**CS445 – Compilers – Project (20 marks) 2021**

This project has two phases. Each team has 3-4 students. The project is to develop a pseudocode editor can fit with specific group of users such as children or blind people.

Each editor has syntax must be follow for each programming components: Primitive types, initialization, declaration, selection statements such as (if, nested if, if-else), expression statements, loop statements (for, while, do while) and Print function and read function.

You can also use comments and blocks.

The language is not case sensitive (small/capital letters are the same). Type casting is accepted for (float=integer only).

**Phase 1: Lexical analyser** (12 marks - **The deadline is 28th March 2022** )

1. GUI for typing the pseudocode and getting the results if there is any lexical errors. The errors should be described via meaningful messages with line number.
2. Tokenize the code. Spaces are not the only delimiters!
3. Write RE for each type.
4. Develop a function that convert each RE into DFA directly and catch the lexical errors if any.
5. Use the function for each token and generate sample table (token, line number, type) in a text file. The types will be used in the syntax parsing and CFG.
6. Design all context free grammars for your language**.** Each program should start with **Do:** then set of declarations, set of statements, and ends with **End.**
7. Test the lexical analyser with correct and incorrect code.
8. Report (RE, CFG, Code, your test and screenshot of the GUI) and submit the report and the code.

**Phase 2: Syntax analyser** (8 marks - **The deadline is** 22 May **2022**)

1. Develop an LR(1) parser for your grammar that developed in phase 1. (You are allowed to use ready tools for generating the parsing table). Report any syntax errors with line number.
2. Print the parsing table.
3. Test your code with correct and incorrect input code.
4. Connect your parser with the GUI.
5. Report (Code of the parser, parsing table, testing the codes, screenshot of the GUI)

**Phase 3: Semantic analyser** (3 marks bonus)

1. Write down all rules needed for type checking.
2. You have to modify the syntax parser to check the semantic (if declared or not, type checking and scope checking) using static scope checking. The program should print the symbol table (stack) in a text file.
3. For any error in semantic the parser should stop and report this error with meaningful message.

**Optional**

If you design this compiler to accept Arabic script or using any additional tools for input the code for the target users, you will gain 3 marks extra.

**Requirements:**

* Report for each phase + the code submitted in the blackboard. Trial of correct and incorrect input. Make sure that you cover all different cases that the user can use.
* You have to report all references and websites that assisting you in this project.
* You have to understand all the codes by all group members. Any member fails in answering any question about one of the task, all group members will gain zero.
* Any cheating without referencing will gain zero for all the project.
* One GUI for all phases is mandatory.

|  |  |  |
| --- | --- | --- |
|  | Marks | Date |
| Quiz1 | 10 | 22 march 2022 |
| Mid-exam1 | 20 | 10 April |
| Project phase1 | 10 | 28 March 2022 |
| Quiz2 | 10 | 15 May |
| Project phase2 | 10 | 22 May |
|  |  |  |