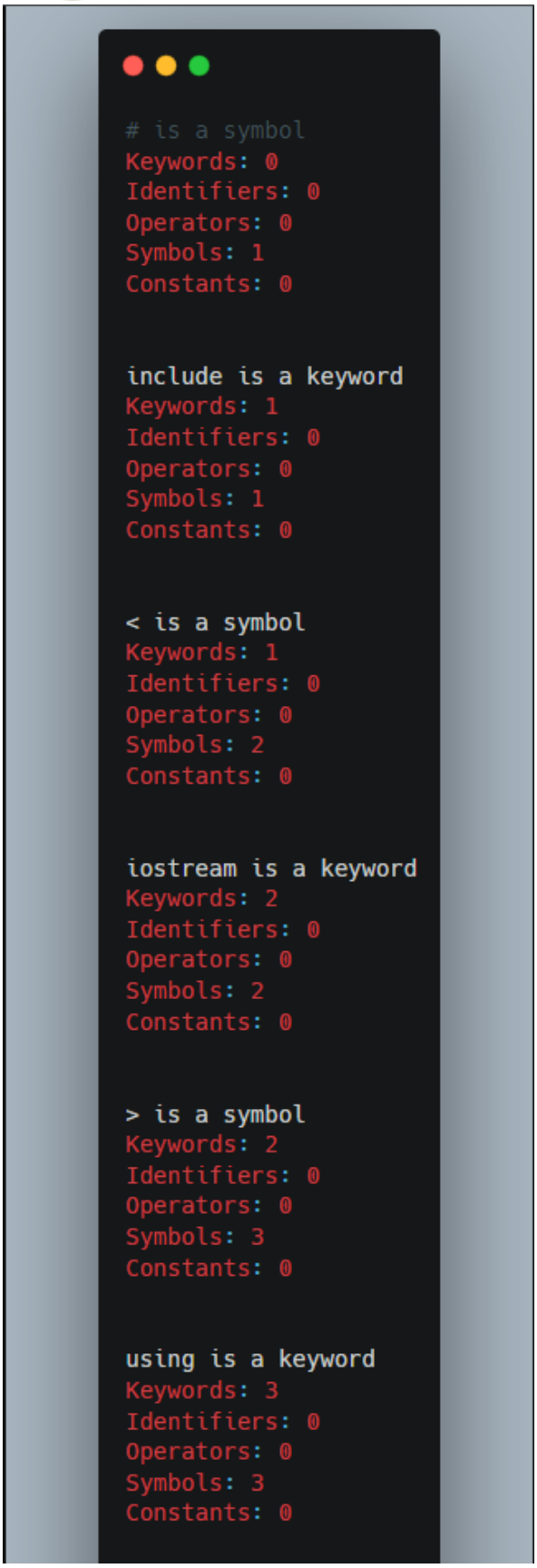

```
# is a symbol
Keywords: 0
Identifiers: 0
Operators: 0
Symbols: 1
Constants: 0


include is a keyword
Keywords: 1
Identifiers: 0
Operators: 0
Symbols: 1
Constants: 0


< is a symbol
Keywords: 1
Identifiers: 0
Operators: 0
Symbols: 2
Constants: 0


iostream is a keyword
Keywords: 2
Identifiers: 0
Operators: 0
Symbols: 2
Constants: 0


> is a symbol
Keywords: 2
Identifiers: 0
Operators: 0
Symbols: 3
Constants: 0


using is a keyword
Keywords: 3
Identifiers: 0
Operators: 0
Symbols: 3
Constants: 0
```

```
namespace is a keyword
Keywords: 4
Identifiers: 0
Operators: 0
Symbols: 3
Constants: 0


std is a keyword
Keywords: 5
Identifiers: 0
Operators: 0
Symbols: 3
Constants: 0


; is a symbol
Keywords: 5
Identifiers: 0
Operators: 0
Symbols: 4
Constants: 0


int is a keyword
Keywords: 6
Identifiers: 0
Operators: 0
Symbols: 4
Constants: 0


main is a keyword
Keywords: 7
Identifiers: 0
Operators: 0
Symbols: 4
Constants: 0


( is a symbol
Keywords: 7
Identifiers: 0
Operators: 0
Symbols: 5
Constants: 0


) is a symbol
Keywords: 7
```

```
) is a symbol
Keywords: 7
Identifiers: 0
Operators: 0
Symbols: 6
Constants: 0

{ is a symbol
Keywords: 7
Identifiers: 0
Operators: 0
Symbols: 7
Constants: 0

int is a keyword
Keywords: 8
Identifiers: 0
Operators: 0
Symbols: 7
Constants: 0

a is an identifier
Keywords: 8
Identifiers: 1
Operators: 0
Symbols: 7
Constants: 0

= is an operator
Keywords: 8
Identifiers: 1
Operators: 1
Symbols: 7
Constants: 0

3 is a constant
Keywords: 8
Identifiers: 1
Operators: 1
Symbols: 7
Constants: 1
```

```
; is a symbol
Keywords: 8
Identifiers: 1
Operators: 1
Symbols: 8
Constants: 1

int is a keyword
Keywords: 9
Identifiers: 1
Operators: 1
Symbols: 8
Constants: 1

b is an identifier
Keywords: 9
Identifiers: 2
Operators: 1
Symbols: 8
Constants: 1

= is an operator
Keywords: 9
Identifiers: 2
Operators: 2
Symbols: 8
Constants: 1

6 is a constant
Keywords: 9
Identifiers: 2
Operators: 2
Symbols: 8
Constants: 2

; is a symbol
Keywords: 9
Identifiers: 2
Operators: 2
Symbols: 9
Constants: 2
```

```
int is a keyword
Keywords: 10
Identifiers: 2
Operators: 2
Symbols: 9
Constants: 2


c is an identifier
Keywords: 10
Identifiers: 3
Operators: 2
Symbols: 9
Constants: 2


= is an operator
Keywords: 10
Identifiers: 3
Operators: 3
Symbols: 9
Constants: 2


9 is a constant
Keywords: 10
Identifiers: 3
Operators: 3
Symbols: 9
Constants: 3


; is a symbol
Keywords: 10
Identifiers: 3
Operators: 3
Symbols: 10
Constants: 3


int is a keyword
Keywords: 11
Identifiers: 3
Operators: 3
Symbols: 10
Constants: 3


d is an identifier
Keywords: 11
Identifiers: 4
Operators: 3
Symbols: 10
Constants: 3


= is an operator
Keywords: 11
Identifiers: 4
Operators: 4
Symbols: 10
Constants: 3


a is an identifier
Keywords: 11
Identifiers: 5
Operators: 4
Symbols: 10
Constants: 3


+ is an operator
Keywords: 11
Identifiers: 5
Operators: 5
Symbols: 10
Constants: 3


b is an identifier
Keywords: 11
Identifiers: 6
Operators: 5
