Indian Institute of Technology Roorkee

Department of Computer Science and Engineering

CSN-362: Compiler Laboratory (Spring 2024-2025)

Lab Assignment-4 (L4) **Date: 11 Feb 2025** **Duration: 3 hrs**

Problem Statement 1:

Write a C/C++ program for a lexical analyzer in case of C language. It should ignore redundant spaces, tabs and new lines to output the list of tokens after tokenization of a C program given as input.

Input:

Enter the C program:

```
int main()
{
int a=10,20;
charch;
float f;
}^Z
```

Output:

```
The numbers in the program are: 10 20
The keywords and identifiers are:
int is a keyword
main is an identifier
int is a keyword
a is an identifier
char is a keyword
ch is an identifier
float is a keyword
f is an identifier
Special characters are ( ) { = , ; ; ; }
Total no. of lines are: 5
List of Tokens: {…}
Total no. of tokens: …
```

Submission folder P1 should contains:

1. Source code file.
2. Testcases – C Program files.
3. Snapshot image files for at least 4 testcases after running the code on C program files taken as inputs.

Problem Statement 2:
FIRST and FOLLOW sets are used in compiler design and parsing techniques under syntax analysis, in case of LL(1) parsing for a given grammar.
Write a C/C++ program to calculate the FIRST and FOLLOW sets of non-terminals of any input grammar given as a set of Productions Rules.

Take the input as follows:
Input/Output: Enter the no. of productions: 10
Enter the productions:

```
E  -> TR
R  -> +T R| #
T  -> F Y
Y  -> *F Y | #
F  -> (E) | i

Output :
First(E)= { (, i, }
First(R)= { +, #, }
First(T)= { (, i, }
First(Y)= { *, #, }
First(F)= { (, i, }
-----------------------------------------------
Follow(E) = { $, ),  }
Follow(R) = { $, ),  }
Follow(T) = { +, $, ),  }
Follow(Y) = { +, $, ),  }
Follow(F) = { *, +, $, ),  }
```

Submission folder P2 should contains:
1. Source code file
2. Snapshot image file showing the outputs on the console after running the code on your C/C++ program file.