```c
#include <stdio.h>
#include <ctype.h>
#include <string.h>

#define MAX_TOKEN_LENGTH 100

int isOperator(char ch) {
    char operators[] = "+-*/=<>^";
    for (int i = 0; i < strlen(operators); i++) {
        if (ch == operators[i]) {
            return 1;
        }
    }
    return 0;
}

int isPunctuation(char ch) {
    char punctuations[] = {"\",;:?!|&`~'"};
    for (int i = 0; i < strlen(punctuations); i++) {
        if (ch == punctuations[i]) {
            return 1;
        }
    }
    return 0;
}

int isSpecialSymbol(char ch) {
    char specialsymbols[] = {".#"};
    for (int i = 0; i < strlen(specialsymbols); i++) {
        if (ch == specialsymbols[i]) {
            return 1;
        }
    }
    return 0;
}

int isOpenbrackets(char ch) {
    char openbrackets[] = {"({["};
    for (int i = 0; i < strlen(openbrackets); i++) {
        if (ch == openbrackets[i]) {
            return 1;
        }
    }
    return 0;
}

int isClosebrackets(char ch) {
    char closebrackets[] = {")}]"};
    for (int i = 0; i < strlen(closebrackets); i++) {
        if (ch == closebrackets[i]) {
            return 1;
        }
    }
    return 0;
}


int isKeyword(char* str) {
    char keywords[32][10] = {"auto", "break", "case", "char", "const", "continue",
"default","do", "double", "else", "enum", "extern", "float", "for",
    "goto", "if", "int", "long", "register", "return", "short","signed", "sizeof",
"static", "struct",
    "switch", "typedef","union", "unsigned", "void", "volatile", "while"};
    int numKeywords = sizeof(keywords) / sizeof(keywords[0]);
    for (int i = 0; i < numKeywords; i++) {
        if (strcmp(str, keywords[i]) == 0) {
            return 1;
        }
    }
    return 0;
}

int isIdentifier(char* str) {
    if (!isalpha(str[0])) {
        return 0;
    }
    for (int i = 1; i < strlen(str); i++) {
        if (!isalnum(str[i])) {
            return 0;
        }
    }
    return 1;
}

int isNumber(char* str) {
    for (int i = 0; i < strlen(str); i++) {
        if (!isdigit(str[i])) {
            return 0;
        }
    }
    return 1;
}

void printTokenTable(char* filename) {
    FILE* file = fopen("exp1.txt", "r");
    if (file == NULL) {
        printf("Error opening file.\n");
        return;
    }

    char line[MAX_TOKEN_LENGTH];
    int lineNo = 1;

    printf("Token\t\tType\t\tLine No.\n");
    printf("-----------------------------------------\n");

    while (fgets(line, sizeof(line), file)) {
        char* token = strtok(line, " \t\n");
        while (token != NULL) {
            if (isKeyword(token)) {
                printf("%s\t\tKeyword\t\t\t%d\n", token, lineNo);
            } else if (isPunctuation(token[0])) {
                printf("%s\t\tPunctuation\t\t%d\n", token, lineNo);
            } else if (isSpecialSymbol(token[0])) {
                printf("%s\t\tSpecial Symbol\t\t%d\n", token, lineNo);
            }   else if (isOperator(token[0])) {
                printf("%s\t\tOperator\t\t\%d\n", token, lineNo);
            } else if (isIdentifier(token)) {
                printf("%s\t\tIdentifier\t\t%d\n", token, lineNo);
            } else if (isNumber(token)) {
                printf("%s\t\tNumber\t\t\t%d\n", token, lineNo);
            } else if (isOpenbrackets(token[0])) {
                printf("%s\t\tOpening Brackets\t%d\n", token, lineNo);
            } else if (isClosebrackets(token[0])) {
                printf("%s\t\tClosing Brackets\t%d\n", token, lineNo);
            } else {
                printf("%s\t\tInvalid\t\t\t%d\n", token, lineNo);
            }
            token = strtok(NULL, " \t\n");
        }
        lineNo++;
    }

    fclose(file);
}

int main() {
    char filename[] = "exp1.txt";
    printTokenTable(filename);
    return 0;
}
```