**University of Asia Pacific**

**Department of Computer Science & Engineering**

**CSE 430: Compiler Design**

**Lab #1 Output Sheet**

**Student Name:** Shawan Das

**Student ID:** 19101020

**Lab Section:** A1

Write a program for lexical analysis i.e. takes a line from file or keyboard and specify each word or character into the following tokens. The lexical analyzer should ignore redundant spaces, tabs and newlines. It should also ignore comments. You can try for identifying duplicate identifiers also.

· Any word either combination of characters and digits or combination of characters: **Identifier**.

· Any number : **Constant**

· Single character token :

o **Parenthesis** : ( ), { }, [ ]

o **Punctuation** sign : ;(semicolon) , : (colon) , , (coma)

o **Arithmetic Operator** : + , -, \*, /

· **Logical Operator** : >, >=, <, <=, ==, !=

· **Keyword**: There are total 32 keywords in C. They are:

| 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 |

**Sample Input (Console Input / File Input):**

void main()

{

Int a, b, c;

int a = b\*c + 10;

}

**Sample output:**

**Keyword (2)**: void, int

**Identifier (4)**: main, a, b, c

**Arithmetic Operator (3)**: =, \*, +

**Constant (1)**: 10

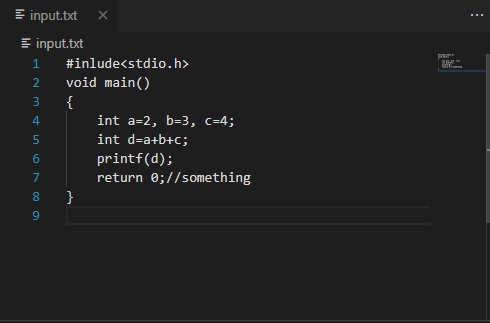
**Punctuation (2)**: , , ;

**Parenthesis (4):** (,), {,}

**Observed Input and Output:**

***//Put your output or screenshot here***

Input Data:



Output:

