TOA Project List for Implementation:

1. **Regular Expression Scanner**
    - Implement a regular expression scanner for recognizing any programming language tokenizer (C, C++, JAVA, PHP, Python).
2. **Text Search Engine**
    - Develop a regular expression-based text search engine.
3. **Spelling Checker**
    - Construct a regular expression-based spelling checker for any text based on at least 100 English dictionary words.
4. **Lexical Analyzer for Mathematical Expressions**
    - Design a regular expression-based lexical analyzer for mathematical expressions, including parentheses.
5. **NFA Simulator**
    - Develop a simulator for a Non-Deterministic Finite Automaton (NFA).
6. **DFA Minimization Algorithm**
    - Implement a DFA minimization algorithm.
7. **Mealy to Moore Machine Converter**
    - Construct a converter for converting a Mealy machine to a Moore machine and vice versa, based on user-provided data.
8. **Regular Grammar Parser**
    - Design and implement a regular grammar parser.
9. **Context-Free Grammar (CFG) for Present Tense**
    - Design and simulate a context-free grammar (CFG) to recognize natural language in the Present Tense for at least 100 verbs.
10. **Context-Free Grammar (CFG) for Past Tense**
    - Design and simulate a context-free grammar (CFG) to recognize natural language in the Past Tense for at least 100 verbs.
11. **Context-Free Grammar (CFG) for Future Tense**
    - Design and simulate a context-free grammar (CFG) to recognize natural language in the Future Tense for at least 100 verbs.
12. **CFG Parser for Programming Constructs**
    - Develop a CFG parser to recognize at least 10 programming constructs (e.g., if-else, for loop, do-while, etc.).
13. **LL(1) Parsing Technique Simulation**
    - Simulate CFG parsing techniques using LL(1) for any programming language.
14. **PDA Simulator**
    - Develop a simulator for a Pushdown Automaton (PDA).
15. **PDA to CFG Converter**
    - Build a converter from a PDA to a CFG based on user-provided data.
16. **Multi-Tape Turing Machine Simulator**
    - Design and simulate multi-tape Turing machines based on customized user-provided arithmetic functions.
17. **Turing Machine for Arithmetic Operations**
    - Design and simulate a Turing machine for performing arithmetic operations with at least 5 user-provided arithmetic operators.

18. **Turing Machine for String Manipulation**
    - Develop a Turing machine for string manipulation tasks, such as reversing a string, changing the case, etc., with at least 5 functions.

Please submit your group selections on the Google Sheet Link by the specified deadline. Should you have any questions or require clarification on any of the projects, feel free to reach out.

Note: Please note that if you have any new motivated ideas from real-world and you believe that the scope of the project is either equal or greater than the above projects, kindly discuss them with your instructor. Any new idea will be applicable if approved by Ms. Shaharbano.

Kindly refrain from having same projects among two different teams. If you feel the remaining projects are not covered in course so far, just do a little research and it will be highly encouraging and easily scorable if you are able to pick those projects and work it out correctly.

Best regards,

Muhammad Shariq Nadeem

K224285@nu.edu.pk

TA, Section: BCS-4J