
    else
     yyin = stdin;
    yylex();
    printf("PIF:\n%s", pif);
    printf("Symbols table id:\n%s", st_id);
    printf("Symbols table const:\n%s", st_const);
}
```