Department of Computer Science and Engineering, Dec 2024-April 2025

Compiler Design Practice (S6-B.Tech) -Assignment 2

Policies for Submission and Evaluation

You must submit your assignment in the moodle (Eduserver) course page, on or before the ubmission deadline. Also, ensure that your programs in the assignment must compile and execute without. During evaluation your uploaded programs will be checked. Failure to execute programs in the assignment without compilation errors may lead to zero marks for that program.

Your submission will also be tested for plagiarism, by automated tools. In case your code fails to pass the test, you will be straightaway awarded zero marks for this assignment and considered by the examiner for awarding **"F"** grade in the course. Detection of ANY malpractice regarding the lab course will also lead to awarding an **"F"** grade.

Last date for submitting : 10/01/2025 20:00 Hrs

Naming Conventions for Submission

Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar or .tar.gz).

The name of this file must be ASSG<NUMBER>_<ROLLNO>_<FIRSTNAME>.zip. (eg: ASSG1_118cs0006_LAXMAN.zip). DO NOT add any other files (like temporary files, inputfiles, etc.) except your source code, into the zip archive. The source codes must be named as ASSG<NUMBER>_<ROLLNO>_<FIRSTNAME>_<PROGRAM-NO>.<extension>. (For example: ASSG1_118cs0006_LAXMAN_1.c). If there are multiple parts for a particular question, then name the source files for each part separately as in ASSG1_118cs0006_LAXMAN_1b.c.

If you do not conform to the above naming conventions, your submission might not be recognized by some automated tools, and hence will lead to a score of 0 for the submission. So, **make sure that you follow the naming conventions.**

Problem Set:

1. Write a program to design Lexical Analyzer in C/C++ Language (to recognize any five keywords, identifiers, numbers, operators and punctuations).

Sample Input:
Enter number of lines: 3
int int nti iiit;
float hi 23.5
1a 2b e3;

Sample output:
Keywords: 3 (int int float)
Operators: 0
Constants: 1 (23.5)
Punctuations: 2 (; ;)
Identifiers: 3 (nti iiit hi)
Tokens: 9

2. Study the LEX and YACC tool and Evaluate an arithmetic expression with parentheses, unary and binary operators using Flex and Yacc.

3. Write a program in LEX to recognize Floating Point Numbers
Sample Input:
Enter any number:
12.4

Sample Output:
12.4 is a floating point number

4. Write a program in LEX to recognize different tokens: Keywords, Identifiers, Constants, Operators and Punctuation symbols.
Sample Input
Enter a statement
int float a1 25 b

Sample Output:
Number of Keywords:2
Number of Constants:1
Number of Identifiers:2
Number of Operators:0
Number of Puntuations:0
Total Number of Tokens are :5

5. Write a LEX program that copies file, replacing each nonempty sequence of white spaces by a single blank.

Sample Input File

```
Hello,    Friends
Service          to humanity
is
service to    divinity.
If
  you
     don't
        know
          how
              compiler works,
then
   you don't
know how
      computer works.
```

Sample Output File:

Hello, Friends Service to humanity is service to divinity. If you don't know how compiler works, then you don't know how computer works.

6. Write a LEX program to recognize the following tokens over the alphabets {0,1,..,9}

i. The set of all string ending in 00.
ii. The set of all strings with three consecutive 222's.
iii. The set of all string such that every block of five consecutive symbols contains at least two 5's.
iv. The set of all strings beginning with a 1 which, interpreted as the binary representation of an integer, is congruent to zero modulo 5.
v. The set of all strings such that the 10^{th} symbol from the right end is 1.
vi. The set of all four digits numbers whose sum is 9

The set of all four digital numbers, whose individual digits are in ascending order from left to right.

7. Write a LEX program that converts a file to “Pig Latin”. Specifically, assume the file is containing a sequence of words (groups of letters) separated by white space. Every time you encounter a word:
(a) If the first letter is a consonant, move it to the end of the word and then add ay.
(b) If the first letter is a vowel, just add ay to the end of the word. All non-letters are copied intact to the output.