SRM

SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing
SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu
Academic Year: 2022 (EVEN)
B.Tech-School of Computing

Test: CLA-T3 **Date: 27.04.2022**
Course Code & Title: 18CSC304J & COMPILER DESIGN **Duration: 2 periods**
Year & Sem: III Year /VI Sem **Max. Marks: 50**

Batch & Set: II & D

S.No.	Course Outcome	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PO1 3	PO1 4	PO1 5
1	CO1	H	H	H	H	M	L	L	L	M	M	L	H	H	H	H
2	CO2	H	H	H	H	M	L	L	L	M	M	L	H	H	H	H
3	CO3	H	H	H	H	M	L	L	L	M	M	L	H	H	H	H
4	CO4	H	H	H	H	M	L	L	L	M	M	L	H	H	H	H
5	CO5	H	H	H	H	M	L	L	L	M	M	L	H	H	H	H
6	CO6	H	H	H	H	M	L	L	L	M	M	L	H	H	H	H

Part – A

Answer all (20 x 1=20 marks)

Q. No	Question	Marks	BL	CO	PO	PI Code
1	The sequence of procedure calls of a program corresponds to which traversal of the activation tree? A. In order traversal B. Pre order traversal C. Post order traversal D. Level order traversal	1	1	4	1	1.6.1

2	Consider the code following to apply the dead code elimination,
	If (condition)
	{ a = y OP z;
	}
	else
	{ ... }
	c = y OP z;
	y OP z should be computed as how many times in optimized code?
	A. one B. Two C. Three D. Four
		1	1	4	2	2.6.2

3	Which algorithm invokes a function GETREG()? A Code motion algorithm B. Code optimization algorithm C Intermediate Code algorithm D. Code generation	1	2	4	2	2.6.2
4	Consider the intermediate code given below. (1) i = 1 (2) j = 1 (3) t1 = 5 * i (4) t2 = t1 + j	1	2	4	2	2.7.1

	(5) t3 = 4 * t2 (6) t4 = t3 (7) a[t4] = – 1 (8) j = j + 1 (9) if j <= 5 goto (3) (10) i = i + 1 (11) if i < 5 goto (2) The number of nodes and edges in the control-flow-graph constructed for the above code, respectively, are a) 5 and 7 b) 5 and 5 c) 6 and 7 d) 7 and 8					
5	Which of the following is not a three-address code? a). a = 5 b) b=a c) c=a+b d) d=a+b-c	1	2	4	1	1.6.1
6	The postfix representation of the following expression: (x + y) * (x – y) a) + x y * - x y b) x y + * x y - c)x y + x y - * d)x y + x y * -	1	2	5	1	1.7.1
7	How many leaders are there in the following code? Goto L4 L1: a=a+b Goto L1 Goto L2 L2: a=a-b Goto L1 Goto L2 L4: c=a+b+8 a) 8 b) 5 c) 4 d) 3	1	1	5	1	1.5.1
8	Which optimization techniques is used to reduce multiple jumps? A. Latter optimization technique B. Peephole optimization technique C. Local optimization technique D. Code optimization technique	1	1	5	2	2.7.1
9	Generation of Intermediate code based on a abstract machine model is useful in Compilers because, A. Makes implementation of LA & Syntax Analysis easier B. SDT can be written for ICG C. Enhances the portability of the front end of the Compiler D. Not possible to generate code for real machines directly from high level language programs	1	1	5	2	2.8.[illegible]
10	Consider the following code. The following variables at line number (3) does not have next use _________ (1) x= y+z (2) z= x* 6 (3) t= z+1 (4) y=z-t (5) x=z+y a) all the variables have next use b) t c) z d) y	1	1	5	1	1.6.1

11	The following code is an example of?
	voidadd_ten(int x)
	{
	return x + 10;
	printf(""value of x is %d"", x);
	}
		1	2	6	1	1.7.1

	(A) Redundant instruction elimination B. Unreachable code (C) Flow of control optimization D. Reachable code					
12	Consider the following three address code. Identify the CORRECT collection of different optimization can be performed? m = 3 j = n v = 2 * n limit = integer n / 2 L1: j = j – 1 t4 = 4 * j t5 = a[t4] if t5 > limit – v goto L1 A. Code Motion, Constant Folding, Induction Variable Elimination, Reduction in Strength B. Copy Propagation , Code Motion, Deadcode Elimination, Reduction in Strength C. Constant Folding, Copy Propagation, Deadcode Elimination, Reduction in Strength D. Code Motion, Constant Folding, Copy Propagation, Induction Variable Elimination	1	2	6	1	**1.6.1**
13	In algebraic expression simplification, a=a+1 can simply be replaced by? A. a B. INC a C. DEC a D. MUL a	1	2	6	2	**2.7.2**
14	SDT scheme is desirable because, A. It is based on the syntax B. Its description is independent of any implementation C. It is easy to modify D. It is unable to modify	1	2	6	1	**1.6.1**
15	Which graph describes the basic block and successor relationship? a) Control Graph b) DAG c) Flow graph d) Hamiltonian graph	1	1	6	2	**2.6.2**
16	Identifying the class of statement when compiled does not produce any executable code A. Structural statement B. I/O statement C. Assignment statement D. Declaration	1	2	6	2	**2.6.3**
17	The definitions that are alive in B2 of the given flow graph are: B1 d1 i := m-1 d2 j := n ↓ B2 d3 j := j-1 a) d2 and d1 b) d1 and d3 b) d1, d2, d3 d) all are alive	1	1	6	3	**3.5.2**
18	_____ technique used to evaluate syntax directed definitions with both synthesized and inherited attributes. A. Eval Graphs B. SD graphs C. SDD Graphs D. Dependency graph	1	1	6	4	**4.4.2**
19	_____ is a Syntax Directed Definition that uses only synthesized attributes.	1	1	6	1	**1.6.1**

	A. A-Syntax Definition Definition B. K. Directed C. N-SD Definition definition D. S-Attributed					
20	Cross-compiler is a compiler __________ A. Which is written in a different language from the source language? B. That generates object code for the machine it's running on. C. Which is written in the same language as the source language? D. That runs on one machine but produces object code for another machine	1	1	6	1	1.6.1

Part – B

Answer any three (3 x 10=30 marks)

Q. No	Question	Marks	BL	CO	PO	PI Code
21	Generate the three-address code statement and construct the DAG representation for the expression I= a+ a*(b-c) +(b-c) *d.	10	6	4	6	6.3.1

22	Apply the peephole optimization for the given code.
	`1. sum = 0`
	`2. i = 1`
	`3. if i > n goto 15`
	`4. t1 = addr(a) - 4`
	`5. t2 = i * 4`
	`6. t3 = t1[t2]`
	`7. t4 = addr(a) - 4`
	`8. t5 = i * 4`
	`9. t6 = t4[t5]`
	`10. t7 = t3 * t6`
	`11. t8 = sum + t7`
	`12. sum = t8`
	`13. i = i + 1`
	`14. goto 3`
		10	5	5	4	4.1.3

23	Consider the following code, find out the minimum number of registers required to compile the given code a) with optimizations and b) without optimization
	`c = a + b;`
	`d = c * a;`
	`e = c + a;`
	`x = c * c;`
	`if (x > a) {`
	`    y = a * a;`
	`}`
	`else {`
	`    d = d * a;`
	`    e = e * e;`
	`}`
		10	3	6	1	2.1.3

24	Discuss in detail about storage allocation strategies.	10	2	6	1	1.3.1