**AIM:**

To implement lexical analyser .

**ALGORITHM:**

- Open the C file in read only format.
- Initialize the different types of tokens.
- Initialize the count values for each token.
- Initialize empty lists to store the tokens.
- Append the tokens into their respective lists and increment the count value for each token if that token is present in the initialized list.
- Print the count values and the tokens.

## PYTHON CODE

```
keywords =
{"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","printf","scanf","%d","#include","stdio
.h","main"}

operators = {"+","-","*","/","//","<",">","=","<=",">=","==","!=","++","--","%"}

delimiters = {'(',')','{','}','[',']','"',"'",';','#',',',''}

def detect_keywords(text):
        arr = []
        for word in text:
                if word in keywords:
                        arr.append(word)
        return list(set(arr))

def detect_operators(text):
        arr = []
        for word in text:
                if word in operators:
                        arr.append(word)
        return list(set(arr))

def detect_delimiters(text):
        arr = []
        for word in text:
                if word in delimiters:
                        arr.append(word)
        return list(set(arr))
```

```
def detect_num(text):
	arr = []
	for word in text:
		try:
			a = int(word)
			arr.append(word)
		except:
			pass
	return list(set(arr))
"""
this is original function for detecting identifier"""
def is_identifier(token):
   if token[0] in numbers or token in keywords:
      return False
   else:
      return identifier(token)
def identifier(token):
   if len(token)<2 and (token[0] in alphabets or token[0] in numbers or token[0] == "_"):
      return True
   elif token[0] in alphabets or token[0] in numbers or token[0] == "_":
      return identifier(token[1:])
   else:
      return False

def detect_identifiers(text):
	k = detect_keywords(text)
	o = detect_operators(text)
	d = detect_delimiters(text)
	n = detect_num(text)
	not_ident = k + o + d + n
	arr = []
	for word in text:
		if word not in not_ident:
			arr.append(word)
	return arr

file=input("Enter File Name for Lexical Analysis: ")
with open(file) as t:
	text = t.read().split()

print("Keywords: ",detect_keywords(text))
print("\nOperators: ",detect_operators(text))
print("\nDelimiters: ",detect_delimiters(text))
print("\nIdentifiers: ",detect_identifiers(text))
print("\nNumbers: ",detect_num(text))
```

## EXAMPLE FILES BEING READ FOR TESTING

## **FILE_1** – inputprogram.c

```c
#include < stdio.h > // This is a header file

void student ( )
{
    printf ( " My name is SHIVAM GUPTA " ) ;
    printf ( " My Roll Number is - RA1811003010074 " ) ;
}

int main ( )
{
    int a ;
    a = 10 ;
    printf ( " The value of a is - %d " , a ) ;
    return 0 ;
}
```

