

Seat No.: _____

Enrolment No. _____

GUJARAT TECHNOLOGICAL UNIVERSITY

BE – SEMESTER- VII EXAMINATION-SUMMER 2023

Subject Code: 2170701

Date: 27/06/2023

Subject Name: Compiler Design

Time: 10:30 AM TO 01:00 PM

Total Marks: 70

Instructions:

1. Attempt all questions.
2. Make suitable assumptions wherever necessary.
3. Figures to the right indicate full marks.
4. Simple and non-programmable scientific calculators are allowed.

	MARKS
Q.1	
(a) Define the following terms:	03
(i) Parse tree	
(ii) Annotated parse tree	
(iii) Syntax tree	
(b) Explain finite Automata in brief and also explain its role in lexical	04
(c) Explain various Phases of compiler with neat diagram and suitable examples.	07
Q.2	
(a) What is ambiguous grammar? Explain your answer with proper example.	03
(b) Explain input buffering with suitable example.	04
(c) Construct $\wedge$ -NFA for following regular expression and convert it into DFA.	07
$(a/b)ca^*c\#$	
OR	
(c) Convert the following regular expression to DFA without using $\wedge$ -NFA.	07
$(a/b)^*abb\#$	
Q.3	
(a) Write a LEX program to count the words entered by the user. i.e. if User enters text: Hi Hello, output should be 2.	03
(b) $S \rightarrow (L) \mid a$ $L \rightarrow L, S \mid S$ 1) What are the terminals, non-terminals, and start symbol? 2) Find parse trees for the following sentences: i. (a, a) ii. (a, (a, a)) iii. (a, ((a, a), (a, a)))	04
(c) Explain Syntax Directed Definitions (SDD). Also Explain Synthesized and Inherited attributes with example.	07
OR	
Q.3	
(a) Explain role of intermediate code in compilers.	03
(b) Consider the given grammar below: $S \rightarrow aAC \mid bB$ $A \rightarrow eD$ $D \rightarrow bE \mid \epsilon$ $E \rightarrow eD \mid dD$ $B \rightarrow f \mid g$ $C \rightarrow h \mid i$	04

- Obtain the First and Follow for each non terminal present in the grammar.
- (c) Construct the SLR parsing table for the following grammar. **07**
 $S \rightarrow 0S0$
 $S \rightarrow 1S1$
 $S \rightarrow 10$
- Q.4** (a) Enlist various Storage allocation strategies. Explain any one of them in detail. **03**
 (b) Explain error recovery strategies in brief. **04**
 (c) Explain working of an operator precedence parser. Construct precedence graph and precedence table for Operators id, +, *, ^, (,), \$. **07**
- OR**
- Q.4** (a) Explain various parameter passing techniques. **03**
 (b) Explain the various methods of implementing three address statements. **04**
 (c) Explain various data structures used in symbol table management. **07**
- Q.5** (a) Define: DAG, Basic Block, flow graph **03**
 (b) Explain how following expression can be converted in a DAG: $a+b*(a+b)+c+d$ **04**
 (c) Describe various Peephole Optimization techniques, each with suitable example. **07**
- OR**
- Q.5** (a) Enlist the various loop optimization techniques. **03**
 (b) Explain Activation record in detail. **04**
 (c) Describe various issues related to the code generation in compiler. **07**
