Date: 25/01/23

EXPERIMENT-1 LEXICAL ANALYSER

Faculty: DO: 2 Hour: 9 & 10

AIM: To write a program to implement a lexical analyser in c/c++.

Lexical Analyser: It reads the characters from source code and convert it into tokens.

ALGORITHM:

1. Start.
2. Get the input program from the file program.txt
3. Read the program line by line and check if each word in a line is either a keyword, identifier, math operators, logical operators, numerical value or others
4. for each lexeme read, generate a token as:
   a. If lexeme is a keyword, then token is keyword.
   b. If it is identifier, then token generated is printed on the console as identifier.
   c. In the same way math operator, logical, numerical values and other symbols are printed on the console.
5. The stream of tokens generated are displayed in the console output.
6. Stop the console.

PROGRAM CODE:

In C++

```
#include <iostream>
#include <fstream>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

using namespace std;

int iskeyword(char buffer[]){
char keywords[32][10] = { "auto", "break", "case", "char",
"const", "continue", "default", "do", "double", "else", "enum",
"extern", "float", "for", "goto", "if", "int", "long",
"register", "return", "short", "signed", "sizeof",
"static", "struct", "switch", "typeof", "union", "void",
"unsigned", "volatile", "while" };

int i, flag=0;
for(i=0; i<32; ++i){
if(strcmp(keywords[i], buffer)==0){
flag = 1;
break;
}}
return flag;
}
int main(){
char ch, buffer[15], operators[] = "+-*/%=";
ifstream fin("program.txt");
int i,j=0;
if(!fin.is_open()){
cout<<"error while opening the file\n";
```

INPUT:

