

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST-3 EXAMINATION- JUNE -2016

B.Tech VI Semester

COURSE CODE: 10B11CI612

MAX. MARKS: 35

COURSE NAME: Compiler Design

COURSE CREDITS: 04

MAX. TIME: 2 HRS

Note: All questions are compulsory. Carrying of mobile phone during examinations will be treated as case of unfair means.

1. Answer the following questions briefly [7x2 = 14 Marks]
 - a. Differentiate between an interpreter, a compiler and an assembler.
 - b. Differentiate between lexeme and token. State the phase(s) of compiler that each of these appears in.
 - c. What is a cross compiler? Justify why preprocessors are considered a type of compiler.
 - d. What is an SLR grammar? Arrange in ascending order the parsers SLR, CLR and LALR according to the parser strength. Write your reasons for choosing your order.
 - e. Explain the significance of the 'closure' function and the 'goto' function in LR parsers.
 - f. Create Abstract Syntax Tree(AST) and Directed Acyclic Graph(DAG) for the expression $c = a + b + c * (a + b)$, considering left associativity and standard operator precedence.
 - g. Generate 'Triples' intermediate representation for: $a = b * c + e[f] - (-h + i(j))$
2. Write a L-attributed Syntax Directed Translation (SDT) to convert the dates of format type DD/MM/YYYY to format type MM/DD/YYYY. [3 Marks]
3. Generate machine target code for the following: [3 Marks]
 - (i) $x = a[i] + 1$
 - (ii) $x = (a * b) + (c - (d + e))$
 - (iii) $*p++ = *q++$
4. Write the semantic rules for type checking and type definition for the given grammar. Consider implicit type casting from integer to real for '+' and '=' operators. [5 Marks]
 - $P \rightarrow D S$
 - $D \rightarrow T V$
 - $T \rightarrow \text{int} \mid \text{real}$
 - $V \rightarrow \text{id} \mid \text{id}[\text{NUM}] \mid * \text{id}$
 - $S \rightarrow S S \mid \text{do } S \text{ while } E \mid \text{id} = E$
 - $E \rightarrow E + E \mid -E \mid E(E) \mid \text{NUM} \mid \text{NUM} . \text{NUM}$

5. Generate a three-address intermediate representation for the code given below:[5 Marks]

```

procedure (a, b, x, y, z)
begin
  c := address-of (x) //where c is an integer pointer
  for i = 1 to 10 do
    b[i] = 0
  i := 1
  while x<y and y>z do
    x := x+1
    if x<y then
      while z<x do
        a[z][i] := b[i]
        z := z+1
      end while
    else
      *c := *c+x
      func (x, y, z)
    end if
  end while
end

```

6. a. Identify basic blocks and build flow graph for the intermediate code given below.
 b. Explain the 'copy propagation' and 'code motion' optimization techniques with suitable examples. Is any of these techniques applicable on the given three-address code? Justify.

[2.5+2.5 = 5 Marks]

(0) max = 0	(10) t3 = 2 * i	(20) t8 = 2 * i
(1) min = 100	(11) t4 = a[t3]	(21) t9 = a[t8]
(2) i = 0	(12) if t4 > max goto (16)	(22) if t9 < min goto (26)
(3) if i < n goto (9)	(13) t5 = i + 1	(23) t10 = i + 1
(4) i = 0	(14) i = t5	(24) i = t10
(5) if i < n goto (19)	(15) goto (3)	(25) goto (5)
(6) t1 = min + max	(16) t6 = 2 * i	(26) t11 = 2 * i
(7) t2 = t1 / 2	(17) t7 = a[t6]	(27) t12 = a[t11]
(8) a = t2	(18) max = t7	(28) min = t12
(9) goto end	(19) goto (13)	(29) goto (23)