Write a C/C++ program that will take a simple java program as input and identify different types of tokens. It can also return some simple syntactic errors like bracketing error, semicolon missing, operator-operand incorrect order, etc.

Input java program must contain single line comment and multi-line comment. Name your program as <registration number>.1.c or <registration number>.1.c++ like 254.1.c and upload only the c/c++ program text file.

NOTE: if you do not follow proper naming, your assignment may be cancelled.

Java supports following types of tokens which are:

1. Multi Line Comment
2. Single Line Comment
3. Keywords
4. Identifiers
5. Operators
6. Separator
7. Constant

List of Java keywords are listed below. You can use it in your program.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Abstract | boolean | Byte | Break | class |
| Case | catch | Char | Continue | default |
| Do | double | Else | Extends | final |
| Finally | float | For | If | implements |
| Import | instanceof | Int | Interface | long |
| Native | new | package | Private | protected |
| Public | return | Short | Static | super |
| Switch | synchronized | This | Throw | throws |
| Transient | try | Void | Volatile | while |
| Assert | const | Enum | Goto | strictfp |

Operators are listed below.

|  |  |
| --- | --- |
| *Arithmetic* | **+ , – , / , \* , %** |
| *Unary* | **++ , – – , !** |
| *Assignment* | **= , += , -= , \*= , /= , %= , ^=** |
| *Relational* | **==, != , < , >, <= , >=** |
| *Logical* | **&& , ||** |
| *Ternary* | **(Condition) ? (Statement1) : (Statement2);** |
| *Bitwise* | **& , | , ^ , ~** |
| *Shift* | **<< , >> , >>>** |

Separators are listed below.

(, ), [, ], {, }, ;, and ,

Example:

Input:

/\* This is a simple Java program.

FileName : "HelloWorld.java". \*/

class HelloWorld

{

// Your program begins with a call to main()

public static void main(String args[])

{

int a = 5;

a=a+1;

}

}

Output:

/\* This is a simple Java program. FileName : "HelloWorld.java". \*/ => Multi Line Comment

class => keyword

HelloWorld => identifier

{ => separator

// Your program begins with a call to main() => Single Line Comment

public => keyword

static => keyword

void => keyword

main => identifier

( => separator

String => keyword

args => identifier

[ => separator

] => separator

) => separator

{ =>separator

int => keyword

a => identifier

= => operator

5 => constant

; => separator

a => identifier

= => operator

a => identifier

+ => operator

1 => constant

; => separator

} => separator

} => separator