**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**Group No.**

**47**

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS**

**Compiler Construction (CS F363)**

**II Semester 2022-23**

**Compiler Project (Stage-1 Submission)**

**Coding Details**

**(March 2, 2023)**

1. IDs and Names of team members

ID: 2019B3A70481P Name: PULKIT GUPTA

ID: 2020A7PS0001P Name: AISHWARYA SAM

ID: 2019B3A70666P Name: ROHAN KUNWAR

ID: 2019B3A70463P Name: AMAAN ZAFFAR

1. Mention the names of the Submitted files:

1\_\_\_\_driver.c\_\_\_\_\_\_\_\_\_ 7\_nonTerminals.txt\_ 13\_\_stackDef.h\_\_\_\_\_

2\_\_\_\_grammar.txt\_\_\_\_\_\_ 8\_\_\_parser.c\_\_\_\_\_\_\_ 14\_terminals.txt\_\_\_\_

3\_\_\_\_lexer.c\_\_\_\_\_\_\_\_\_\_\_ 9\_\_\_ parser.h\_\_\_\_\_\_\_ 15\_group\_47 coding details

4\_\_\_\_lexer.h\_\_\_\_\_\_\_\_\_\_\_ 10\_\_ parserDef.h\_\_\_\_ 16\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5\_\_\_\_lexerDef.h\_\_\_\_\_\_\_\_ 11\_\_\_stack.c\_\_\_\_\_\_\_\_ 17\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6\_\_\_\_makefile\_\_\_\_\_\_\_\_\_ 12\_\_\_stack.h\_\_\_\_\_\_\_\_ 18\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Total number of submitted files: \_\_15\_\_ (All files should be in **ONE folder** named exactly as Group\_#, # is your group number)
2. Have you mentioned your names and IDs at the top of each file (and commented well)? (Yes/ no) \_\_Yes\_\_ [Note: Files without names will not be evaluated]
3. Have you compressed the folder as specified in the submission guidelines? (yes/no) Yes
4. **Lexer Details:**
   1. Technique used for pattern matching: \_Finite Automaton has been used for pattern matching.
   2. DFA implementation (State transition using switch case, graph, transition table, any other (specify): DFA Implementation has been done through State Transition using Switch case.
   3. Keyword Handling Technique: Hash Table
   4. Hash function description, if used for keyword handling: Hashing Technique used involves linear probing.
   5. Have you used twin buffer? (yes/ no) No
   6. Lexical error handling and reporting (yes/No): Yes
   7. Describe the lexical errors handled by you: We are handling the following lexical errors –
5. Unidentified expression
6. Expression exceeds size limit.
   1. Data Structure Description for tokenInfo (in maximum two lines):

int linenumber

char array - tokenID

char array- lexeme

* 1. Interface with parser - getNextToken() in lexer.c gives us the token that we parse using the parseToken function in parser.c

1. **Parser Details:**

**[A]. High Level Data Structure Description (in maximum three lines each, avoid giving C definitions used):**

* + 1. grammar: It is a 1D Array of grammar nodes with each node containing a single non-terminal, its rules, FIRST set and FOLLOW set
    2. parse table: It is a 2D Array of grammar rules.
    3. parse tree: (Describe the node structure also) We haven’t implemented the parse tree yet.
    4. Parsing Stack node structure: It is a linked list-based stack implementation of strings (char Arrays)
    5. Any other (specify and describe): none.
  1. **Parse tree** 
     1. Constructed (yes/no): \_\_\_No\_\_\_\_
     2. Printing as per the given format (yes/no): Not implemented yet.
     3. Describe the order you have adopted for printing the parse tree nodes (in maximum two lines)

Not implemented yet.

**Grammar and Computation of First and Follow Sets**

* + 1. Data structure for original grammar rules: Linked List has been used for the original grammar rules.
    2. FIRST and FOLLOW sets computation automated (yes /no) Yes.
    3. Data structure for representing sets: Linked lists.
    4. Time complexity of computing FIRST sets: O (G.T) where G is the sum of the lengths of all productions and T is the number of non-terminals.
    5. Name the functions (if automated) for computation of First and Follow sets: calculateFirstAndFollows(). This function in turn calls the following functions-

calculateFirstSet(int ind)

calculateFollowSet(int ind)

* + 1. If computed First and Follow sets manually and represented in file/function (name that): not computed manually.
  1. **Error Handling** 
     1. Attempted (yes/ no): \_Yes\_
     2. Printing errors (All errors/ one at a time): \_Yes\_
     3. Describe the types of errors handled

Lexical errors - Unidentified expression and Expression exceeds size limit.

Syntax errors – Unexpected Expression

* + 1. Synchronizing tokens for error recovery (describe): Not implemented.
    2. Total number of errors detected in the given testcase t6(with\_syntax\_errors).txt- 14.

1. **Compilation Details:**
   1. makefile works (yes/no): Yes
   2. Code Compiles (yes/ no): Yes
   3. Mention the .c files that do not compile none- all .c files compile
   4. Any specific function that does not compile: none- all functions compile
   5. Ensured the compatibility of your code with the specified gcc version(yes/no): Yes.
2. **Driver Details**: Does it take care of the options specified earlier(yes/no): Yes.
3. **Execution**
4. status (describe in maximum 2 lines): the parser does not handle moduleReuseStmt rule properly, otherwise the execution status is- working.
   1. Execution time taken for
      * t1.txt (in ticks) \_\_\_\_\_ \_1658\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and (in seconds) \_0.000165\_
      * t2.txt (in ticks) \_\_\_\_\_ \_ 2347\_\_\_\_\_\_\_\_\_\_\_\_\_\_and (in seconds) \_0.000234\_
      * t3.txt (in ticks) \_\_\_\_\_\_\_2531\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and (in seconds) \_ 0.000253\_
      * t4.txt (in ticks) \_\_\_\_\_\_\_3453\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and (in seconds) \_0.000345\_
      * t5.txt (in ticks) \_\_\_\_\_\_\_2718 and (in seconds) \_0.000271\_
      * t6.txt (in ticks) \_\_\_\_\_\_\_3116 and (in seconds) \_0.000311\_
   2. Gives segmentation fault with any of the test cases (1-6) uploaded on the course page. If yes, specify the testcase file name: none.
5. Specify the language features your lexer or parser is not able to handle (in maximum one line): The lexer handles everything correctly but the parser does not handle moduleReuseStmt rule properly.
6. Are you availing the lifeline (Yes/No): Yes
7. Declaration: We, PULKIT GUPTA, AISHWARYA SAM, ROHAN KUNWAR AND AMAAN ZAFFAR, declare that we have put our genuine efforts into creating the compiler project code and have submitted the code developed only by our group. We have not copied any piece of code from any source. If our code is found plagiarized in any form or degree, we understand that disciplinary action as per the institute rules will be taken against us and we will accept the penalty as decided by the department of Computer Science and Information Systems, BITS, Pilani. [Write your ID and name below]

ID: 2019B3A70481P Name: PULKIT GUPTA

ID: 2020A7PS0001P Name: AISHWARYA SAM

ID: 2019B3A70666P Name: ROHAN KUNWAR

ID: 2019B3A70463P Name: AMAAN ZAFFAR

Date: \_3rd March 2023

---------------------------------------------------------------------------------------------------------------------------------------------

Should not exceed 4 pages.