# BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
## DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

**Compiler Construction (CS F363)**
**II Semester 2022-23**
**Compiler Project (Stage-2 Submission)**
**Coding Details**
**(April 12, 2023)**
**Group number : 34 (Write your group number here)**

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

1. IDs and Names of team members

| | |
|---|---|
| **ID: 2020A7PS0087P** | **Name: Aditya Kanthi** |
| **ID: 2020A7PS0131P** | **Name: Shirish Kumaravel** |
| **ID: 2020A7PS1210P** | **Name: Arvind Ram** |
| **ID: 2020A7PS1689P** | **Name: Aryan Kapoor** |

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

| | | | | | |
|---|---|---|---|---|---|
| 1.Lexer.c | 5.Parser.h | 9.ASTDef.h | 13.driver.c | 17.testcase3.txt | 21.testcase7.txt |
| 2.Lexer.h | 6.ParserDef.h | 10.symbolTable.c | 14.makefile | 18.testcase4.txt | 22.testcase8.txt |
| 3.LexerDef.h | 7.AST.c | 11.symbolTable.h | 15.testcase1.txt | 19.testcase5.txt | 23.testcase9.txt |
| 4.Parser.c | 8.AST.h | 12.symbolTableDef.h | 16.testcase2.txt | 20.testcase6.txt | 24.testcase10.txt |

3. Total number of submitted files: __24__ (All files should be in **ONE** folder named exactly as Group number)
4. Have you mentioned names and IDs of all team members at the top of each file (and commented well)? (Yes/ no) :**__YES__**
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):**__NO__**

7. **Execution Status**:
   a. Code generator produces code.asm (Yes/ No): **No**
   b. code.asm produces correct output using NASM for testcases (C#.txt, #:1-11): NA

c. Semantic Analyzer produces semantic errors appropriately (Yes/No): **Yes**

d. Static Type Checker reports type mismatch errors appropriately (Yes/ No): **Yes**

e. Dynamic type checking works for arrays and reports errors on executing code.asm (yes/no): **No**

f. Symbol Table is constructed (yes/no) **Yes** and printed appropriately (Yes /No) **Yes**

g. AST is constructed (yes/ no) **Yes** and printed (yes/no) **Yes**

h. Name the test cases out of 21 as uploaded on the course website for which you get the segmentation fault (t#.txt ; # 1-10 and c@.txt ; @:1-11):NA

8. **Data Structures** (Describe in maximum 2 lines and avoid giving C definition of it)
    a. AST node structure: Each node in our AST consists of an AST Node Identifier, input grammar symbol, line number, corresponding lexeme, integer/float value and the scope details for that particular node.
    b. Symbol Table structure: Global Symbol table hashed array of module entries, each module entry in turn links to its own Local Symbol table entry, and there are nested local tables for nested scopes
    c. array type expression structure: Have integer values storing array start, array end and array type variables to facilitate type checking
    d. Input parameters type structure: Linked list of parameter structs, located in the global symbol table inside each module entry
    e. Output parameters type structure: Linked list of parameter structs, located in the global symbol table inside each module entry
    f. Structure for maintaining the three address code(if created) : NA

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 : Check tree of local tables and global tables input and output list
    b. Multiple declarations: Global hash Symbol table entry already found.
    c. Number and type of input and output parameters: traversal of linked list
    d. assignment of value to the output parameter in a function:
    e. function call semantics: Check for declaration earlier in symbol Table and match input and out parameter list
    f. static type checking : Traversal of linked lists containing start and end index of array
    g. return semantics:
    h. Recursion : Cannot call current function name
    i. module overloading: Global Symbol table entry already present.
    j. 'switch' semantics : Search symbol table for id and match datatype with presence of default.
    k. 'for' and 'while' loop semantics: for loop ID checked for declaration and no reaasignent in loop.
    l. handling offsets for nested scopes: Send parent local table’s scope value to child
    m. handling offsets for formal parameters:Starting local table’s offset from width of input and output list
    n. handling shadowing due to a local variable declaration over input parameters: Add if variable not present in same scope/nesting level.
    o. array semantics and type checking of array type variables:

p. Scope of variables and their visibility :Line Numbers of START and END keyword

q. computation of nesting depth: Increment depth for every nesting call.

10. **Compilation Details**:
    a. Makefile works (yes/No):**<u>YES</u>**
    b. Code Compiles (Yes/ No):**<u>YES</u>**
    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 versions [GCC, UBUNTU, NASM] (yes/no)**<u>YES</u>**
11. Execution time for compiling the test cases [lexical, syntax and semantic analyses including symbol table creation, type checking and code generation] :

|  |  |  |
|---|---|---|
| i. | t1.txt (in ticks) : 3796 | and (in seconds) : 0.003796 |
| ii. | t2.txt (in ticks) : 4063 | and (in seconds) : 0.004063 |
| iii. | t3.txt (in ticks) : 5479 | and (in seconds) : 0.005479 |
| iv. | t4.txt (in ticks) : 4411 | and (in seconds) : 0.004411 |
| v. | t5.txt (in ticks) : 6357 | and (in seconds) : 0.006357 |
| vi. | t6.txt (in ticks) : 6488 | and (in seconds) : 0.006488 |
| vii. | t7.txt (in ticks) : 5530 | and (in seconds) : 0..05530 |
| viii. | t8.txt (in ticks) : 6081 | and (in seconds) : 0.006081 |
| ix. | t9.txt (in ticks) : 6512 | and (in seconds) : 0.006512 |
| x. | t10.txt (in ticks): 4625 | and (in seconds) : 0.004625 |

12. **Driver Details**: Does it take care of the **TEN** options specified earlier?(yes/no):**<u>YES</u>**
13. Specify the language features your compiler is not able to handle (in maximum one line)
    ______________________________________________________________________________
14. Are you availing the lifeline (Yes/No): **<u>YES</u>**
15. 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] **<u>NA</u>**
16. **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
17. Declaration: We, **Aditya Kanthi, Shirish Kumaravel, Arvind Ram and Aryan Kapoor** declare that we have put our genuine efforts in 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 a 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 names below]

|  |  |
|---|---|
| ID : **2020A7PS0087P** | Name: **Aditya Kanthi** |
| ID : **2020A7PS0131P** | Name: **Shirish Kumaravel** |
| ID : **2020A7PS1210P** | Name: **Arvind Ram** |
| ID : **2020A7PS1689P** | Name: **Aryan Kapoor** |

Date : **13th April 2023** Group number : **34**

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

Should not exceed 6 pages.