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

int symbols_cap = 0;
int symbols_len = 0;
char** symbols = NULL;

void grow() {
  int size = 2 * symbols_len;
  if(size < 1) {
    size = 1;
  }
  char** new_symbols = malloc(size * sizeof(char*));
  for(int i = 0; i < symbols_len; i++) {
    new_symbols[i] = symbols[i];
  }
  if(symbols != NULL) {
    free(symbols);
  }
  symbols = new_symbols;
  symbols_cap = size;
}

int get_string(char* s) {
  for(int i = 0; i < symbols_len; i++) {
    if(strcmp(symbols[i], s) == 0) {
      return i;
    }
  }
  if(symbols_cap == symbols_len) {
    grow();
  }
  symbols[symbols_len] = malloc(strlen(s) + 1);
  strcpy(symbols[symbols_len], s);
  return symbols_len++;
}

%}

LETTER                [A-Za-z]
DIGIT                 [0-9]
NONZERODIGIT          [1-9]
INTCONSTANT           [+-]?({NONZERODIGIT}{DIGIT}*|0)
STRINGCONSTANT        \"([^"]|\\\")*\"
```

```
BOOLCONSTANT          true|false
FLOATCONSTANT         {INTCONSTANT}(\.{DIGIT}+)?
WHITESPACE            [\n\t\r ]
COMMENT               \/\/.*$
CONSTANT              {INTCONSTANT}|{FLOATCONSTANT}|{BOOLCONSTANT}|{STRINGCONSTANT}
IDENTIFIER            (_|{LETTER})({LETTER}|{DIGIT}|_)*
%%
{WHITESPACE} {}
{COMMENT} {}
"!" { printf("-1: "); ECHO; printf("\n"); }
"+" { printf("-2: "); ECHO; printf("\n"); }
"-" { printf("-3: "); ECHO; printf("\n"); }
"*" { printf("-4: "); ECHO; printf("\n"); }
"/" { printf("-5: "); ECHO; printf("\n"); }
"%" { printf("-6: "); ECHO; printf("\n"); }
"==" { printf("-7: "); ECHO; printf("\n"); }
"!=" { printf("-8: "); ECHO; printf("\n"); }
"<" { printf("-9: "); ECHO; printf("\n"); }
"<=" { printf("-10: "); ECHO; printf("\n"); }
">" { printf("-11: "); ECHO; printf("\n"); }
">=" { printf("-12: "); ECHO; printf("\n"); }
"=" { printf("-13: "); ECHO; printf("\n"); }
"&&" { printf("-14: "); ECHO; printf("\n"); }
"||" { printf("-15: "); ECHO; printf("\n"); }
"{" { printf("-16: "); ECHO; printf("\n"); }
"}" { printf("-17: "); ECHO; printf("\n"); }
"(" { printf("-18: "); ECHO; printf("\n"); }
")" { printf("-19: "); ECHO; printf("\n"); }
";" { printf("-20: "); ECHO; printf("\n"); }
"," { printf("-21: "); ECHO; printf("\n"); }
"'" { printf("-22: "); ECHO; printf("\n"); }
":" { printf("-23: "); ECHO; printf("\n"); }
"." { printf("-24: "); ECHO; printf("\n"); }
"let" { printf("-25: "); ECHO; printf("\n"); }
"if" { printf("-26: "); ECHO; printf("\n"); }
"else" { printf("-27: "); ECHO; printf("\n"); }
"while" { printf("-28: "); ECHO; printf("\n"); }
"print" { printf("-29: "); ECHO; printf("\n"); }
"readI32" { printf("-30: "); ECHO; printf("\n"); }
"readU32" { printf("-31: "); ECHO; printf("\n"); }
"readStr" { printf("-32: "); ECHO; printf("\n"); }
"readBool" { printf("-33: "); ECHO; printf("\n"); }
"readF32" { printf("-34: "); ECHO; printf("\n"); }
"i32" { printf("-35: "); ECHO; printf("\n"); }
"u32" { printf("-36: "); ECHO; printf("\n"); }
"str" { printf("-37: "); ECHO; printf("\n"); }
```

```
"bool" { printf("-38: "); ECHO; printf("\n"); }
"f32" { printf("-39: "); ECHO; printf("\n"); }
"array" { printf("-40: "); ECHO; printf("\n"); }
"true" { printf("-41: "); ECHO; printf("\n"); }
"false" { printf("-42: "); ECHO; printf("\n"); }
"[" { printf("-43: "); ECHO; printf("\n"); }
"]" { printf("-44: "); ECHO; printf("\n"); }
{IDENTIFIER} { printf("%d: ", get_string(yytext)); ECHO; printf("\n"); }
{CONSTANT} { printf("%d: ", get_string(yytext)); ECHO; printf("\n"); }
. { printf("UNKNOWN "); ECHO; printf("\n");exit(1); }
%%
```

## Demo

### Program

```
let x: i32;
let y: i32;

x = readI32();
y = readI32();

while y != 0 {
  let z: i32;
  z = x % y;
  x = y;
  y = z;
}

// This is a comment
print("Gcd is ", x);
```

### Output

```
-25: let
0: x
-23: :
-35: i32
-20: ;
-25: let
1: y
-23: :
-35: i32
-20: ;
0: x
-13: =
-30: readI32
```

```
-18: (
-19: )
-20: ;
1: y
-13: =
-30: readI32
-18: (
-19: )
-20: ;
-28: while
1: y
-8: !=
2: 0
-16: {
-25: let
3: z
-23: :
-35: i32
-20: ;
3: z
-13: =
0: x
-6: %
1: y
-20: ;
0: x
-13: =
1: y
-20: ;
1: y
-13: =
3: z
-20: ;
-17: }
-29: print
-18: (
4: "Gcd is "
-21: ,
0: x
-19: )
-20: ;
```