Batch No. :

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
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: 2014B3A70615P
2. Name: Amitojdeep Singh

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

1. lexerDef.h 7. astDef.h 13. codeGenerator.c 19 testcase3.txt 25. c3.txt
2. lexer.h 8 ast.h 14. driver.c 20 testcase4.txt 26. makefile
3. lexer.c 9 ast.c 15. Grammar.c21 testcase5.txt 27. codingDetails.docx
4. parerDef.h 10 symbolTypeCheckDef.h 16. codeGenerator.h 22 testcase6.txt
5. parser.h 11 symbolTypeCheck.h 17 testcase1.txt 23 c1.txt
6. parser.c 12 symbolTypeCheck.c 18 testcase2.txt 24. c2.txt

4. Total number of submitted files: 27 (All files should be in ONE folder named exactly as your ID)
5. Have you compressed the folder as specified in the submission guidelines? (yes/no) Yes

6. **Status of Code development**: Mention 'Yes' if you have developed the code for the given module, else mention 'No'.
   a. Lexer (Yes/No):Yes
   b. Parser (Yes/No): Yes
   c. Abstract Syntax tree (Yes/No): Yes
   d. Symbol Table (Yes/ No): Yes
   e. Type checking Module (Yes/No) Yes
   f. Semantic Analysis Module (Yes/ no):Yes (reached LEVEL 4 as per the details uploaded)
   g. Code Generator (Yes/No): Yes(Partial)
7. **Execution Status**:
   a. Code generator produces code.asm (Yes/ No):No
   b. code.asm produces correct output using NASM for testcases (C#.txt, #:1-3): No
   c. Semantic Analyzer produces semantic errors appropriately (Yes/No):Yes
   d. Type Checker reports type mismatch errors appropriately (Yes/ No): Yes
   e. Symbol Table is constructed (yes/no) Yes and printed appropriately (Yes /No): Yes
   f. AST is constructed (yes/ no) Yes and printed (yes/no) Yes
   g. 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): NA
8. **Data Structures** (Describe in maximum 2 lines and avoid giving C definition of it)
   a. AST node structure: type : 0 for NT 1 for T, val : enum of NT or T, token, parent:for traversal,child:leftmost child,next :sibling on the right, tempVar:temporaries for codegenration

   b. Symbol Table structure: Hashtable of nodes containing lexeme, type, width, offset, flag for assignment, matrix dimensions, pointer to symbol table of nested function. Also has pointer to symbol table of parent and own ast node.
   c. Matrix type expression structure: Linked list of rows, and each row has linked list of elements. Dimension, size and offset are as stored in symbol table
   d. Input parameters type structure: Linked list of parameter nodes with each node having type and id
   e. Output parameters type structure: Linked list of parameter nodes with each node having type and id
   f. Structure for maintaining the three-address code (if created): NA, Directly using pointers from ast nodes to temporaries in symbol table

9. **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.]
   a. Variable not Declared : symbol table entry empty
   b. Multiple declarations : symbol table entry already found populated
   c. Number and type of input and output parameters: traversal of linked list of parameters and respective types
   d. assignment of value to the output parameter in a function: flag for assignment in symbol table entry
   e. function call semantics: pointer to function’s symbol table is stored in symbol table of parent
   f. type checking : traversal of subtrees and checking corresponding types in symbol table
   g. return semantics: comparing ast structure of formal input and output parameters with number and type(from symbol table) of actual parameters
   h. Recursion : Check if name of called function is same as parent using symbol table
   i. module overloading: check symbol table of parent for symbol table entry already found populated
   j. 'If' semantics : type checking for operands of Boolean operators
   k. Matrix semantics and type checking of matrix type variables: Matrix is assigned size, offset and dimensions on assignment, compare symbol table entries while adding/subtracting, matrix elements are taken as integers
   l. register allocation (your manually selected heuristic) : NA
   m. Scope of variables and their visibility : symbol table entries

10. **Compilation Details**:
   a. Makefile works (yes/No): Yes
   b. Code Compiles (Yes/ No): Yes
   c. Mention the .c files that do not compile: NA
   d. Any specific function that does not compile: NA
   e. Ensured the compatibility of your code with the specified gcc version(yes/no) Yes

11. **Driver Details**: Does it take care of the options specified earlier?(yes/no): Yes

12. Specify the language features your compiler is not able to handle (in maximum one line)
In test case c2.txt parsing doesn't complete as grammar doesn't support matrix element in boolean expression.

13. Are you availing the lifeline (Yes/No): Yes

14. 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]
NA

---

15. **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

16. Any other point you wish to mention: Symbol table creation, type checking and semantic checks are implemented together in one file as they run in a single pass. Code generation is implemented only till creation of temporaries. Done additional checks for sizeof operator on matrix, strings and variables.

17. **Declaration:** I, Amitojdeep Singh (your name) 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.
ID 2014B3A70615P
Name: Amitojdeep Singh
Date: 21/4/18

/*not to exceed three pages*/