Q3.) Write a program to find different tokens in a program

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
#include <cctype>

using namespace std;

class TokenDetector {
private:
    string str;

public:
    TokenDetector() {
        cout << "\n == Enter the string for token detection == \n";
        getline(cin, str);
        cout << "All Tokens are:\n";
        detectTokens();
    }

    bool isValidDelimiter(char ch) {
        const string delimiters = " +-*/,<;>=()[]{}";
        return delimiters.find(ch) != string::npos;
    }

    bool isValidOperator(char ch) {
        const string operators = "+-*/<=>";
        return operators.find(ch) != string::npos;
    }

    bool isValidKeyword(const string& word) {
        const vector<string> keywords = {
            "if", "else", "while", "do", "break", "continue", "int", "double", "float",
"return",
            "char", "case", "sizeof", "long", "short", "typedef", "switch", "unsigned",
            "void", "static", "struct", "goto"
        };
        return find(keywords.begin(), keywords.end(), word) != keywords.end();
    }

    bool isValidInteger(const string& word) {
        if (word.empty() || (!isdigit(word[0]) && word[0] != '-' && word[0] != '+')) return
false;
        return all_of(word.begin() + 1, word.end(), ::isdigit);
    }
```

```
    bool isRealNumber(const string& word) {
        bool hasDecimal = false;
        for (char ch : word) {
            if (!isdigit(ch)) {
                if (ch == '.' && !hasDecimal) hasDecimal = true;
                else return false;
            }
        }
        return hasDecimal;
    }

    bool isvalidIdentifier(const string& word) {
        if (word.empty() || isdigit(word[0]) || isValidDelimiter(word[0])) return false;
        return all_of(word.begin() + 1, word.end(), [](char ch) {
            return isalnum(ch) || ch == '_';
        });
    }

    void detectTokens() {
        int left = 0, right = 0;
        int length = str.length();

        while (right <= length && left <= right) {
            if (!isValidDelimiter(str[right])) right++;

            if (isValidDelimiter(str[right]) && left == right) {
                if (isValidOperator(str[right]))
                    cout << "Valid operator: '" << str[right] << "'\n";
                right++;
                left = right;
            } else if (isValidDelimiter(str[right]) && left != right || (right == length &&
left != right)) {
                string subStr = str.substr(left, right - left);

                if (isValidKeyword(subStr))
                    cout << "Valid keyword: '" << subStr << "'\n";
                else if (isValidInteger(subStr))
                    cout << "Valid Integer: '" << subStr << "'\n";
                else if (isRealNumber(subStr))
                    cout << "Real Number: '" << subStr << "'\n";
                else if (isvalidIdentifier(subStr))
                    cout << "Valid Identifier: '" << subStr << "'\n";
                else
                    cout << "Invalid Identifier: '" << subStr << "'\n";
                left = right;
            }
        }
    }
```

```
};

int main() {

    while(true){
        TokenDetector tkn_detect_obj;
    }

    return 0;
}
```

Output)

```
 == Enter the string for token detection ==
int main() {
All Tokens are:
Valid keyword: 'int'
Valid Identifier: 'main'

 == Enter the string for token detection ==
for(int i=0; i<5; i++){
All Tokens are:
Valid Identifier: 'for'
Valid keyword: 'int'
Valid Identifier: 'i'
Valid operator: '='
Valid Integer: '0'
Valid Identifier: 'i'
Valid operator: '<'
Valid Integer: '5'
Valid Identifier: 'i'
Valid operator: '+'
Valid operator: '+'

 == Enter the string for token detection ==
pass;}
All Tokens are:
Valid Identifier: 'pass'

 == Enter the string for token detection ==
return 0;}
All Tokens are:
Valid keyword: 'return'
Valid Integer: '0'
Invalid Identifier: ''
```