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 ______________37_______________(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:_2019B3A70351P____Name:____MANAF____________

   ID:___2019B3A70381P________Name:_NISHANT____

   ID:__2019B3A70443P________________Name:_ADARSH____

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

1___ast.c___________________	7___arraylist.c__________________	13____stack.c_____________	19_____hashmap.c______________
2_____variable_list.c________	8_____binary_operator.c______	14___assignment_statment.c______	20_________program.c____
3____array_variable_node.c____	9_____________________	15________________	21____________________
4______________________	10____________________	16_________________	22___________________
5______________________	11____________________	17_________________	23____________________
6______________________	12____________________	18__________________	24____________________

2. Total number of submitted files: ___________ (All files should be in **ONE** folder named exactly as Group number)
3. Have you mentioned names and IDs of all team members at the top of each file (and commented well)? (Yes/ no) _______Yes___ [Note: Files without names will not be evaluated]
4. Have you compressed the folder as specified in the submission guidelines? (yes/no)_______________

5. **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 ____ as per the details uploaded)
   g. Code Generator (Yes/No):________yes_________

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

7. **Data Structures** (Describe in maximum 2 lines and avoid giving C definition of it)
   a. AST node structure_Nodes inheriting from a base ast node____

   ____
   b. Symbol Table structure:__Tree of hashtables____
   c. array type expression structure:_________Operator, or unary operator or id or number node____
   d. Input parameters type structure:__List of variable node___________________
   e. Output parameters type structure:_List of variable nodes_________________________________
   f. Structure for maintaining the three address code(if created) :_____NA________________

8. **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 : not found in symbol table____
   b. Multiple declarations: already found in symbol table_________________________________
   c. Number and type of input and output parameters:__Type descriptor set ____
   d. assignment of value to the output parameter in a function __is_assigned filed in symbol table_____________________________
   e. function call semantics:__compare type descriptors____
   f. static type checking :___AST traversal during Semantic Analysis phase____
   g. return semantics:_____Compare type descriptors____
   h. Recursion :__Check caller's symbol table in tree of symbol table and see if the entry found is same as callee ____
   i. module overloading:___Check symbol table for already defined functions ____
   j. 'switch' semantics :______Type Descriptor of ID__________________________________
   k. 'for' and 'while' loop semantics: ____

l. handling offsets for nested scopes:____

m. handling offsets for formal parameters:____

n. handling shadowing due to a local variable declaration over input parameters:___________________
____

o. array semantics and type checking of array type variables: ______________________________

____

p. Scope of variables and their visibility :___From current ST to upwards____

q. computation of nesting depth:___Depth in tree of hashtables (aka cactus stack)____

9. Code Generation:
   a. NASM version as specified earlier used (Yes/no):____Yes____________________
   b. Used 32-bit or 64-bit representation:________________64_____________________________
   c. For your implementation: 1 memory word = ____________2________(in bytes)
   d. Mention the names of major registers used by your code generator:
      - For base address of an activation record: _____rbp_____________
      - for stack pointer:_____rsp_________
      - others (specify):_____________________________________
   e. Mention the physical sizes of the integer, real and boolean data as used in your code generation module
      size(integer): _________1__________(in words/ locations), __2_____________(in bytes)
      size(real): ____________2_____________(in words/ locations), 4_______________(in bytes)
      size(booelan): ________0.5______________(in words/ locations), ___1____________(in bytes)

   f. How did you implement functions calls?(write 3-5 lines describing your model of implementation)
      ____
      ____
      ____

   g. Specify the following:
      - Caller's responsibilities:____
      - Callee's responsibilities:____

   h. How did you maintain return addresses? (write 3-5 lines): _________________________________
      ____
      ____
      ____

   i. How have you maintained parameter passing? How were the statically computed offsets of the parameters used by the callee? ________________________

   j. How is a dynamic array parameter receiving its ranges from the caller? ______________________

   k. What have you included in the activation record size computation? (local variables, parameters, both): ____both____

   l. register allocation (your manually selected heuristic) :____
      ____

m. Which primitive data types have you handled in your code generation module?(Integer, real and boolean):__________Integer, boolean________________________________
n. Where are you placing the temporaries in the activation record of a function? __in the dynamic stack of the function________________
____

10. **Compilation Details**:
    a. Makefile works (yes/No):_____Yes______
    b. Code Compiles (Yes/ No):_______Yes_______
    c. Mention the .c files that do not compile:____None____
    d. Any specific function that does not compile:_____None___________________________________
    e. Ensured the compatibility of your code with the specified versions [GCC, UBUNTU, NASM] (yes/no)___Yes_________
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) ______________________ and (in seconds) _____________________
    ii. t2.txt (in ticks) ______________________ and (in seconds) _____________________
    iii. t3.txt (in ticks) ______________________ and (in seconds) _____________________
    iv. t4.txt (in ticks) ______________________ and (in seconds) _____________________
    v. t5.txt (in ticks) ______________________ and (in seconds) _____________________
    vi. t6.txt (in ticks) ______________________ and (in seconds) _____________________
    vii. t7.txt (in ticks) ______________________ and (in seconds) _____________________
    viii. t8.txt (in ticks) ______________________ and (in seconds) _____________________
    ix. t9.txt (in ticks) ______________________ and (in seconds) _____________________
    x. t10.txt (in ticks) ______________________ and (in seconds) _____________________

12. **Driver Details**: Does it take care of the **TEN** options specified earlier?(yes/no):___________
13. Specify the language features your compiler is not able to handle (in maximum one line) ____Module reuse____
14. Are you availing the lifeline (Yes/No): _No_____________
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]
    ____
    ____
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** ALL applicable
17. Any other point you wish to mention: __Data Structures have been appropriately made for various purposes that serve the purpose. __________________________
    ____
    ____
18. Declaration: We, ___Manaf, Nishant, Adarsh__________ (your names) 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______2019B3A70351P_________________________
Name:________________MANAF__________
ID__________________________2019B3A70381P____
Name:__NISHANT________________________
ID________2019B3A70443P______________________ Name:_
_______ADARSH________

Date: __12-04-2023____________ Group number _37__________________

Should not exceed 6 pages.