

Seat No.: _____

Enrolment No. _____

GUJARAT TECHNOLOGICAL UNIVERSITY

BE - SEMESTER-V (NEW) EXAMINATION – SUMMER 2019

Subject Code: 2150708

Date: 03/06/2019

Subject Name: System Programming

Time: 02:30 PM TO 05:00 PM

Total Marks: 70

Instructions:

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

- Q.1**
- (a) Draw neat sketch diagram of Life cycle of a source program. **03**
 - (b) Define following: Language Migrator, Execution gap, Token, Handle **04**
 - (c) Explain Analysis and Synthesis phase of Compiler. **07**
Perform lexical, syntax and semantic analysis on below C statement:
int i;
float a, b;
a = b + i;
- Q.2**
- (a) Differentiate Application software with System software. **03**
 - (b) Write Regular expression for all string end with 'abb' and Construct equivalent DFA. **04**
 - (c) (i) Write unambiguous production rules to produce arithmetic expression consisting of +, *, (,), id. **07**
(ii) Remove left recursion from that unambiguous production rules and generate LL(1) parsing table for that grammar.
- OR**
- (c) (i) Define Operator precedence grammar. Convert following production rules of grammar into suitable Operator precedence grammar. **07**
 $E \rightarrow EAE \mid id$
 $A \rightarrow - \mid *$
(ii) Generate operator precedence relation matrix for converted Operator precedence grammar. Show how id - id * id will be parsed using Operator Precedence Matrix.
- Q.3**
- (a) Define Assembler. List out tasks performed during different phase of assembler. **03**
 - (b) Given the Grammar, evaluate the string id - id * id using shift reduce parser. **04**
 $E \rightarrow E - E$
 $E \rightarrow E * E$
 $E \rightarrow id$
 - (c) Compare Variant I and Variant II of intermediate code generation for assembler. **07**
Write intermediate code for Variant I and Variant II of below program fragment.
START 200
READ A
LOOP MOVER AREG, A
:
:
:
SUB AREG, '1'
BC GT, LOOP
STOP
A DS 1
LORG
...

OR

- Q.3** (a) Define forward references. How it can be solved using back-patching. **03**
(b) Describe following data structures: OPTAB, SYMTAB, LITAB and POOLTAB. **04**
(c) List out assembler directives and explain any two advance assembler directives. **07**

- Q.4** (a) Differentiate Linker and Loader. **03**
(b) Define Macro - preprocessor. Explain steps of Macro Preprocessor Design. **04**
(c) Explain use and field of following tables of a macro KPDTAB, MDT, EVTAB, SSTAB **07**

OR

- Q.4** (a) Explain Nested macro call with suitable example. **03**
(b) Explain attributes of formal parameter and expansion time variable in macro. **04**
(c) Write a brief note on MS-DOS Linker. **07**

- Q.5** (a) Explain the term self-relocating program. **03**
(b) Define overlay. Explain the execution of an overlay structured program. **04**
(c) List out various Code Optimization techniques used in Compiler. Explain any three technique with suitable example. **07**

OR

- Q.5** (a) Differentiate Compiler and Interpreter. **03**
(b) Compare Absolute Loader with Relocating Loader (BSS Loader). **04**
(c) Explain triple, quadruple and indirect triples representation with example. **07**
