**ABHIJIT DASH**

**RA1811031010112**

**LEXICAL ANALYZER**

**AIM:-**

Implementation of Lexical Analyser by using C, C++, Java, or Python.

**CODING:-**

#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","typedef","union","unsigned","void","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";

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))

{

buffer[j++] = ch;

}

else if((ch == ' ' || ch == '\n') && (j != 0))

{

buffer[j] = '\0';

j = 0;

if(isKeyword(buffer) == 1)

cout<<buffer<<" is keyword\n";

else

cout<<buffer<<" is indentifier\n";

}

}

fin.close();

return 0;

}

**THE “PROGRAM.TXT” FILE CONTENT:-**

#include<stdio.h>

int main()

{

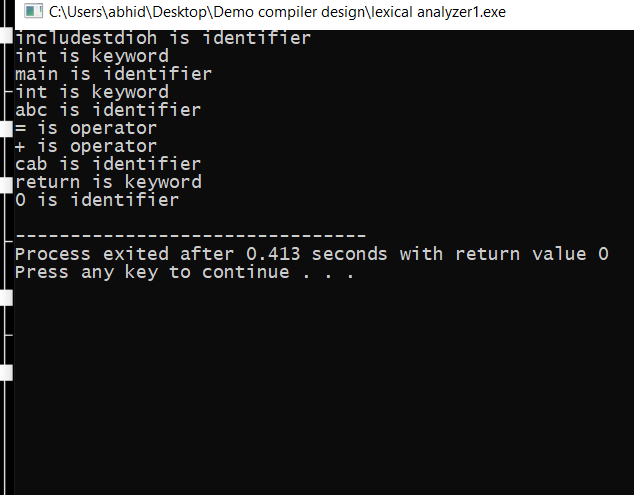
int a,b,c;

c=a+b;

return 0;

}

**OUTPUT SCREENSHOTS:-**



**RESULT:-**

The following code was compiled successfully.