Symbols: 10
Constants: 3


+ is an operator
Keywords: 11
Identifiers: 6
Operators: 6
Symbols: 10
Constants: 3
```

```
c is an identifier
Keywords: 11
Identifiers: 7
Operators: 6
Symbols: 10
Constants: 3


; is a symbol
Keywords: 11
Identifiers: 7
Operators: 6
Symbols: 11
Constants: 3


cout is an identifier
Keywords: 11
Identifiers: 8
Operators: 6
Symbols: 11
Constants: 3


<< is an operator
Keywords: 11
Identifiers: 8
Operators: 7
Symbols: 11
Constants: 3


d is an identifier
Keywords: 11
Identifiers: 9
Operators: 7
Symbols: 11
Constants: 3


; is a symbol
Keywords: 11
Identifiers: 9
Operators: 7
Symbols: 12
Constants: 3


return is a keyword
Keywords: 12
Identifiers: 9
Operators: 7
Symbols: 12
Constants: 3


0 is a constant
Keywords: 12
Identifiers: 9
Operators: 7
Symbols: 12
Constants: 4


; is a symbol
Keywords: 12
Identifiers: 9
Operators: 7
Symbols: 13
Constants: 4


} is a symbol
Keywords: 12
Identifiers: 9
Operators: 7
Symbols: 14
Constants: 4


SUMMARY:
Keywords: 12
list of keywords:
include
iostream
using
namespace
std
int
main
int
int
int
int
```

```
Identifiers: 9
list of identifiers:
a
b
c
d
a
b
c
cout


Operators: 7
list of operators:
=
=
=
=
+
+


Symbols: 14
list of symbols:
#
<
>
;
(
)
{
;
;
;
;
;
;


Constants: 4
list of constants:
3
6
9
```

```
#include <iostream>

#include <fstream>

#include <string>

#include <regex>

#include <sstream>

#include <vector>

#include <cctype>

#include <unordered_set>


using namespace std;


// Function to check if a token is a keyword

bool isKeyword(const string &token)

{

    unordered_set<string> keywords = {"auto", "bool", "break", "case", "catch", "char","class","const",

                                      "continue", "default", "delete", "do", "double", "else", "enum",
```

```
                    "extern", "float", "for", "goto", "if", "int", "include", "inline",
                    "iostream", "long", "main", "namespace", "register","return", "short",
                    "signed", "sizeof", "static", "std", "struct", "switch", "typedef",
                    "template", "typename", "try", "this" "union", "using", "unsigned",
                    "void", "volatile", "virtual", "while"};
    return keywords.count(token) > 0;
}

bool isOperator(const string &token){
    unordered_set<string> operators = {"+", "-", "*", "/", "%", "==", "!=", "<=", ">=", "&&", "||", "!", "=",
                     "+=", "-=", "/=", "*=", "++", "--", "&", "|", "^", "~", "<<", ">>", "::"};

    return operators.count(token) > 0;
}

bool isConstant(const string &token){
    regex characterConstant("'.'");
    regex stringConstant("\".*\"");
```

```
    regex integerConstant("^\\d+$");
    regex floatingPointConstant("^[+-]?\\d*\\.\\d+$");

    return (regex_match(token, characterConstant) || regex_match(token, stringConstant) ||
        regex_match(token, integerConstant) || regex_match(token, floatingPointConstant));
}

int main()
{
    string filename = "sourceFile.txt";
