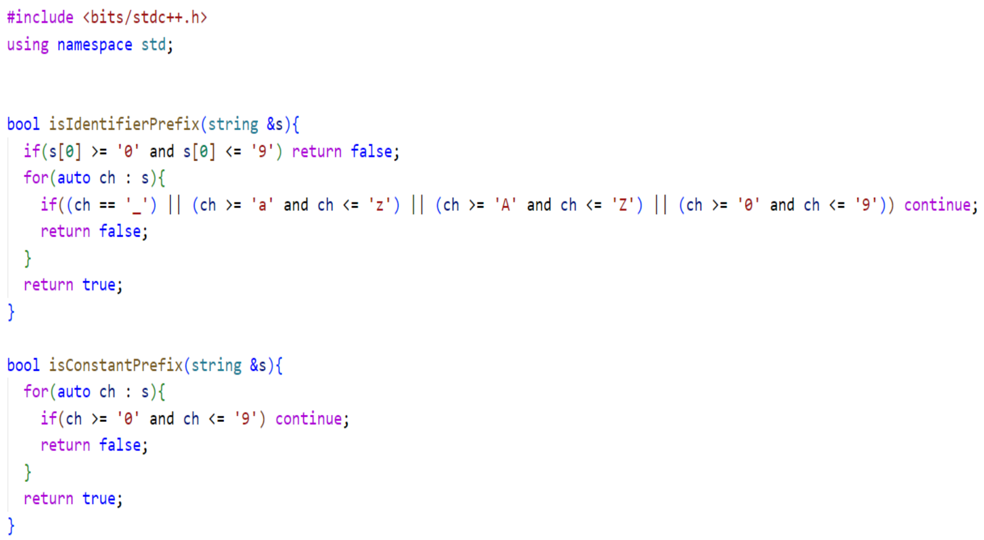
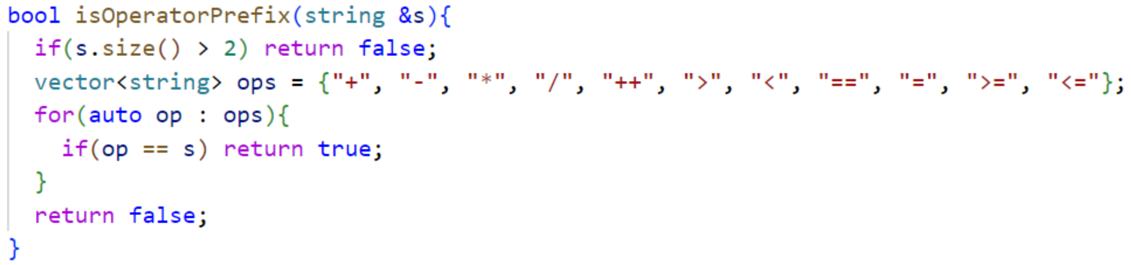
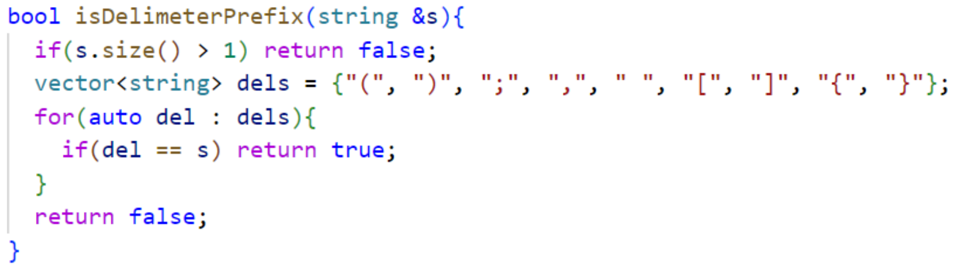
PRACTICAL



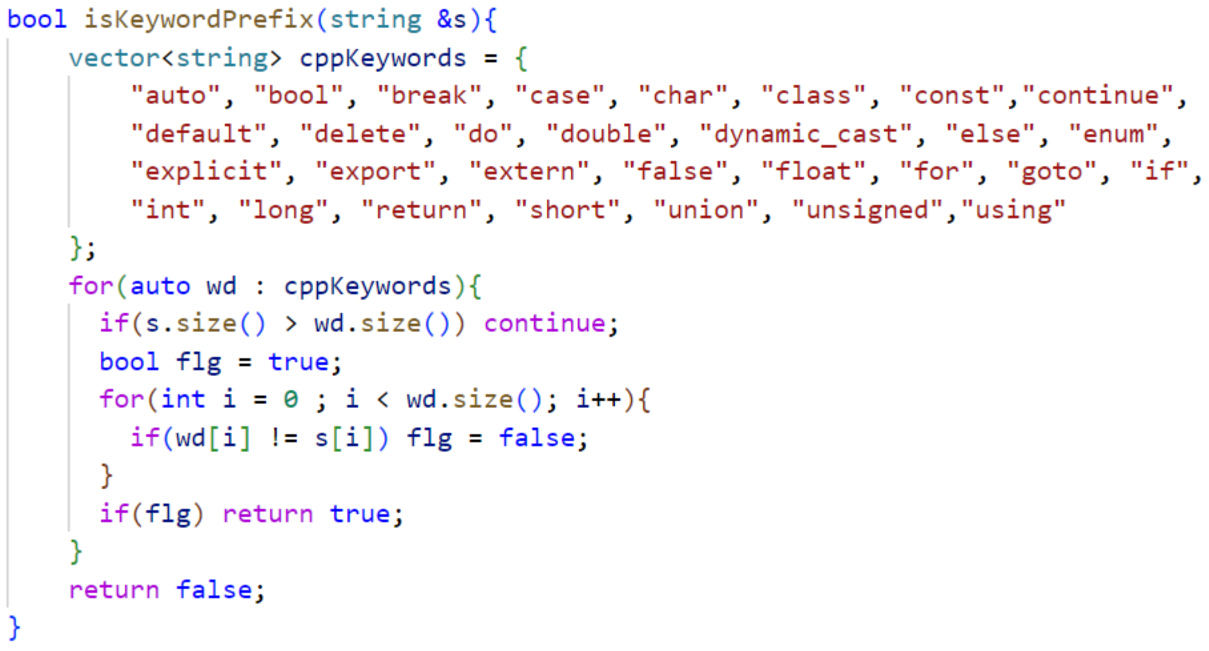
```cpp
#include <bits/stdc++.h>
using namespace std;


bool isIdentifierPrefix(string &s){
  if(s[0] >= '0' and s[0] <= '9') return false;
  for(auto ch : s){
    if((ch == '_') || (ch >= 'a' and ch <= 'z') || (ch >= 'A' and ch <= 'Z') || (ch >= '0' and ch <= '9')) continue;
    return false;
  }
  return true;
}

bool isConstantPrefix(string &s){
  for(auto ch : s){
    if(ch >= '0' and ch <= '9') continue;
    return false;
  }
  return true;
}
```



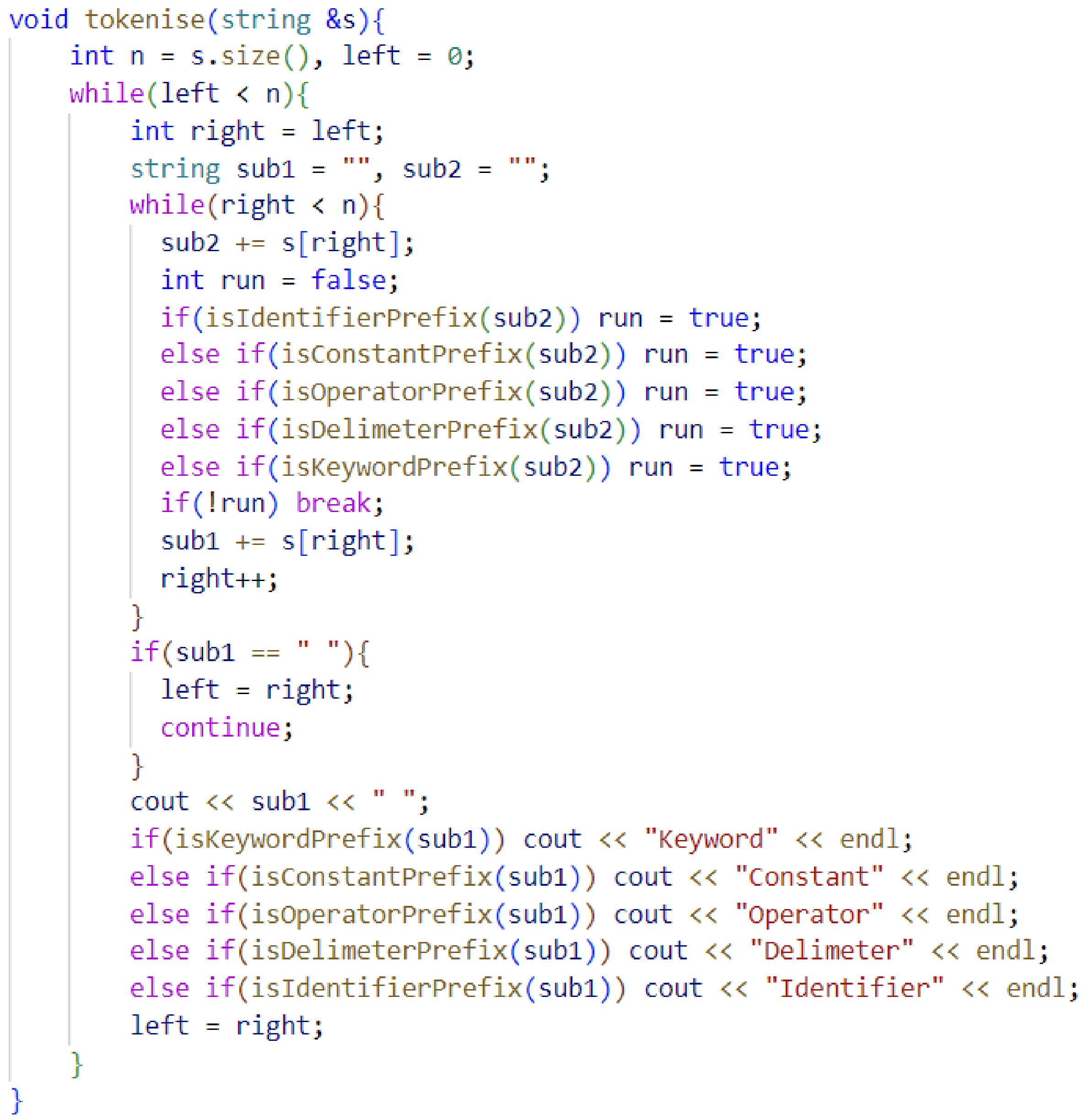
```cpp
bool isOperatorPrefix(string &s){
  if(s.size() > 2) return false;
  vector<string> ops = {"+", "-", "*", "/", "++", ">", "<", "==", "=", ">=", "<="};
  for(auto op : ops){
    if(op == s) return true;
  }
  return false;
}
```



