Bangladesh Agricultural University

Department of Bioinformatics Engineering

Course Code & Title: CSM 3222 Compiler Lab

Level-3, Semester-2, July-December/2024

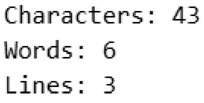
Deadline: 19 Nov, 2025 11:59 PM Lab 1 Tasks Platform: Google Classroom

1. Write a program that reads a source file and counts the total number of characters, words, and lines. The program must scan the file sequentially and handle spaces, tabs, and newline characters correctly. Display the final counts.

   Sample Input (input.txt):

   ```
   int main() {
       printf("Hello World\n");
   }
   ```

   Sample Output:

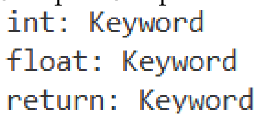
   ```
   Characters: 43
   Words: 6
   Lines: 3
   ```

2. Given an input line of C code, identify and print all the keywords present in the line. Non-keyword identifiers should not be printed.

   Sample Input:

   ```
   int main() { float num = 2.5;
       return 0; }
   ```

   Sample Output:

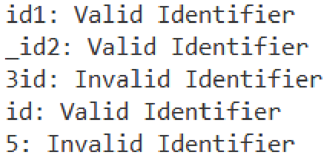
   ```
   int: Keyword
   float: Keyword
   return: Keyword
   ```

3. Write a program to extract all identifiers from a given line and classify them as valid or invalid. Valid identifiers must follow a regular expression. First, write the regular expression, then implement the expression in code to validate identifiers.

   Sample Input:

   ```
   Enter a line:
   id1, _id2, 3id, id 5
   ```

   Sample Output:

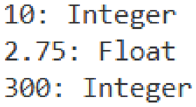
   ```
   id1: Valid Identifier
   _id2: Valid Identifier
   3id: Invalid Identifier
   id: Valid Identifier
   5: Invalid Identifier
   ```

4. Write a program to detect and classify numeric constants into integer and floating-point. First, write the regular expression, then implement the expression in code to classify numeric constants.

   Sample Input:

   ```
   Enter a line:
   a = 10; b = 2.75; c = 300;
   ```

   Sample Output:

   ```
   10: Integer
   2.75: Float
   300: Integer
   ```

5. Given an input line, identify all assignment, arithmetic, relational, and logical operators.

   Operators Include:

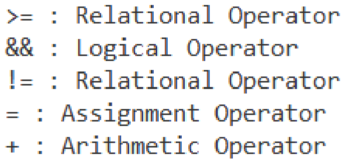
   - **Assignment:** =

- **Arithmetic:** +, -, *, /, %
- **Relational:** `==`, `!=`, `>`, `<`, `>=`, `<=`
- **Logical:** `&&`, `||`, `!`

Sample Input:

```
Enter a line:
if (a >= 10 && b != 5) a = a + 1;
```

Sample Output:



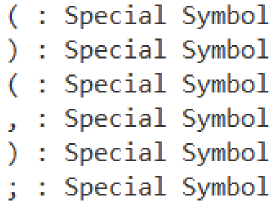
```
>= : Relational Operator
&& : Logical Operator
!= : Relational Operator
= : Assignment Operator
+ : Arithmetic Operator
```

6. Identify and print all punctuation/special symbols from the following set:
   { } ( ) ; , [ ] .

Sample Input:

```
Enter a line:
if(a%2==0) printf("%d is even", a
   );
```

Sample Output:



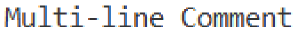
```
( : Special Symbol
) : Special Symbol
( : Special Symbol
, : Special Symbol
) : Special Symbol
; : Special Symbol
```

7. Write a program that detects and prints single-line comments (//...) and multi-line comments (/*...*/). Handle multiple comments in the same file and report unterminated comments as lexical errors.

Sample Input:

```
Enter a line or comment:
/* multi line comment */
```

Sample Output:



```
Multi-line Comment
```

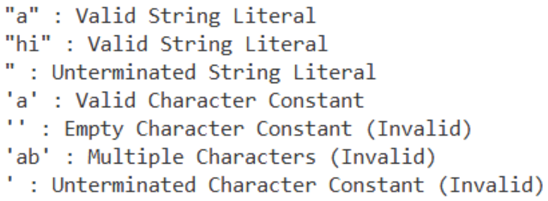
8. Write a program to detect valid and invalid string literals and character constants. Also detect errors such as:

   - Unterminated strings ”hello
   - Empty char constants ”
   - Multiple characters ’ab’

Sample Input (input.txt):

```
"a"
"hi"
"hello
'a'
''
'ab'
'a
```

Sample Output:



```
"a" : Valid String Literal
"hi" : Valid String Literal
" : Unterminated String Literal
'a' : Valid Character Constant
'' : Empty Character Constant (Invalid)
'ab' : Multiple Characters (Invalid)
' : Unterminated Character Constant (Invalid)
```