

```
1  #include <stdio.h>
2  #include <ctype.h>
3  #include <string.h>
4
5  #define MAX_TOKEN_LENGTH 100
6
7  int isOperator(char ch) {
8      char operators[] = "+-*/=<>^";
9      for (int i = 0; i < strlen(operators); i++) {
10         if (ch == operators[i]) {
11             return 1;
12         }
13     }
14     return 0;
15 }
16
17 int isPunctuation(char ch) {
18     char punctuations[] = {"\\", ";", ":", "!", "&", "~", "'"};
19     for (int i = 0; i < strlen(punctuations); i++) {
20         if (ch == punctuations[i]) {
21             return 1;
22         }
23     }
24     return 0;
25 }
26
27 int isSpecialSymbol(char ch) {
28     char specialsymbols[] = {".", "#"};
29     for (int i = 0; i < strlen(specialsymbols); i++) {
30         if (ch == specialsymbols[i]) {
31             return 1;
32         }
33     }
34     return 0;
35 }
36
37 int isOpenbrackets(char ch) {
38     char openbrackets[] = {"{", "["};
39     for (int i = 0; i < strlen(openbrackets); i++) {
40         if (ch == openbrackets[i]) {
41             return 1;
42         }
43     }
44     return 0;
45 }
46
47 int isClosebrackets(char ch) {
48     char closebrackets[] = {"}", "]}"]};
49     for (int i = 0; i < strlen(closebrackets); i++) {
50         if (ch == closebrackets[i]) {
51             return 1;
52         }
53     }
54     return 0;
55 }
56
57
58 int isKeyword(char* str) {
59     char keywords[32][10] = {"auto", "break", "case", "char", "const", "continue",
60 "default", "do", "double", "else", "enum", "extern", "float", "for",
61 "goto", "if", "int", "long", "register", "return", "short", "signed", "sizeof",
62 "static", "struct",
63 "switch", "typedef", "union", "unsigned", "void", "volatile", "while"};
64     int numKeywords = sizeof(keywords) / sizeof(keywords[0]);
65     for (int i = 0; i < numKeywords; i++) {
66         if (strcmp(str, keywords[i]) == 0) {
67             return 1;
68         }
69     }
70     return 0;
71 }
72
73 int isIdentifier(char* str) {
74     if (!isalpha(str[0])) {
75         return 0;
76     }
77 }
```

```
75     for (int i = 1; i < strlen(str); i++) {
76         if (!isalnum(str[i])) {
77             return 0;
78         }
79     }
80     return 1;
81 }
82
83 int isNumber(char* str) {
84     for (int i = 0; i < strlen(str); i++) {
85         if (!isdigit(str[i])) {
86             return 0;
87         }
88     }
89     return 1;
90 }
91
92 void printTokenTable(char* filename) {
93     FILE* file = fopen("exp1.txt", "r");
94     if (file == NULL) {
95         printf("Error opening file.\n");
96         return;
97     }
98
99     char line[MAX_TOKEN_LENGTH];
100    int lineNo = 1;
101
102    printf("Token\t\tType\t\tLine No.\n");
103    printf("-----\n");
104
105    while (fgets(line, sizeof(line), file)) {
106        char* token = strtok(line, " \t\n");
107        while (token != NULL) {
108            if (isKeyword(token)) {
109                printf("%s\t\tKeyword\t\t\t%d\n", token, lineNo);
110            } else if (isPunctuation(token[0])) {
111                printf("%s\t\tPunctuation\t\t\t%d\n", token, lineNo);
112            } else if (isSpecialSymbol(token[0])) {
113                printf("%s\t\tSpecial Symbol\t\t\t%d\n", token, lineNo);
114            } else if (isOperator(token[0])) {
115                printf("%s\t\tOperator\t\t\t%d\n", token, lineNo);
116            } else if (isIdentifier(token)) {
117                printf("%s\t\tIdentifier\t\t\t%d\n", token, lineNo);
118            } else if (isNumber(token)) {
119                printf("%s\t\tNumber\t\t\t\t%d\n", token, lineNo);
120            } else if (isOpenbrackets(token[0])) {
121                printf("%s\t\tOpening Brackets\t\t\t%d\n", token, lineNo);
122            } else if (isClosebrackets(token[0])) {
123                printf("%s\t\tClosing Brackets\t\t\t%d\n", token, lineNo);
124            } else {
125                printf("%s\t\tInvalid\t\t\t\t%d\n", token, lineNo);
126            }
127            token = strtok(NULL, " \t\n");
128        }
129        lineNo++;
130    }
131
132    fclose(file);
133 }
134
135 int main() {
136     char filename[] = "exp1.txt";
137     printTokenTable(filename);
138     return 0;
139 }
```