Unique Paper Code	: 32347501
Name of Course	: B.Sc. Hons. Computer Science
Name of the Paper	: Systems Programming
Semester	: V
Duration of Examination	: 3 Hours
Maximum Marks	: 75 Marks
Students admitted in the year	: 2015-2018

Instructions for Candidates:

1. Answer any **FOUR** questions.
2. All questions carry equal marks.

1. Show the result after each phase of compilation (Lexical Analysis, Syntax Analysis, Semantic Analysis, Intermediate Code Generation, and Code Generation) for the following assignment statement. Also, show the symbol table generated during the process.

   a = b – c + d / e

   Clearly state the assumptions, if any.

2. Consider the following Syntax Directed Translation (SDT):

   $N \rightarrow L \quad \{N.dval = L.dval\}$
   $L \rightarrow L_1\ B \quad \{L.dval = L_1.dval * 2 + B.dval\}$
   $L \rightarrow B \quad \{L.dval = B.dval\}$
   $B \rightarrow 0 \quad \{B.dval = 0\}$
   $B \rightarrow 1 \quad \{B.dval = 1\}$

   Draw the annotated parse tree for a bit string "`011000101`" showing all dependencies and evaluate the final value of `N.dval`. Identify the synthesized and inherited attributes (if any). Modify the above SDT to determine the number of bits in the bit string.

3. Consider a two-address hypothetical machine with two general purpose registers, which supports load, store and arithmetic operations (assume the mnemonics of these instructions). The arithmetic operations require that at least one of the operands is in the register. Generate the machine code for the following three address instructions. Also, compute the cost of each instruction, stating any assumptions you make.

$a = n + 1$
$b = 2 * n$
$c = b + 1$
$d = n * a$
$x = d * c$
$s = x/6$

4. Consider a hypothetical machine with a 16-bit accumulator register (AX) and two general purpose 16-bit registers, P and R. The machine also supports two pseudo-instructions, `db` to reserve and initialize 8-bit memory location and `dw` to reserve and initialize 16-bit memory locations. Instruction set of the machine is given below:

Assembly Instruction	**Size of instruction (in bytes)**	**Meaning**
MOV R,AX	3	R ← AX
MOV P,AX	3	P ← AX
MOV AX,R	3	AX ← R
MOV AX,P	3	AX ← P
ADD AX,R	1	AX← AX + R
ADD AX,P	1	AX←AX + P
SUB AX,R	1	AX← AX – R
SUB AX,P	1	AX← AX – P
MOV AX,M	3	AX←M (memory location to AX)
MOV P,M	4	P←M (memory location to P)
MOV R,M	4	R←M (memory location to R)
CLEAR	5	To initialize the accumulator with 0
APPEND M,00h	6	Append a hexadecimal 00 in front of content of a memory location (M)
JMP <label>	2	Jump to some label
RET	1	To stop the execution of program

Consider the two following assembly language programs, namely P1 and P2. Construct and populate a symbol table and a section table for each of these programs.

P1: To add two 16-bit numbers

```
section . SegA
        op1 dw 0200h
        op2 dw 0400h

section . SegB
        CLEAR
        MOV AX, op1
        MOV P, op2
        ADD AX, P
```

```
        JMP UVW
UVW: MOV R, AX
        RET
```

P2: To subtract two 8-bit numbers

```
section . SegA
        ab1 db A0h
        ab2 db 10h

section . SegC
        APPEND ab1,00h
        APPEND ab2,00h
        CLEAR
        MOV AX, ab1
        MOV R, ab2
        SUB AX, R
        JMP STU
STU:    MOV P, AX
        RET
```

5. Consider the following grammar with set of non-terminals as `{S, V, E}` and set of terminals as `{x, =, n}`. `S` is the start symbol.

$S \rightarrow x$
$S \rightarrow V = E$
$V \rightarrow x$
$E \rightarrow V \mid n$

Construct LR (1) set of items and `GOTO` graph (DFA). Also, construct LR (1) parse table and parse the string "`x=n`".

6. Consider the following grammar with set of non-terminals as `{S, V}` and set of terminals as `{(, ',', c, )}`. `S` is the start symbol.

$S \rightarrow (V) \mid c$
$V \rightarrow V, S \mid S$

Construct the parse tree for sentences $(c, c), \big(c, (c, c)\big), and\ (c, (c, c), (c, c))$. Also, construct rightmost derivation of each sentence and determine the handle in each step.