**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: 2015A7PS0127P

Name: SAMIP JASANI

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

1 \_HashTable.c 7 \_Trie.h 13 Grammer.txt 19 parser.h 25 codingDetails.docx

2 \_HashTable.h 8 \_Trie.c 14 lexer.c 20 parserDef.h 26 testcase1.txt

3 \_Stack.c 9 ast.c 15 lexer.h 21 semantic.c 27 testcase2.txt

4 \_Stack.h 10 ast.h 16 lexerDef.h 22 semantic.h 28 testcase3.txt

5 \_Tree.c 11 astDef.h 17makefile 23 symbolTable.c 29testcase4.txt

6 \_Tree.h 12 driver.c 18 parser.c 24 symbolTable.h 30 testcase5.txt

31 testcase6.txt

1. Total number of submitted files: \_\_\_\_\_\_31\_\_\_\_\_ (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 \_\_4\_\_ 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\_\_\_\_\_\_
   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):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_none\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. **Data Structures** (Describe in maximum 2 lines and avoid giving C definition of it)
   1. AST node structure: tree of nodes with node having a pointer to astnode or token ast node contains info about nonterminal node \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   2. Symbol Table structure: tree of nodes with node having pointer to hashtable head which has pointer to hashtable\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   3. Matrix type expression structure: it is stored using type MATRIX and int x and int y for its sizes
   4. Input parameters type structure: its a linked list containing id and type
   5. Output parameters type structure:its a linked list containing id and type
   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 not found in local and all parent scopes
   2. Multiple declarations: entry found in local scope symbol table
   3. Number and type of input and output parameters: parameter of function and variable traverse simultaneously checking type info and also checking number of them
   4. assignment of value to the output parameter in a function : output variable added to symbol table with initialisation flag = 0 which are set to 1 when assigned, when function ends we check initialisztion flags
   5. function call semantics: check if function is defined in its scope, check number of input-output parameters and type
   6. type checking : get type of lhs and compare it with type recoverd in rhs by using a binary recursive calls
   7. return semantics: stored as output linklist for each function
   8. Recursion : check if function is name of local or global scope
   9. module overloading: check if function is already defined in current scope
   10. 'If' semantics : check if boolean expression evaluated to bool which is intermediate type recovered using binary recursive calls
   11. Matrix semantics and type checking of matrix type variables: after normal MATRIX type checking, size check is implemented for matrix type variables.
   12. register allocation (your manually selected heuristic) :\_\_\_\_\_\_\_\_N\A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   13. Scope of variables and their visibility : using tree structure for symbol tables
7. **Compilation Details**:
   1. Makefile works (yes/No): yes
   2. Code Compiles (Yes/ No): yes
   3. Mention the .c files that do not compile: none
   4. Any specific function that does not compile: none
   5. Ensured the compatibility of your code with the specified gcc version(yes/no) yes
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)

\_\_\_\_\_handles everything\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Are you availing the lifeline (Yes/No): \_\_\_\_\_yes\_\_\_\_\_\_\_\_\_
2. 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]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_N/A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **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
2. Any other point you wish to mention: \_\_\_\_\_N\A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Declaration:** I, \_\_\_SAMIP JASANI\_\_\_ (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.

Sign:\_\_\_\_\_\_$@|\/|!|>\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ID\_\_\_\_\_\_\_2015A7PS0127\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name:\_\_\_SAMIP JASANI\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_21-April-2018\_\_\_\_\_\_\_\_\_\_\_

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

/\*not to exceed three pages\*/