Course Code : CST 402/412 **CXDW/RW – 18 / 5103**

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

LANGUAGE PROCESSORS

Time : 3 Hours] [Max. Marks : 60

Instructions to Candidates :—

(1) Each Question carry marks as indicated.
(2) Due credit will be given to neatness
(3) Assume suitable data wherever necessary.
(4) Illustrate your answer, wherever necessary, with the help of neat sketches.

1. (a) Explain the working of input buffering using the buffer pair method for the input statement for (i=1; i<=20; i++). Assume that the buffer 1 and buffer 2 can hold 10 characters. 3 (CO 1)

   (b) Design the output of the lexical analysis phase for the given code fragment
   int i;
   printf ("String is %d", ++ i ++ &&&i ** a);
   Find the total number of tokens and distinct tokens present in the code fragment. 2 (CO 1)

   (c) Unsigned numbers are strings such as 5280.39.37, 6.336E4, 1.894E-4. Give the regular expression for above mentioned string. Also draw transition diagram. 3 (CO 1)

   (d) Write a LEX code to display the histograms of the length of words. 2 (CO 1)

2. (a) Can any ambiguous grammar be SLR ? Consider the ambiguous grammar given and find the type of conflicts that arise in SLR parser.

   S –> A | B | AB
   A –> aA | c
   B –> bB | c 5 (CO 2)

(b) Construct the operator precedence table and precedence function table for parsing a string consisting of id, –, *, + and $ by Considering the associatively and precedence of C language. Show the parsing of the string $ id-id*id*id+id$. Specify the statement used to handle associativity and precedence in YACC for the same. 5 (CO 2)

3. (a) Generate the 3 address code using SDTS

```
Do
  If  A == 0  then
        A = B + C * D
  else
  repeat
  A = A + 1
  until A<5
  i = i + 1
while (i<10)
```

5 (CO 3)

(b) Justify the need of the intermediate code. Represent the expression Z = (-A * B) + (C * D) + (E – F) / (C * D) in various intermediate code representations. Also convert the given expression to postfix form using SDTS. 5 (CO 3)

OR

(c) Generate TAC using SDTS for A[I, J + 1] = B[I, C[I, J]] + D [I, J + 1] where w = 4 and the size of arrays A, B, C and D are 10 x 20, 10 x 5, 5 x 5 and 10 x 5 respectively. 5 (CO 3)

4. (a) Construct the LL(1) parsing table for the given grammar and show the modified parsing table with error handling routines using phrase level error recovery.

A –> A + B | B

B –> B * C | C

C –> C – D | D

D –> a | b 8 (CO 1, 2)

(b) Build the symbol table for the code fragment given below using nesting depth and static distance.

```
Main()
{
  int  x,  y;
  int  A(){
              Char   R(){ int  c;  char   a,  b;}
          }
  int  B(){
            real   s;
            int   p,  q;
            int   C()
            {
                int  s,  t;
                      int   D()   {int   r:}
            }
            }
 }
```

2 (CO 1)

5. (a) Perform live variable analysis using IN, OUT method on the following code.

```
int  func  (int a,  int b)
int  i,  j,  k;
i = 45;
j = a + b;
if (a + i) > 100 {
       k = a + j
} else {
     k = b + j
}
return  (k)
```

8 (CO 4)

OR

(b) Find the reaching definition for the following

```
i = m–1;   J = n;   a = U1
do
    i = i + 1;
    j = j –1;
```

```
if  e1  then
     a  =  U2;
else
   i  =  U3
while  e2
```
8 (CO 4)

(c) Enlist and explain the machine dependent code optimization techniques. 2 (CO 4)

6. (a) Generate the code for the given expression using gencode() algorithm $a = b*(c*d) + e/(f-g) + e$. Obtain the optimal order of execution of statement of expression using heuristic algorithm. 6 (CO 4)

(b) Generate the code for given array assignment statements using simple code generation algorithm.

```
X  =  a[i]
Y  =  b[j]
Z  =  X*Y
```
4 (CO 4)