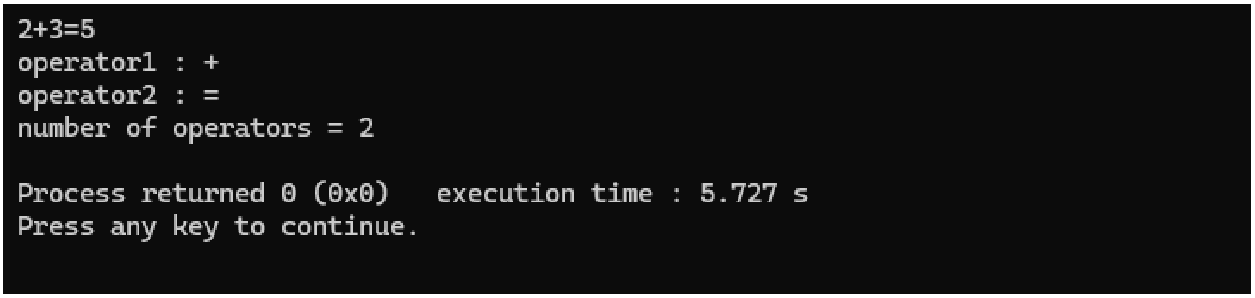
Task 1:

```
#include <iostream>

using namespace std;

int main()
{
  string line;
  int count=0;

  cin>>line;

  for(int i=0;i<line.length();i++){
    char ch=line[i];

    if(ch=='+'||ch=='-'||ch=='*'||ch=='/'||ch=='='||ch=='%'||ch=='/'||ch=='<'||ch=='>'||ch=='!'||ch=='&'||ch=='|'){
      count++;
      cout<<"operator"<<count<<" : "<<ch<<endl;
    }
  }

  cout<<"number of operators = "<<count<<endl;

  return 0;
}
```

Output:

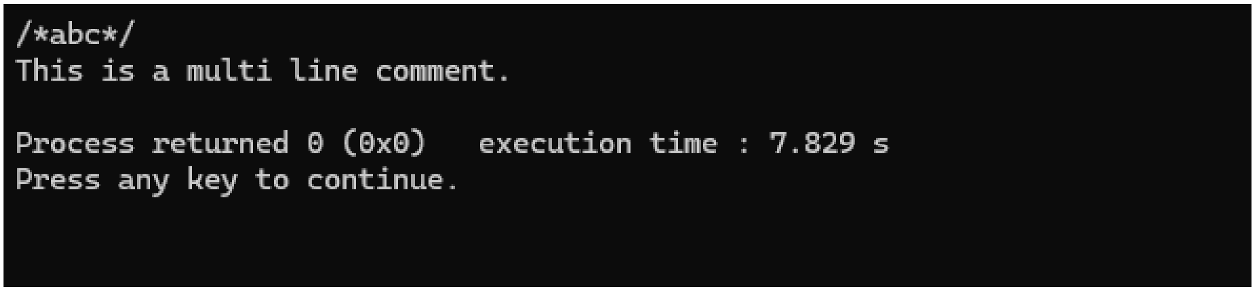


```
2+3=5
operator1 : +
operator2 : =
number of operators = 2

Process returned 0 (0x0)   execution time : 5.727 s
Press any key to continue.
```

Task 2:

```
#include <iostream>


using namespace std;


int main()
{
    string line;
    cin>>line;


    if(line.substr(0,2)=="//"){
        cout<<"This is a single line comment."<<endl;
    } else if(line.substr(0,2)=="/*" && line.substr(line.length()-2,2)=="*/"){
        cout<<"This is a multi line comment."<<endl;
    } else{
        cout<<"This is not a comment."<<endl;
    }


    return 0;
}
```

Output:

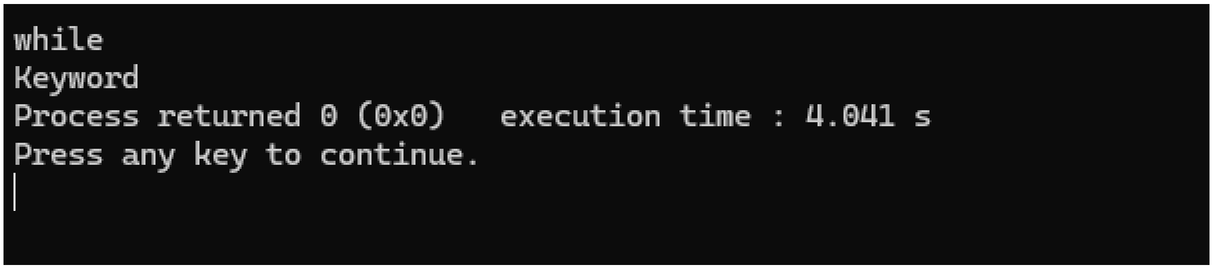


```
/*abc*/
This is a multi line comment.

Process returned 0 (0x0)   execution time : 7.829 s
Press any key to continue.
```