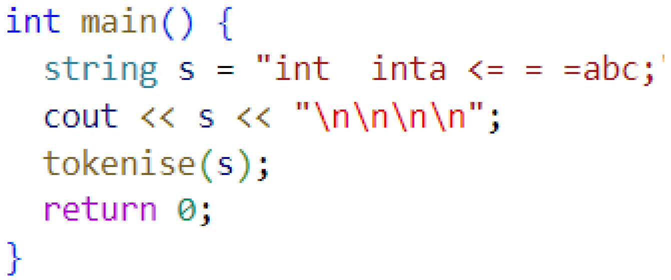
```cpp
bool isDelimeterPrefix(string &s){
  if(s.size() > 1) return false;
  vector<string> dels = {"(", ")", ";", ",", " ", "[", "]", "{", "}"};
  for(auto del : dels){
    if(del == s) return true;
  }
  return false;
}
```



```cpp
bool isKeywordPrefix(string &s){
    vector<string> cppKeywords = {
        "auto", "bool", "break", "case", "char", "class", "const","continue",
        "default", "delete", "do", "double", "dynamic_cast", "else", "enum",
        "explicit", "export", "extern", "false", "float", "for", "goto", "if",
        "int", "long", "return", "short", "union", "unsigned","using"
    };
    for(auto wd : cppKeywords){
      if(s.size() > wd.size()) continue;
      bool flg = true;
      for(int i = 0 ; i < wd.size(); i++){
        if(wd[i] != s[i]) flg = false;
      }
      if(flg) return true;
    }
    return false;
}
```



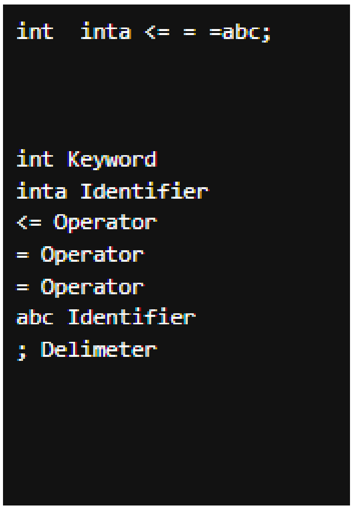
```
void tokenise(string &s){
    int n = s.size(), left = 0;
    while(left < n){
        int right = left;
        string sub1 = "", sub2 = "";
        while(right < n){
          sub2 += s[right];
          int run = false;
          if(isIdentifierPrefix(sub2)) run = true;
          else if(isConstantPrefix(sub2)) run = true;
          else if(isOperatorPrefix(sub2)) run = true;
          else if(isDelimeterPrefix(sub2)) run = true;
          else if(isKeywordPrefix(sub2)) run = true;
          if(!run) break;
          sub1 += s[right];
          right++;
        }
        if(sub1 == " "){
          left = right;
          continue;
        }
        cout << sub1 << " ";
        if(isKeywordPrefix(sub1)) cout << "Keyword" << endl;
        else if(isConstantPrefix(sub1)) cout << "Constant" << endl;
        else if(isOperatorPrefix(sub1)) cout << "Operator" << endl;
        else if(isDelimeterPrefix(sub1)) cout << "Delimeter" << endl;
        else if(isIdentifierPrefix(sub1)) cout << "Identifier" << endl;
        left = right;
    }
}
```



```
int main() {
  string s = "int  inta <= = =abc;'
  cout << s << "\n\n\n\n";
  tokenise(s);
  return 0;
}
```

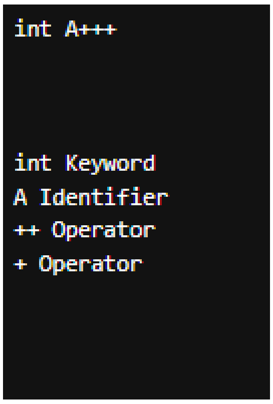
OUTPUT:

Testcase-1



```
int  inta <= = =abc;



int Keyword
inta Identifier
<= Operator
= Operator
= Operator
abc Identifier
; Delimeter
```

Testcast-2



```
int A+++



int Keyword
A Identifier
++ Operator
+ Operator
```