

Total No. of Questions : 10]

SEAT No. :

P1358

[Total No. of Pages : 4

[4858] - 1106

T.E. (Information Technology)

SYSTEMS PROGRAMMING

(2012 Pattern) (Semester - II) (End-Sem.)

Time : 3 Hours]

[Max. Marks : 70

Instructions to the candidates:

- 1) Answer Q1 or Q2, Q3 or Q4, Q5 or Q6, Q7 or Q8, Q9 or Q10.
- 2) Neat diagrams must be drawn wherever necessary.
- 3) Figures to the right side indicate full marks.
- 4) Your answers will be valued as whole.
- 5) Assume suitable data if necessary.

UNIT - I, II & III

Q1) a) Explain how forward reference is handled in Single Pass assembler and solve it for the given example : [6]

```
START 202
MOVER AREG, = 5
MOVEM AREG , A
LOOP  MOVER AREG, A
      MOVER CREG, B
      ADD CREG, = '1'
      MOVEM CREG, B
      SUB CREG,A
      BC ANY, NEXT
      LTORG
      ADD CREG,B'
      BC LE LOOP
```

P.T.O.

```

NEXT SUB AREG, = '1'
      BC LT, BACK
      STOP
      ORIGIN 219
      MULT CREG, B
A     DS 1
BACK EQU LOOP
B     DS 1
      END

```

- b) Explain different parameter passing mechanisms in Macro-processor. [4]

OR

- Q2) a) Compare how the four basic tasks of loader are performed in BSS loading scheme and DLL scheme. [4]  
 b) Explain various advanced macro facilities like AIF, AGO, LCL, GBL, REPT and IRP statements with syntax and example. [6]

### UNIT - I, II & III

- Q3) For the code given below, show macro name table, macro definition table, Expanded code and the stack frame. [10]

```

MACRO
  XYZ &A, &B, &C
  READ &A
  ADD&B, ='5'
  PRINT & C
MEND
MACRO
  MIT &Z
  MACRO
    &z&W
    SUB &W, ='6'
    XYZ AREA, BREG, CR
    ADD &W, ='5'
  MEND

```

```

ADD CREG , = '5'
MEND
START 200
MIT HELLO
MULT BREG , = '4'
HELLO BREG
PRINT C
C DS 1
END

```

OR

- Q4)** a) List down various phases of compiler. What are functions of Lexical Analyzer? Explain patterns, tokens and lexemes with examples. [6]
- b) Give the flowchart for pass I of DLL scheme. [4]

#### UNIT - IV

- Q5)** a) Differentiate between top down parser and bottom up parser. [4]
- b) Generate SLR parsing table for the grammar given below and parse the string  $id1 + id2 + id3 * id4$  [10]
- Grammar:

```

E → E + T/T
T → T * F/F
F → id

```

- c) Explain operator precedence parser. [4]

OR

- Q6)** a) Design LR(1) parser for the given grammar. Also show the moves by the parser for input string "ab". [10]

```

S → aABb
A → c/ε
B → d/ε

```

- b) Compare SLR and LALR parsing methods. [4]
- c) Define Handle and handle pruning w.r.t. bottom up parser. For the grammar given, [4]

```
S → 0S1 / 01
```

Identify the handles at each step and parse the string 000111.

## UNIT - V

- Q7)** a) Explain the following : [8]
- i) Dependency graph of Type expression.
  - ii) Synthesized and inherited attributes.
- b) Write the context free grammar for following and also perform syntax directed translation for the same into three address code. [8]

while condition do S

OR

- Q8)** a) Explain the need of Intermediate code generation in compiler. Generate quadruple and triple for the given expression:  $a = b + c * d$ . [8]
- b) Write the method of generating intermediate code for the Boolean expression. [8]

## UNIT - VI

- Q9)** a) Obtain the TAC for the following code before and after applying the optimization techniques using. [12]

- i) Removal of Loop Invariants
- ii) Elimination of common sub expressions

for ( $i = 1$ ;  $i \leq 10$ ;  $i++$ )

$X[i][2*j-1] = Y[i][2*j-1]$

- b) Explain different intermediate code generation techniques. [4]

OR

- Q10)** a) Discuss code generation issues. [4]
- b) Discuss with suitable example machine dependent code optimization. [8]
- c) Write a short note on activation record. [4]