Task 3:

```
#include <iostream>

using namespace std;

int main() {
    string token;
    cin >> token;

    string keywords[] = {"int","float","double","char","if","else","while","return"};

    for (int i = 0; i < 8; i++) {
        if (token == keywords[i]) {
            cout << "Keyword";
            return 0;
        }
    }

    if (!(isalpha(token[0]) || token[0] == '_')) {
        cout << "Invalid";
        return 0;
    }

    for (int i = 1; i < token.length(); i++) {
        if (!(isalnum(token[i]) || token[i] == '_')) {
            cout << "Invalid";
            return 0;
        }
    }

    cout << "Identifier";
    return 0;
}
```

Output:



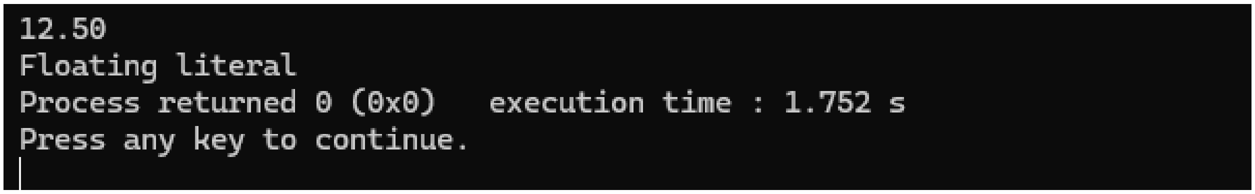
```
while
Keyword
Process returned 0 (0x0)   execution time : 4.041 s
Press any key to continue.
```

Task 4:

```
#include <iostream>

using namespace std;

int main()
{
    string token;
    cin >> token;

    int start = 0;
    int dotCount = 0;

    if (token[0] == '+' || token[0] == '-')
    {
```

```
    start = 1;
}

if (start == token.length())
{
    cout << "Invalid number";
    return 0;
}

for (int i = start; i < token.length(); i++)
{
    if (isdigit(token[i]))
    {
        continue;
    } else if (token[i] == '.')
    {
        dotCount++;
    } else
    {
        cout << "Invalid number";
        return 0;
    }
}

if (dotCount == 0)
{
    cout << "Integer literal";
    } else if (dotCount == 1)
    {
        int pos = token.find('.');

        if (pos > start && pos < token.length() - 1)
        {
            cout << "Floating literal";
        }
        else
        {
            cout << "Invalid number";
        }
    } else
    {
        cout << "Invalid number";
    }

    return 0;
}
```

Output:

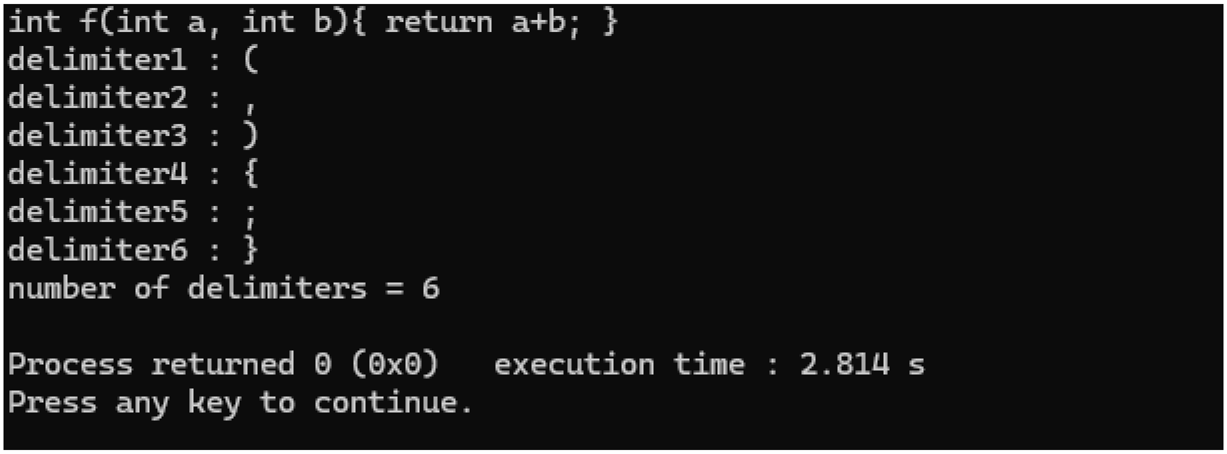


```
12.50
Floating literal
Process returned 0 (0x0)   execution time : 1.752 s
Press any key to continue.
```

Task 5:

```
#include <iostream>


using namespace std;


int main()
{
    string line;
    int count = 0;

    getline(cin, line);

    for (int i = 0; i < line.length(); i++)
    {
        char ch = line[i];

        if (ch == '(' || ch == ')' || ch == '{' || ch == '}' || ch == '[' || ch == ']' || ch == ',' || ch == ';')
        {
            count++;
            cout << "delimiter" << count << " : " << ch << endl;
        }
    }

    cout << "number of delimiters = " << count << endl;

}
```

Output:



```
int f(int a, int b){ return a+b; }
delimiter1 : (
delimiter2 : ,
delimiter3 : )
delimiter4 : {
delimiter5 : ;
delimiter6 : }
number of delimiters = 6

Process returned 0 (0x0)   execution time : 2.814 s
Press any key to continue.
```