    ifstream inputFile(filename);

    if (!inputFile.is_open())
    {
        cerr << "Error: Unable to open the input file." << endl;
        return 1;
    }
```

```
string line;

int keywordsCount = 0;

int identifiersCount = 0;

int operatorsCount = 0;

int symbolsCount = 0;

int constantsCount = 0;

string keyarr[100];

string idarr[100];

string oparr[100];

string symarr[100];

string conarr[100];



while (getline(inputFile, line))

{

   istringstream inputStream(line);

   string token;



   while (inputStream >> token)
```

```
{


    if (isKeyword(token))

    {

        keywordsCount++;

        cout << token << " is a keyword" << endl;

        // push token to array

        keyarr[keywordsCount] = token;

    }

    else if (isalpha(token[0]) || token[0] == '_')

    {

        identifiersCount++;

        cout << token << " is an identifier" << endl;

        // push token to array

        idarr[identifiersCount] = token;

    }

    else if (isOperator(token))

    {
```

```
    operatorsCount++;

    cout << token << " is an operator" << endl;

    // push token to array

    oparr[operatorsCount] = token;

}

else if (isConstant(token))

{

    constantsCount++;

    cout << token << " is a constant" << endl;

    // push token to array

    conarr[constantsCount] = token;

}

else if (ispunct(token[0]))

{

    symbolsCount++;

    cout << token << " is a symbol" << endl;

    // push token to array

    symarr[symbolsCount] = token;
```

```
        }


        cout << "Keywords: " << keywordsCount << endl;

        cout << "Identifiers: " << identifiersCount << endl;

        cout << "Operators: " << operatorsCount << endl;

        cout << "Symbols: " << symbolsCount << endl;

        cout << "Constants: " << constantsCount << endl;

        cout << " \n " << endl;

    }

}



inputFile.close();

cout << "SUMMARY:" << endl;

cout << "Keywords: " << keywordsCount << endl;

cout << "list of keywords: ";

// for loop for printing keyarr

for (int i = 0; i < keywordsCount; i++)

{
```

```
    cout << keyarr[i] << endl;

}

cout << " \n " << endl;



cout << "Identifiers: " << identifiersCount << endl;

cout << "list of identifiers: ";

// for loop for printing idarr

for (int i = 0; i < identifiersCount; i++)

{

    cout << idarr[i] << endl;

}

cout << " \n " << endl;

cout << "Operators: " << operatorsCount << endl;

cout << "list of operators: ";

// for loop for printing oparr

for (int i = 0; i < operatorsCount; i++)

{

    cout << oparr[i] << endl;
```

```
}

cout << " \n " << endl;

cout << "Symbols: " << symbolsCount << endl;

cout << "list of symbols: ";

// for loop for printing symarr

for (int i = 0; i < symbolsCount; i++)

{

   cout << symarr[i] << endl;

}

cout << " \n " << endl;

cout << "Constants: " << constantsCount << endl;

cout << "list of constants: ";

// for loop for printing conarr

for (int i = 0; i < constantsCount; i++)

{

   cout << conarr[i] << endl;

}

return 0;
```

```
}
```