```
LD RO, i
MUL R0, R0,8
LD R1, a(R0)
ST b, R1
```

4

4/12/19 (Morning)

This question paper contains 6 printed pages.

Your Roll No. ....

38

Sl. No. of Ques. Paper	: 7877 J
Unique Paper Code	: 32347501
Name of Paper	: Systems Programming
Name of Course	: B.Sc. (Hons.) Computer Science : : DSE – 2
Semester	: V
Duration	: 3 hours
Maximum Marks	: 75

(Write your Roll No. on the top immediately on receipt of this question paper.)

The question paper consists of two Sections.

Section A is compulsory.

*Attempt any **four** questions from Section B.*

SECTION A

1. (a) Explain forward reference in assembler with the help of an example. 2

(b) Draw the activation tree for the function calls :

```
main ( ){pl( );}
pl( ){p2( ); p3( );}
p2( ){ }
p3( ){ }
```

2

(c) Define inherited attribute with the help of a suitable example. 2

P.T.O.

(d) Define type synthesis and type inference in the context of type checking. 2

(e) Write the structure of a Yacc Program. 2

(f) Translate the expression $a = (-c) * b + (-c)$ to quadruples. 2

(g) Draw the structures of Machine Opcode Table (MOT), Pseudo Opcode Table (POT) and Symbol Table (SYMTAB) used in the assembler. 3

(h) Find the FOLLOW( ) for every non-terminal in the following grammar : 3

$S \rightarrow Bb \mid Cd$

$B \rightarrow aB \mid \in$

$C \rightarrow cC \mid \in$

(i) What is the difference between shared library and static library? 3

(j) What are the functions performed by lexical analyser? 3

(k) How is the symbol table used by lexical and syntax analyzers of the compiler? 3

(l) Write token name, lexeme and attribute value for the following statement :

int * *a*; 4

(m) Generate the assembly code for the following sequence of statements. Assume *x*, *y* and *z* are in memory.

```
if x < y goto L1
z = 0
goto L2
```

(c) Compare and contrast single-pass and two-pass assemblers. 3

7. (a) What is the output of the following code?

```
int f (int x, int *y, int ** z)
{
    **z+= 1;
    *y+= 2;
    x+= 3;
    return x +*y +**z;
    }
int main( )
{
    int x, c, *b, **a;
    c = 4, b = &c, a = &b;
    x = f(c, b, a);
    printf("%d", x);
    return 0;
}
```

4

(b) "Values communicated between caller and callee are placed in the beginning of callee's activation record." Why? 2

(c) Determine the cost of each instruction in the following assembly code. Assume that each memory access (including the machine instruction) has cost 1.

$L1 : z = 1$

$L2 : z = 2$ 4

SECTION B

2. (a) Consider the following section tables for two object files x.obj and y.obj. After Pass I of linking process, show the content of appropriate data structures formed and also show the layout of final executable module. 7

Name	Size	Align
.text	305	16
.data	59	4
.data1	65	4

Section table for x.obj

Name	Size	Align
.text	200	16
.data	175	4
.data2	300	4

Section table for y.obj

(b) How does Yacc resolve ambiguity in the following expression grammar? 3

$$E \rightarrow E + E \mid E * E \mid id$$

3. (a) What is Position Independent Code? 2

(b) Consider the augmented expression grammar

$$E' \rightarrow E$$
$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow id$$ 8

Construct the LR (0) automaton and parse the input string *id* ∗*id* using shift/reduce parser.

4. (a) Write a type expression for an "array of 4 arrays of 3 integers each". 2

(b) What are the benefits of using quadruples over the triples in three address code generator? 2

(c) Consider the following augmented grammar :

$$S' \rightarrow S$$

Construct the SLR (1) parsing table. 6

5. (a) Consider the grammar :

$S \rightarrow BC \mid DA$

$C \rightarrow AA$

$B \rightarrow b$

$A \rightarrow a$

$D \rightarrow ba$

Construct LALR parsing table for the above grammar. 7

(b) What are the limitations of Load-and-Go Assembler? 3

6. (a) Consider the Syntax Directed Definitions :

$T \rightarrow T_1 * F\{T.val = T_1 . val \times F . val\}$

$E \rightarrow T \{E . val = T . val\}$

$E \rightarrow E_1 + T \{E . val = E_1 . val + T . val\}$

$T \rightarrow F \{T . val = F . val\}$

$F \rightarrow G \uparrow F \{F . val = POWER(F . val, G . val)\}$

$F \rightarrow G \{F . val = G. val\}$

$G \rightarrow digit \{G . val = digit.lexval\}$

Construct annotated parse tree for $1 * 3 \uparrow 2 + 5 * 3$ and give the output. 5

(b) Define handle pruning with the help of a suitable example. 2