SRN										

PES UNIVERSITY, Bangalore

UE18/19CS351

(Established under Karnataka Act No. 16 of 2013)

END SEMESTER ASSESSMENT (ESA) - B.TECH VI SEMESTER – May, 2022

UE18/19CS351 - COMPILER DESIGN

Time: 3 Hrs **Answer All Questions** **Max Marks: 100**

1	a	With a neat diagram, explain the interaction between first two phases of the compiler using the following input: x = a + b * c; (Note : clearly mention ALL input(s) and output of each phase and their role)	10

	b	Given the lex script, answer the following :
		%%
		a?aab printf("2");
		a?b printf("1");
		aab printf("3");
		I. What will be output of a lexer for the input string : **aab**
		II. Specify an input string for which the lexer will print **321** as the output.
		III. Which of the aforementioned rules are useless and why?
			5
			(2+1+2)

	c	With a neat diagram explain the design of a lexical analyzer.	5
2	a	Given the grammar: A → B C | a B → b C | λ C → ab Answer the following : I. What is first(A)? II. What is follow(A) and follow(C)? III. Construct the LL(1) table and specify whether the grammar is in LL(1) or not?	10 (2+3+5)
	b	Given the LALR parsing table, parse the string: ***id=id** Grammar: 1. $S' \rightarrow S$ 2. $S \rightarrow L = R$ 3. $S \rightarrow R$ 4. $L \rightarrow * R$ 5. $L \rightarrow$ **id** 6. $R \rightarrow L$	5

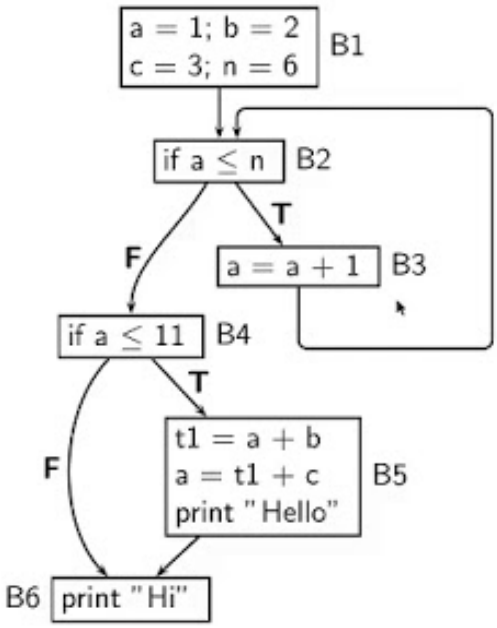
	id	*	=	$	*S*	*L*	*R*
0	s5	s4			1	2	3
1				acc			
2			s6	r6			
3				r3			
4	s5	s4				9	7
5			r5	r5			
6	s5	s4				9	8
7			r4	r4			
8				r2			
9			r6	r6			

SRN										

	c	When does s/r and r/r conflict occur in SLR parser? Explain with an example or a dummy state diagram to depict the scenarios.	5
3	a	Write short notes on: I. Types of attributes II. Types of Syntax Directed Definitions III. Procedure to convert SDD to SDT scheme	10 (4+4+2)
	b	Write an SDD for simple type declaration consisting of basic type T and list of identifiers L. D → T L T → int | float L → L, id | id	5
	c	Carry out the given SDD over the input string **5 * 2 * 3** and provide an annotated parse tree as output. PRODUCTION — SEMANTIC RULES 1) $T \rightarrow F\ T'$ — $T'.inh = F.val$; $T.val = T'.syn$ 2) $T' \rightarrow *\ F\ T_1'$ — $T_1'.inh = T'.inh \times F.val$; $T'.syn = T_1'.syn$ 3) $T' \rightarrow \epsilon$ — $T'.syn = T'.inh$ 4) $F \rightarrow \mathbf{digit}$ — $F.val = \mathbf{digit}.lexval$	5
4	a	Answer the following : I. What is a basic block? [1 mark] A. What is a leader? [1 mark] B. How are leaders identified? [3 marks] II. What is Code Optimization? [1 mark] Briefly explain the following optimizations with an example: A. Constant folding and propagation [2 marks] B. Loop optimization (Any one) [2 marks]	10 (5+5)

	b	Convert the given program to SSA form:
		x = a[i];
		if x > n
		x = x - n;
		else
		x = x + n;
		y = x * 5;
			5

SRN										

	c	Given the flow graph, perform live variable analysis.  	5

5	a	Consider the C code to compute Fibonacci numbers recursively. The questions below assume that the initial call is f(5).
		int f(int n) {
		int t, s;
		if (n < 2) return 1;
		s = f(n-1);
		t = f(n-2);
		return s+t;
		}
		I. Show the complete activation tree.
		II. How does the stack and its activation records look like the first time f(1) is about to return?
			10
			(5+5)

	b	Using Simple Code Generator algorithm, generate target code sequence for the given basic block: 1. x = y+z 2. z = x*x 3. y = z 4. x = y+z Assume number of registers available are 2 i.e. R1 and R2. All the variables (x,y,z) are live on exit from the block.	5
	c	Provide the Activation Record structure.	5