Total No. of Questions : 10]

SEAT No. :

P3322

[Total No. of Pages : 4

[5353]-197
T.E. (IT) (Semester - II)
SYSTEMS PROGRAMMING
(2012 Pattern)

Time : 2½ Hours] *[Max. Marks : 70*

Instructions to the candidates:

1) *Answer Q1 or Q2, Q3 or Q4, Q5 or Q6, Q7or Q8, Q9 or Q10.*
2) *Neat diagrams must be drawn wherever necessary.*
3) *Figures to the right side indicate full marks.*
4) *Assume Suitable data if necessary.*

Q1) a) Give the various data structures in the design of 2-pass macroprocessor. **[6]**

b) Define the following : **[4]**

i) Assembler

ii) Macroprocessor

iii) Compiler

iv) Loader/Linker

OR

Q2) a) With the structure explain the different tables that would be generated as output of lexical analysis. **[4]**

b) What are the assembler directives? Explain how assembler directives LTORG, ORIGIN, START, END and EQU are processed with examples. **[6]**

Q3) a) For the following piece of assembly language code, show the contents of symbol table, literal table and pool-tab. Assume size of instruction equal to one. **[6]**

START 500

MULT BREG, A

P.T.O.

```
        MOVEM AREG , = '10'
LOOP    MOVER AREG, A
        MOVER CREG,B
        ADD CREG, = '1'
        SUB CREG,A
        LTORG
        ADD CREG,B'
NEXT    SUB AREG,='1'
        STOP
        ORIGIN 300
        MULT CREG, B
A       DS 1
BACK    EQU LOOP
B       DS 1
        END
```

b) Define loader and enlist the basic functions of loader. **[4]**

OR

Q4) a) Explain the phases of compiler w.r.t the following statement : **[8]**

R = (b*b–4*a*c)/(2*a)

b) Define the term forward reference in an assembler. **[2]**

Q5) a) Consider the grammar **[8]**

$E \rightarrow E\text{-}E$

$E \rightarrow E*E$

$E \rightarrow id$

Perform shift Reduce parsing of i/p string "id – id * id"

b) Explain LEX file structure. **[6]**

c) Compare bottom up and top down parser. **[4]**

OR

Q6) a) Consider the following grammar

$S \rightarrow iEtSS'/a$

$S' \rightarrow eS/e$

$E \rightarrow b$

Design a table driven predictive parser and parse the string 'ibtae'. **[8]**

b) Explain YACC file structure. **[5]**

c) With a neat diagram explain the classification of parsers. **[5]**

Q7) a) Define synthesized and inherited attributes. For the grammar given, **[8]**

$T \rightarrow FT'$

$T' \rightarrow *FT'/\epsilon$

$F \rightarrow digit$

Draw the annotated parse tree for the expression 3*5*2 and list down the synthesized and inherited attributes.

b) Write the following expression in the form of postfix notation, Directed acyclic graph, quadruple and triple. **[8]**

a = b* (–c) + b*(–c)

OR

Q8) a) Draw the dependency graph for the example of 7 a. **[4]**

b) Translate the following C fragment into the three address code. **[6]**

```
int i,j,k;
int T[5][100];
j = 10; k = j–1;
for (i = 1; i <=5; i++)
{
    T[i] [j]=k*k;
    J=j+k
    T[i][j]=k*k–1;
}
```

c) What is the need for intermediate code generation?

Explain the different IC forms with examples. **[6]**

Q9) a) Optimize the following code **[8]**

```
i=1
j=1
t1=i * 80
t2=j*4
t3=t1+t2
t4=a[t3]
t5=j*80
t6=i*4
t7=t5+t6
t8=a[t7]
```

b) Write short note on activation record. **[4]**

c) Explain in brief run time storage allocation. **[4]**

OR

*Q10)*a) Generate three address code for **[8]**

```
while (a<c) and (b>d) do
{
If a =1 then c=c+1
Else while (a<=d) do
      A=a+3
}
```

b) Explain different storage allocation strategies. **[4]**

c) Explain various code generation issues. **[4]**