Course Code : CST 402 **JSRK/MW – 17 / 2099**

Seventh Semester B. E. (Computer Science and Engineering) Examination

LANGUAGE PROCESSORS

Time : 3 Hours] [Max. Marks : 60

Instructions to Candidates :—

(1) All questions carry equal marks.
(2) Assume suitable data wherever necessary.
(3) Due credit will be given to neatness and adequate answers.

1. (a) Given the following tokens and their associated regular expressions, show what output is produced when scanner is run over the following strings :—

a * b [printf (“1”) ;]
(a | b) * b [printf (“2”) ;]
c * [printf (“3”) ;]

(i) aaabccabbb (ii) cbbbbac (iii) cbabc. 3(CO 1)

(b) Construct parse tree, syntax tree and DAG for (a+b)*(b+c)+a by considering the grammar $E \rightarrow E + E \mid E * E \mid (E) \mid id$. 3(CO 1)

(c) Differentiate between re.search and re.match by considering suitable example. 2(CO 1)

(d) List out phases under Front End and Back End of Compiler. 2(CO 1)

2. (a) Determine whether given grammar is LL (1) or not by constructing Parsing Table.

$S \rightarrow a\ I\ J\ h$
$I \rightarrow I\ b\ Se \mid c$
$J \rightarrow K\ L\ K\ r \mid \in$
$K \rightarrow d \mid \in$
$L \rightarrow P \mid \in$ 5(CO 2)

(b) Comment on "Design of SLR parser for Ambiguous grammar'. Construct SLR parser for the given ambiguous grammar and resolve the ambiguity.

$S \rightarrow iSeS \mid iS \mid a$ 5(CO 2)

OR

3. Determine whether given grammar is LALR or not,

$A \rightarrow aCDq \mid aBg \mid \in$

$D \rightarrow d \mid \in$

$B \rightarrow e \mid \in$

$C \rightarrow Ct \mid P \mid \in \mid BD \mid rAb$

10(CO 2)

4. (a) Generate Three Address Code (TAC) for B [i , j , k] : = C [i + j , k , I + z]. Compute 1 – value of C [4 , 3 , 3] considering base address of array C as 100 and bpw = 4. Consider size of array B as $6 \times 6 \times 4$ and size of array C as $5 \times 6 \times 4$. Also validate your result by giving memory representation of array. 7(CO 3)

(b) Generate 3 Address Code for the given program fragment.

```
if ( ( A > B and not B > C ) or ( C > D ) )
{
        while ( A > Z ) do
        D = D + 1
}
        else
{
        D = D – 1
}
```

3(CO 3)

OR

Draw the annotated parse tree for the expression (3+4)*(5*6) by considering the syntax directed definitions given below. Also identify type of attributed definition used in given SDD.

$L \rightarrow E$	L.val = E.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
$T \rightarrow T_1 * F$	$T.val = T_1.val * F.val$
$F \rightarrow (E)$	F.val = E.val
F → digit	**F.val = digit.lexval**

3(CO 3)

5. (a) Explain format of an Activation Record by considering following C program :—

```
main ( )
{
        int a , b , c
        a = 10 ;
        b = 20 ;
        c = sum ( a , b ) ;
}
        int sum ( x , y )
        int x , y ;
{
        int z ;
        z = x , y ;
        return z ;
}
```

5(CO 4)

(b) Implement phrase level error recovery routines for LR parsing by considering given grammar.

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

Also validate the string w = id + id). 5(CO 2)

6. Solve any **Two** :—

(a) Consider given program fragment,

```
FOR I : 1 TO n – 1 DO
        FOR J := 1 TO I DO
            IF A [ J ] > A [ J + 1 ] THEN
                BEGIN
                    Temp := A [ J ]
```

```
        A [ J ] := A [ J + 1 ]
        A [ J + 1 ] := Temp
    END
```

(i) Obtain 3 – Address Code

(ii) Obtain Basic Blocks

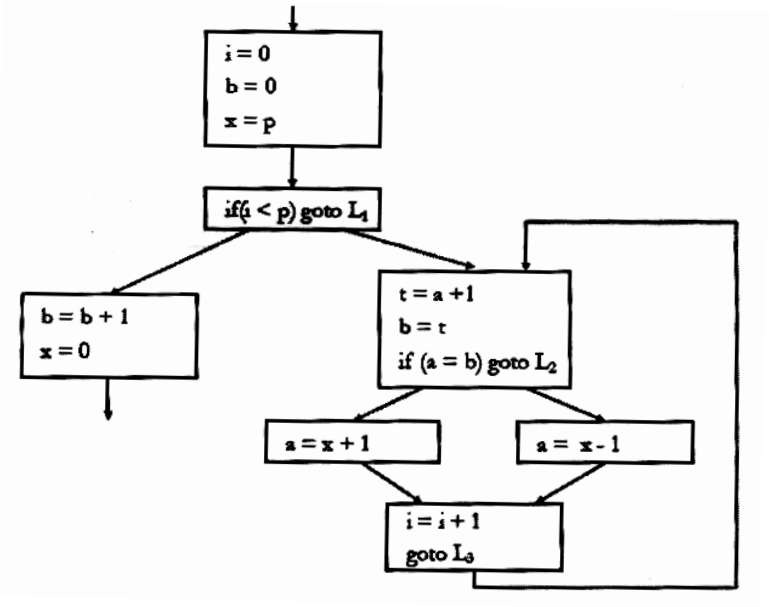
(iii) Obtain the Flow Graph. 5(CO 3 , 4)

(b) Identify and explain code optimizing transformations required to optimize given code and apply the same for optimization,

```
main ( )
{
int n = 10 , c = 2 , x , y , s ;
for (i = 0 ; i < n ; i + +)
{
x = 1.1 * 3 ;
y = s ;
s = y + i * c ;
x = s + 22 / 7 ;
}
}
```

5(CO 4)

(c) Perform **Live Variable Analysis** for the given flow graph.



5(CO 4)

7. (a) List out various issues to be considered for code generation phase. Explain each in brief. 5(CO 4)

(b) Generate Target Code for the given program fragment by considering **opitmal** code generation algorithm.
expr : ((a*b) / (c+d)) – (e+f) 5(CO 4)