Seat No.: _____ Enrolment No._____

GUJARAT TECHNOLOGICAL UNIVERSITY

ME - SEMESTER– II (Old course)• REMEDIAL EXAMINATION – SUMMER 2015

Subject Code: 1720202 **Date:13/05/2015**

Subject Name: Design of Language Processors

Time: 02:30 pm to 5:00 pm **Total Marks: 70**

Instructions:

1. **Attempt all questions.**
2. **Make suitable assumptions wherever necessary.**
3. **Figures to the right indicate full marks.**

Q1	**(a)**	**(i)**	What do you understand by operator precedence parsing? Parse the following string giving the diagrammatic trace of the algorithm. $<id>_a * <id>_b / <id>_c$	**07**
	(b)	**(i)**	Write three address codes for following expression and generate final code by clearly showing register descriptors and address descriptors. x = a / (b - c) + d * (e - f)	**04**
		(ii)	Explain peephole optimization.	**03**
Q2	**(a)**	**(i)**	Write a regular expression for 1. The set of strings of 0ôs and 1ôs not containing 11001 as a substring. 2. The set of strings with odd number of aôs followed by even number of bôs	**04**
		(ii)	Explain code improving transformations.	**03**
	(b)	**(i)**	Construct a SLR parsing table for following grammar. S Ÿ xRy | xTy | xRz R Ÿ aS | b T Ÿ b	**07**
			OR	
	(b)	**(i)**	Construct a CLR parsing table for following grammar. S Ÿ Ba | bBc | dc | bda B Ÿ d	**07**
Q3	**(a)**	**(i)**	What is a annotated parse tree? Explain with example.	**04**
		(ii)	Explain first three phases and error handling of a compiler. Give appropriate examples.	**03**
	(b)	**(i)**	Explain Simple LR parser	**04**
		(ii)	List the major steps of relocation and linking algorithms. Explain in brief.	**03**
			OR	
Q3	**(a)**	**(i)**	Explain macros and macro processors for the assembler.	**04**
		(ii)	Explain synthesis and analysis phase of assembler.	**03**
	(b)	**(i)**	What is an ambiguous grammar? Why is it called so? Justify the ambiguity with appropriate example.	**03**
		(ii)	Explain macro call and macro expansion. How do macros differ from Subroutine? Explain.	**04**
Q4	**(a)**	**(i)**	What is DAG? How it differs from syntax tree? Explain.	**04**
		(ii)	Write the skeletal of the Parser which performs parsing without backtracking	**03**
	(b)	**(i)**	Construct a DFA for recognizing the unsigned real numbers with fractions, identifiers and unsigned integers.	**07**
			OR	

Q4	**(a)**	**(i)**	What is the difference between syntax tree and parse tree? Explain it with proper example	**04**
		(ii)	What is left factoring? Give example. Write unambiguous production rules for if then else construct.	**03**
	(b)	**(i)**	Construct DFA without constructing NFA for following regular expression: (x | y) * (y | z) * x*# .Write production rules from constructed DFA.	**07**
Q5	**(a)**	**(i)**	List the tasks performed by each pass of a two pass assembler. Also explain the following directives for an assembler: ASSUME, EQU, EXTERN, ORIGIN.	**07**

	(b)	**(i)**	Given the following program:
			START 500
			ID1 DS 4
			L1 MOVER AREG,D
			ADD AREG,C
			SUB AREG,C
			MOVEM AREG,ID1
			D EQU ID2
			L2 PRINT D
			ORIGIN ID1-2
			STOP
			ID2 DC
			END L1
			Show the contents of symbol table and intermediate code using variant 1 at the end of pass-I Assembling
				07

OR

Q5	**(a)**	**(i)**	Define token. Explain how it differs from pattern and lexeme. Find token, pattern and lexeme from following expressions. 1. if(y<=30) 2. total = sum + 16.5	**07**

	(b)	**(i)**	Given the following program:
			START 600
			ID1 DS 5
			L5 MOVER AREG,D
			ADD AREG,C
			SUB AREG,C
			MOVEM AREG,ID1
			D EQU ID2
			L2 PRINT D
			ORIGIN ID1-1
			STOP
			ID2 DC
			END L5
			Show the contents of symbol table and intermediate code using variant 1 at the end of pass-I Assembling
				07

****