**BOWEN UNIVERSITY, IWO**
COLLEGE OF COMPUTING AND COMMUNICATION STUDIES
COMPUTER SCIENCE PROGRAMME
B.Sc. SECOND SEMESTER EXAMINATION 2022/2023 SESSION
COURSE CODE: CIT 306 COURSE TITLE: COMPILER CONSTRUCTION I
COURSE CREDITS: 3   DATE :   TIME: 2½ hours
INSTRUCTION: Attempt any four questions

1a. Write a short note on language processing.   **(4 marks)**
1b. Write short notes on the types of compiler you know.   **(8 marks)**
1c. Contrast a compiler with an interpreter.   **(4 marks)**
1d. The object code generated by a compiler mainly consists mainly of the code written by the programmer. Justify or dispute this assertion.   **(3 marks)**
1e. Distinguish between the functions of the analysis and synthesis phases of a compiler.
                                                                                        (6 marks)

2a. Differentiate between a token and a lexeme. Hence, classify the following into tokens and lexemes: **interest = (principal * rate * time) / 100.**   **(7 marks)**
2b. You just developed a new high level programing language. How would you define the language towards writing a compiler for it?   **(6 marks)**
2c i. What is the main usefulness of finite automata?   **(2 marks)**
ii. Which kinds of strings are accepted by the automaton below?   **(4 marks)**
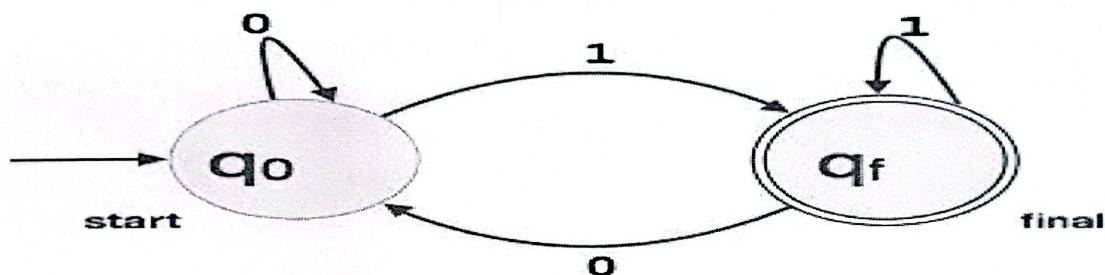
2d. Draw a finite automata that can recognize the following: **(4 marks)**
0110, 0000, 0100, 0010
2e. Why is code optimization important in compiler construction?   **(2 marks)**

3a. Given the grammar $G = (V, \Sigma, P, S)$, where
$V = \{ Q, Z, N \}$
$\Sigma = \{ 0, 1 \}$
$P = \{ Q \rightarrow Z \mid Q \rightarrow N \mid Q \rightarrow \Sigma \mid Z \rightarrow 0Q0 \mid N \rightarrow 1Q1 \}$
$S = \{ Q \}$
Show how 00100 is generated.   **(6 marks)**
3b.
Given the production rules:
$E \rightarrow E + E$
$E \rightarrow E * E$
$E \rightarrow id$

Show how the expression **id * id * id + id** is derived using leftmost derivation.   **(7 marks)**
3c. As a system programmer, you have been assigned the duty of writing semantics analyzer for the job at hand. What are the various errors you will need to take care of?   **(5 marks)**
3d. Why constructing a symbol table and how can you implement one?   **(7 marks)**

4a. Write short notes on the following as pertaining to finite state automata: **(10 marks)**
i. State ii. Start State iii. Intermediate States   iv. Final State v. Transition
4b. List and write short notes on the components of a context free grammar   **(12 marks)**
4c. Write briefly on ambiguity with an appropriate example of the kind of production that can lead to ambiguity.   **(3 marks)**

5a. Given the grammar B => Bb |∝, write an equivalent grammar by removing the left recursion.
    (4 marks)
5b. What is predictive parsing?   **(2 marks)**
5c. What is the difference between predictive parsing and recursive descent parsing? **(3 marks)**
5d. List the kinds of errors that a program can have.   **(4 marks)**
5e. List and write short notes on the different error recovery strategies that can be employed by parsers.   **(12 marks)**

6a. List the different ways of representing intermediate code.   **(3 marks)**
6b. Draw the tables representing the quadruples and triples of the statement: **(10 marks)**
    y = mx + c
6c. Why in your opinion is it necessary to translate the source code into an intermediate code before it is then translated to its target code?   **(6 marks)**
6d. Optimize the following code snippet by eliminating the dead code: **(6 marks)**
int global;
void f ()
{
  int i;
  i = 1;
  global = 1;
  global = 2;
  return;
  global = 3;
}