```
/*Added header files here*/
%{

#include<bits/stdc++.h>
using namespace std;
map<string,set<string>> symbolTable;
%}

/*Definitions*/
digit       [[:digit:]]
letter      [[:lower:]]
label       [[:upper:]]
id          {letter}({letter}|{digit})*
literal     true|false|{digit}+|\"({letter}|\w)+\"
type        int|bool|string
relop       ">"|">="|"<"|"<="|"!="|"=="
arithop     "*"|"+"|"/"|"-"|"%"
logop       "and"|"or"|"not"
symbol      ";"|"("|")"|":"|"{"|"}"

/*Keywords*/
fn          "fn"
main        "main"
let         "let"
jump        "jump"
print       "print"
input       "input"
return      "return"


/*Translation*/
%%
"END" {
    return 0;
}

{logop} {
    cout << "Logical Op";
}

{relop} {
    cout << yytext;
}

{arithop} {
    cout << "Arithmetic op";
}

{literal} {
    symbolTable["Literal"].insert(yytext);
    cout << "literals";
}

{label} {
    symbolTable["Label"].insert(yytext);
    cout << "Label";
}

{type} {
    cout << "Data type";
}

{fn}|{main}|{let}|{jump}|{print}|{input}|{return} {
    cout << "keyword";
}

{id} {
    symbolTable["Id"].insert(yytext);
    cout << "identifier";
}

%%
/*Declaration*/
```

```
int yywrap(void){
    return 0;
}

int main(int argc, char *argv[])
{

    symbolTable["Type"]     =   {"int","bool","string"};
    symbolTable["Relop"]    =   {">",">=","<","<=","!=","=="};
    symbolTable["Logop"]    =   {"and","or","not"};
    symbolTable["Symbol"]   =   {";","(",")",":","{","}"};
    symbolTable["Arithop"]  =   {"*","+","/","-","%"};
    symbolTable["Keywords"] =   {"fn","main","let","jump","print","input","return"};
    symbolTable["Id"]       =   {};
    symbolTable["Literal"]  =   {};
    symbolTable["Label"]    =   {};

    freopen("input.txt","r",stdin);
    freopen("output.txt","w",stdout);
    yylex();

    for (auto &i: symbolTable) {
        for (auto &j: i.second)
            cout << i.first << " " << j << endl;
    }

    return 0;
}
```