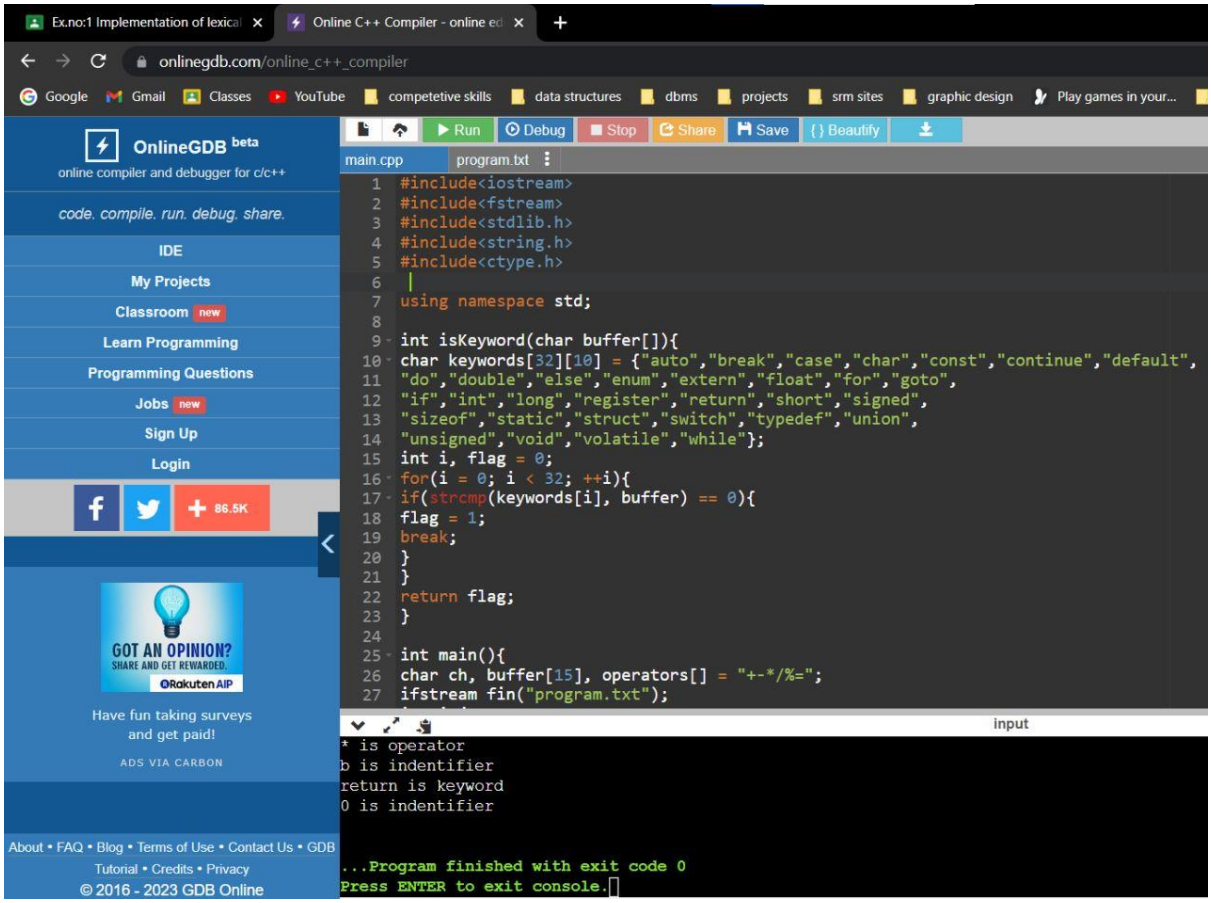
program.txt

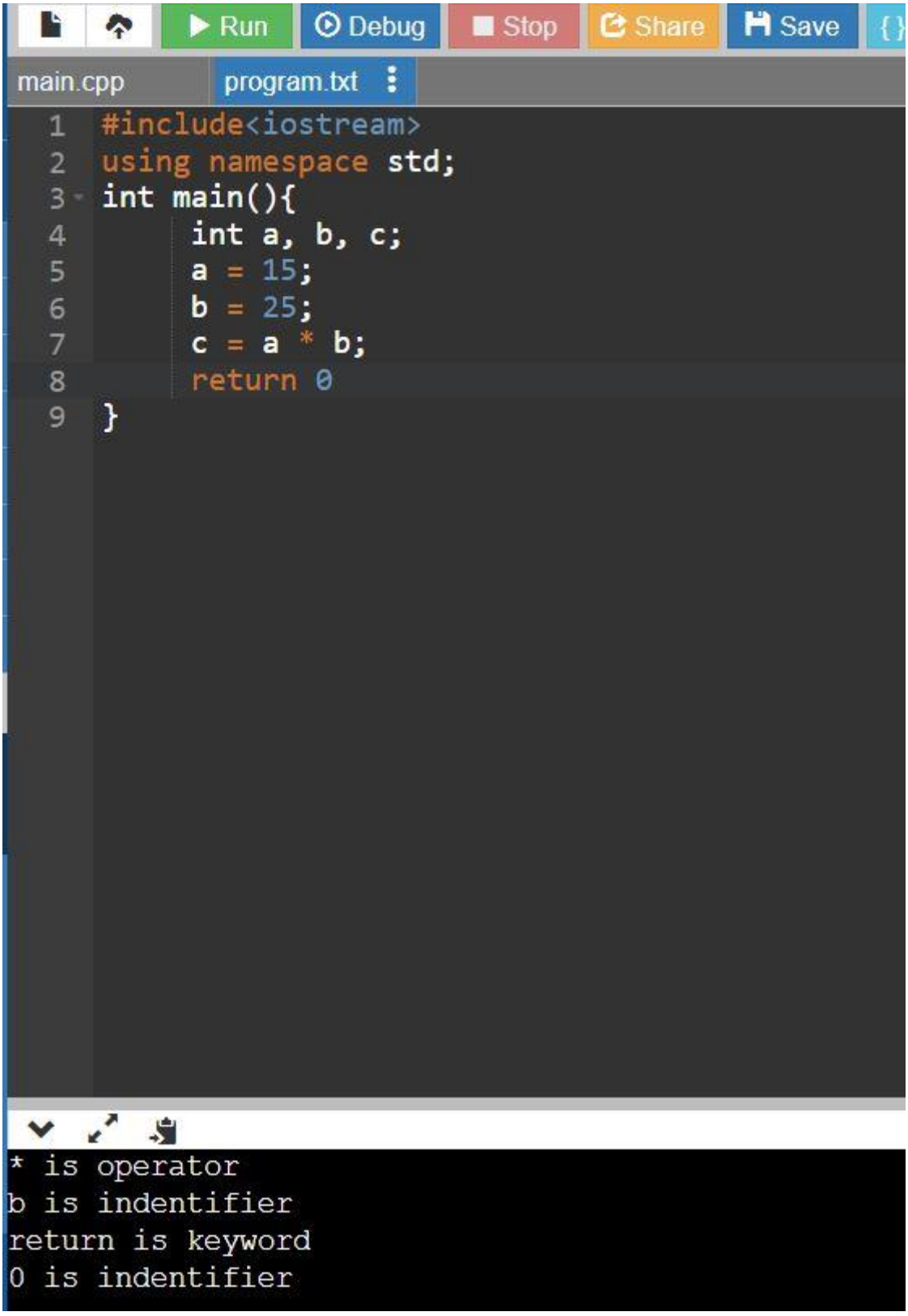
```
#include <iostream>
using namespace std;
int main() {
    int a, b, c;
    a = 15;
    b = 25;
    c = a * b;
    return 0;
}
```

CONSOLE OUTPUT:

```
* is operator
b is indentifier
return is keyword
0 is indentifier
```

RA2011028010099
DURGA CHANDANA SREE M.




Run
Debug
Stop
Share
Save
main.cpp
program.txt
#include<iostream>
using namespace std;
int main(){
int a, b, c;
a = 15;
b = 25;
c = a * b;
return 0
}
* is operator
b is indentifier
return is keyword
0 is indentifier

```
exit(0);
}
while (!fin.eof()) {
   ch = fin.get();

for(i=0; i<6; ++i) {
      if (ch== operators[i])
   cout << ch<< " is operator \n";
}
if (isalnum(ch)){
buffw [j++] = ch;
}
else if ((ch == ' ' || ch =='\n') && (j!=0)) {
buffer [j] = '\0';
j= 0;
 if (is-keyword (buffw) == 1)
cout <<buffw << " is ~~keyword~~ \n";
 else
cout << buffw << " is iridentifier \n";
 }
}
fin.close();
return 0;
}
```

RESULT: Hence the implimentation of lexical analyzer in c++ was compiled, executed successfully.