

Total No. of Questions : 6]

SEAT No. :

P543

[Total No. of Pages : 3

APR - 18/TE/Insem. - 146
T.E. (Information Technology)
SYSTEMS PROGRAMMING
(2015 Course) (Semester - II)

Time : 1 Hour]

[Max. Marks : 30

Instructions to the candidates:

- 1) Answer Q1 or Q2, Q3 or Q4, Q5 or Q6.
- 2) Neat diagrams must be drawn wherever necessary.
- 3) Figures to the right indicate full marks.
- 4) Assume suitable data, if necessary.

- Q1)** a) What is language processing? Explain application, PL and execution domain? [4]
- b) For the following piece of assembly language code, show the contents of symbol table, literal table, pool table and IC. Assume machine opcodes and size of instruction as 1. [6]

START 100

A DC 10

MOVER AREG, B

MOVEM BREG, = '1'

ADD AREG, = '2'

SUB BREG, = '1'

B EQU A + 20

LTORG

STORE AREG, NUM

MOVER CREG, LOOP

ADD BREG, = '1'

NUM DS 5

LOOP DC 10

END

OR

P.T.O.

- Q2)** a) Compare 2-pass and single-pass assembler. [4]
b) List and explain the types of assembly language statements with examples. [6]

- Q3)** a) Explain basic functions of loader. Also explain the working of compile & go loader. [4]
b) With format explain the following macro-directives : MACRO, MEND, AIF, AGO, LBL, GBL. [6]

OR

- Q4)** a) What is subroutine linkage? How is it resolved? [4]
b) For the given assembly code generate MNT, MDT and expanded code [6]

MACRO

M1 &N, &A1 = , &R = AREG

MOVEM &R, &N

SUB &R, &A1

ADD &R, &N

MEND

MACRO

M2 &P, &Q = B, &OPR = DIV

MOVER AREG, &P

&OPR AREG, &Q

MOVEM BREG, &P

MEND

```

START 100

READ  VAR

M2    A, OPR = SUB

ADD   AREG, VAR

LDA   CREG, BREG

SUB   CREG, A

M1    C, R = BREG, A1 = A

A     DS 1
VAR   DC 2
C     DS 3
      END

```

- Q5)** a) With a neat diagram explain the phases of compiler. [5]
- b) Using the RE to DFA algorithm generate the DFA for the given regular expression. $(a + b)^* a^* b\#$. [5]

OR

- Q6)** a) Explain the structure of lex program. [4]
- b) Explain the processing of given statement w.r.t. first three phases of compiler. [6]

$$R = (b * b - 4 * a * c) / (2 * a).$$

