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

Batch No. :

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS**

**Compiler Construction (CS F363)**

**II Semester 2017-18**

**Compiler Project (Stage-2 Submission)**

**Coding Details**

**(April 20, 2018)**

*Instruction: Write the details precisely and neatly. Places where you do not have anything to mention, please write NA for Not Applicable.*

1. ID Number: 2014B4A70867P

Name: Karan Makhija

1. Mention the names of the Submitted files ( Include Stage-1 and Stage-2 both)

1 lexer.c 7 ast.c 13 typecheck.c 19 codingDetails.docx 25 testcase6.txt

2 lexer.h 8 ast.h 14 typecheck.h 20 testcase1.txt

3 lexerDef.h 9 astDef.h 15 mygrammar.txt 21 testcase2.txt

4 parser.c 10 symboltable.c 16 makefile 22 testcase3.txt

5 parser.h 11 symboltable.h 17 AST.txt 23 testcase4.txt

6 parserDef.h 12 symboltableDef.h 18 driver.c 24 testcase5.txt

1. Total number of submitted files: 25 (All files should be in ONE folder named exactly as your ID)
2. Have you compressed the folder as specified in the submission guidelines? (yes/no) yes
3. **Status of Code development**: Mention 'Yes' if you have developed the code for the given module, else mention 'No'.
   1. Lexer (Yes/No): Yes
   2. Parser (Yes/No): Yes
   3. Abstract Syntax tree (Yes/No): Yes
   4. Symbol Table (Yes/ No): Yes
   5. Type checking Module (Yes/No): Yes
   6. Semantic Analysis Module (Yes/ no): Yes (reached LEVEL 1 as per the details uploaded)
   7. Code Generator (Yes/No): No
4. **Execution Status**:
   1. Code generator produces code.asm (Yes/ No): No
   2. code.asm produces correct output using NASM for testcases (C#.txt, #:1-3): No
   3. Semantic Analyzer produces semantic errors appropriately (Yes/No): Yes(not all)
   4. Type Checker reports type mismatch errors appropriately (Yes/ No): Yes
   5. Symbol Table is constructed (yes/no) Yes and printed appropriately (Yes /No): Yes
   6. AST is constructed (yes/ no) Yes and printed (yes/no) Yes
   7. Name the test cases out of 9 as uploaded on the course website for which you get the segmentation fault (testcase#.txt ; # 1-6 and c@.txt ; @:1-3): c2.txt
5. **Data Structures** (Describe in maximum 2 lines and avoid giving C definition of it)
   1. AST node structure : It contains token from lexeme, node from parsetree, startscope and endscope, to maintain scopes, datatype to store data types, assigned to check if the variables in output parameter lists are assigned and pointers to child, parent and sibling.
   2. Symbol Table structure: Symbol Table is a tree of hash tables. The tree node contains the function name, start and end scope of function, pointer to input and output parameter list, pointer to hashtable corresponding to the function, and child, parent and sibling pointer to traverse. The hash table contains the datatype, identifier name, lineno, offset, size of the variable.
   3. Matrix type expression structure: Same as Ast node
   4. Input parameters type structure: Same as Ast node
   5. Output parameters type structure: Same as Ast node
   6. Structure for maintaining the three address code(if created) : Not created
6. **Semantic Checks:** Mention your scheme NEATLY for testing the following major checks (in not more than 5-10 words)[ Hint: You can use simple phrases such as 'symbol table entry empty', 'symbol table entry already found populated', 'traversal of linked list of parameters and respective types' etc.]
   1. Variable not Declared : Symbol Table entry empty.
   2. Multiple declarations: Not implemented
   3. Number and type of input and output parameters: Not implemented
   4. assignment of value to the output parameter in a function: change the assigned integer to 1 after each assignment. In the end check if all are assigned.
   5. function call semantics: Not implemented
   6. type checking : Check datatype of ast nodes to compare left and right data type of each sub expression.
   7. return semantics: Not implemented
   8. Recursion : Not implemented
   9. module overloading: Not implemented
   10. 'If' semantics : Not implemented
   11. Matrix semantics and type checking of matrix type variables: Not implemented
   12. register allocation (your manually selected heuristic) : Not implemented
   13. Scope of variables and their visibility: Maintained start and end scope in Ast node as well as in HashTree node for each function.
7. **Compilation Details**:
   1. Makefile works (yes/No): Yes
   2. Code Compiles (Yes/ No): Yes
   3. Mention the .c files that do not compile: NA
   4. Any specific function that does not compile: NA
   5. Ensured the compatibility of your code with the specified gcc version(yes/no) No
8. **Driver Details**: Does it take care of the options specified earlier?(yes/no): Yes
9. Specify the language features your compiler is not able to handle (in maximum one line): NA
10. Are you availing the lifeline (Yes/No): Yes
11. Write exact command you expect to be used for executing the code.asm using NASM simulator [We will use these directly while evaluating your NASM created code]: Code generation not attempted
12. **Strength of your code**(Strike off where not applicable): (a) correctness ~~(b) completeness~~ (c) robustness ~~(d) Well documented~~ (e) readable (f) strong data structure (f) Good programming style (indentation, avoidance of goto stmts etc) (g) modular (h) space and time efficient
13. Any other point you wish to mention: The driver file does not execute the commands in a loop. For each option you need to execute the file again.
14. **Declaration:** I, Karan Makhija declare that I have put my genuine efforts in creating the compiler project code and have submitted the code developed by me. I have not copied any piece of code from any source. If my code is found plagiarized in any form or degree, I understand that a disciplinary action as per the institute rules will be taken against me and I will accept the penalty as decided by the department of Computer Science and Information Systems, BITS, Pilani.

Sign: kgmakhija

ID 2014B4A70867P

Name: Karan Makhija

Date: 21/04/18

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

/\*not to exceed three pages\*/