## **FILE_2** – Example.txt

Shivam Gupta

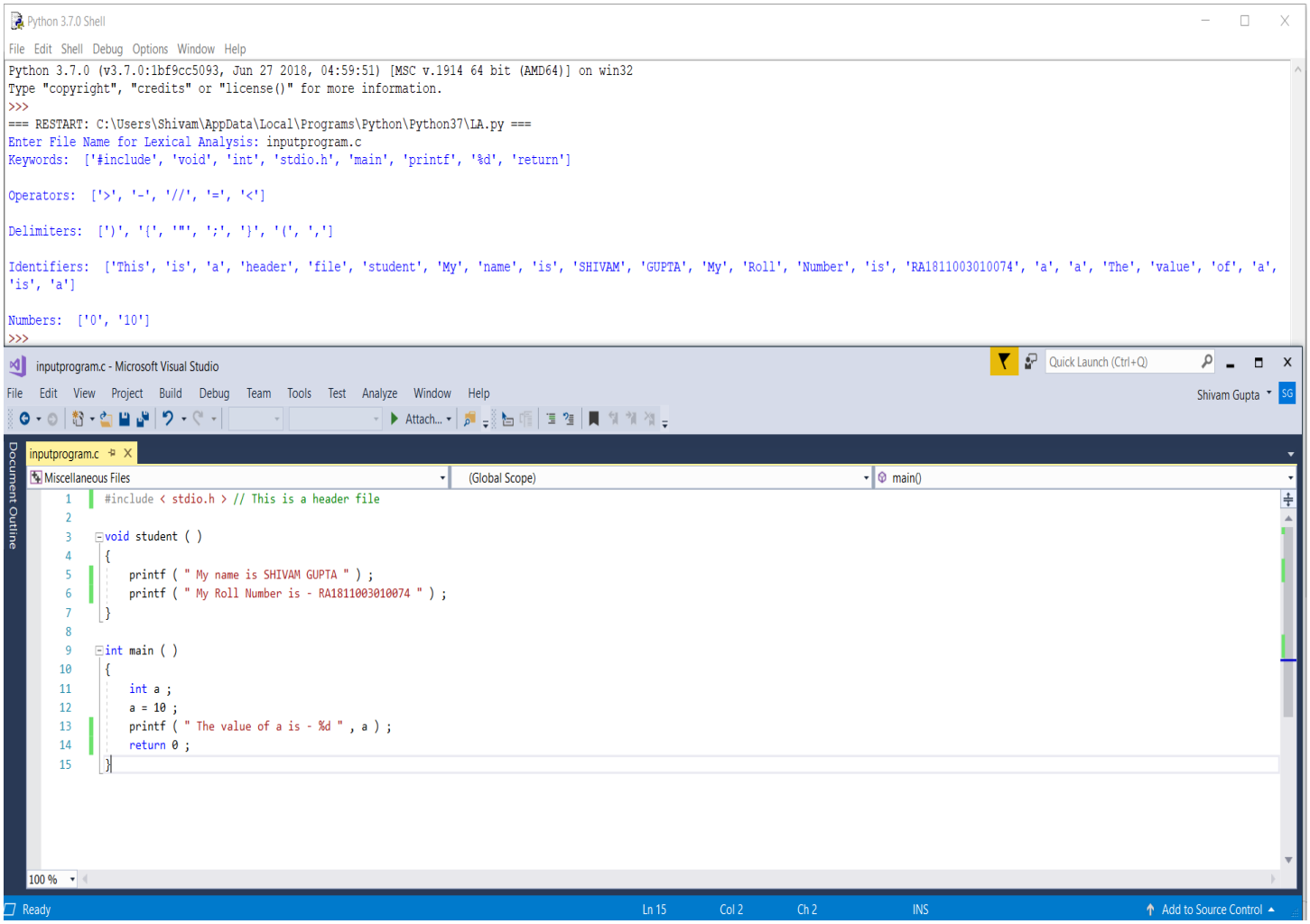
Class - BTech 3 Year

0074

---

---

## **IMPLEMENTATION**


Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\Shivam\AppData\Local\Programs\Python\Python37\LA.py ===
Enter File Name for Lexical Analysis: inputprogram.c
Keywords: ['#include', 'void', 'int', 'stdio.h', 'main', 'printf', '%d', 'return']
Operators: ['>', '-', '//', '=', '<']
Delimiters: [')', '{', '"', ';', '}', '(', ',']
Identifiers: ['This', 'is', 'a', 'header', 'file', 'student', 'My', 'name', 'is', 'SHIVAM', 'GUPTA', 'My', 'Roll', 'Number', 'is', 'RA1811003010074', 'a', 'a', 'The', 'value', 'of', 'a', 'is', 'a']
Numbers: ['0', '10']
>>>


inputprogram.c - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Shivam Gupta
Quick Launch (Ctrl+Q)
Attach...
Document Outline
inputprogram.c
Miscellaneous Files
(Global Scope)
main()
#include < stdio.h > // This is a header file
void student ( )
{
printf ( " My name is SHIVAM GUPTA " ) ;
printf ( " My Roll Number is - RA1811003010074 " ) ;
}
int main ( )
{
int a ;
a = 10 ;
printf ( " The value of a is - %d " , a ) ;
return 0 ;
}
100 %
Ready
Ln 15
Col 2
Ch 2
INS
Add to Source Control


## RESULT

Code was successfully implemented and the output was verified.