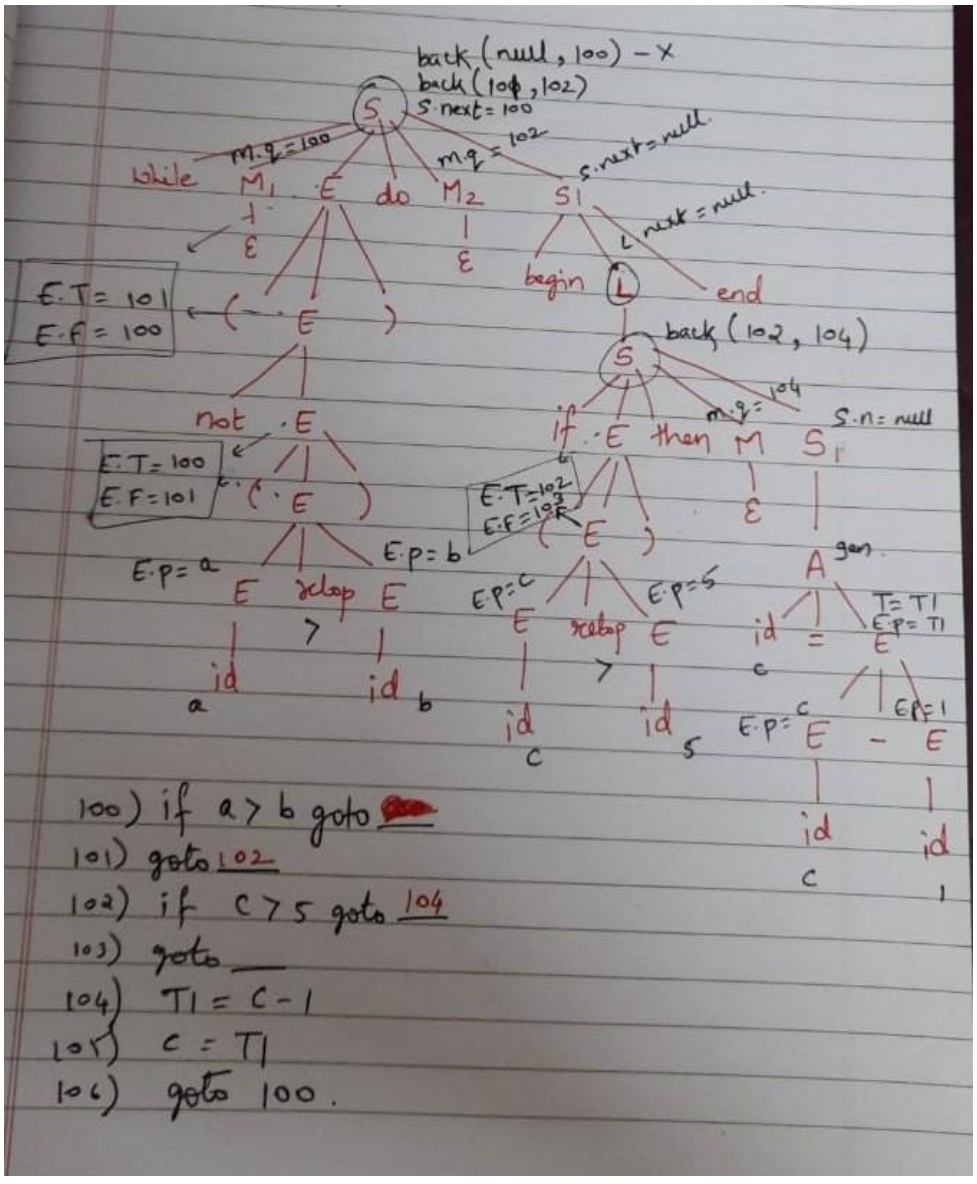
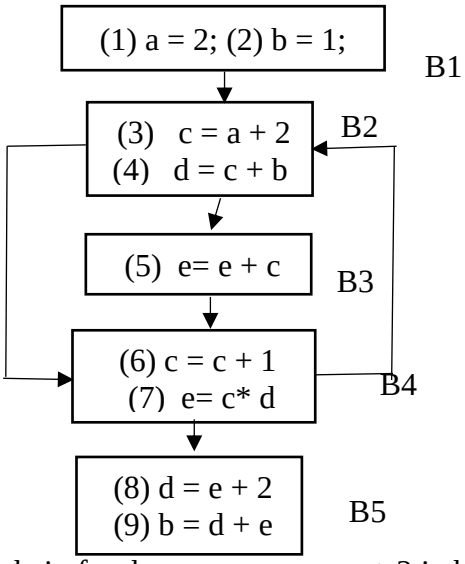
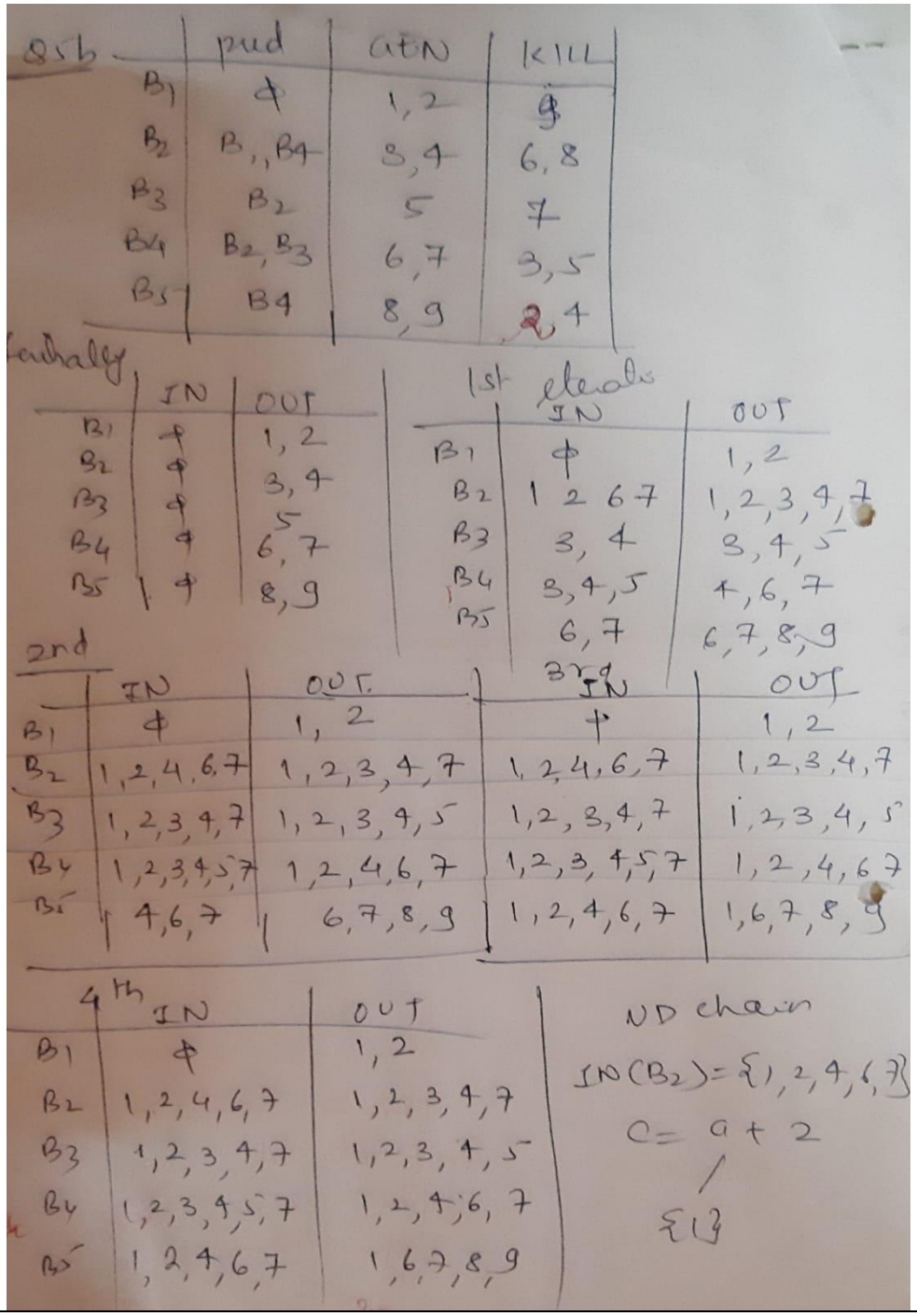
ACAD-27 a)	Shri Ramdeobaba College of Engineering and Management, Nagpur -440013	Iss. No.: 01, Rev. No.: 00
Ref. Clause(s): 9.1		Date of Rev: 01/01/2018
Department: Computer Science and Engineering	Semester: VI Semester Shift: I Course Code: CST 358 Course Name: Compiler Design	Page: 01/02
Programme: B.E.(CSE)	Test 2 Model Answer	Date of Exam: 16/04/2022
Max Marks: 15	Session: 2021-22	Time: 1 Hour [4:00 PM - 5:00 PM]

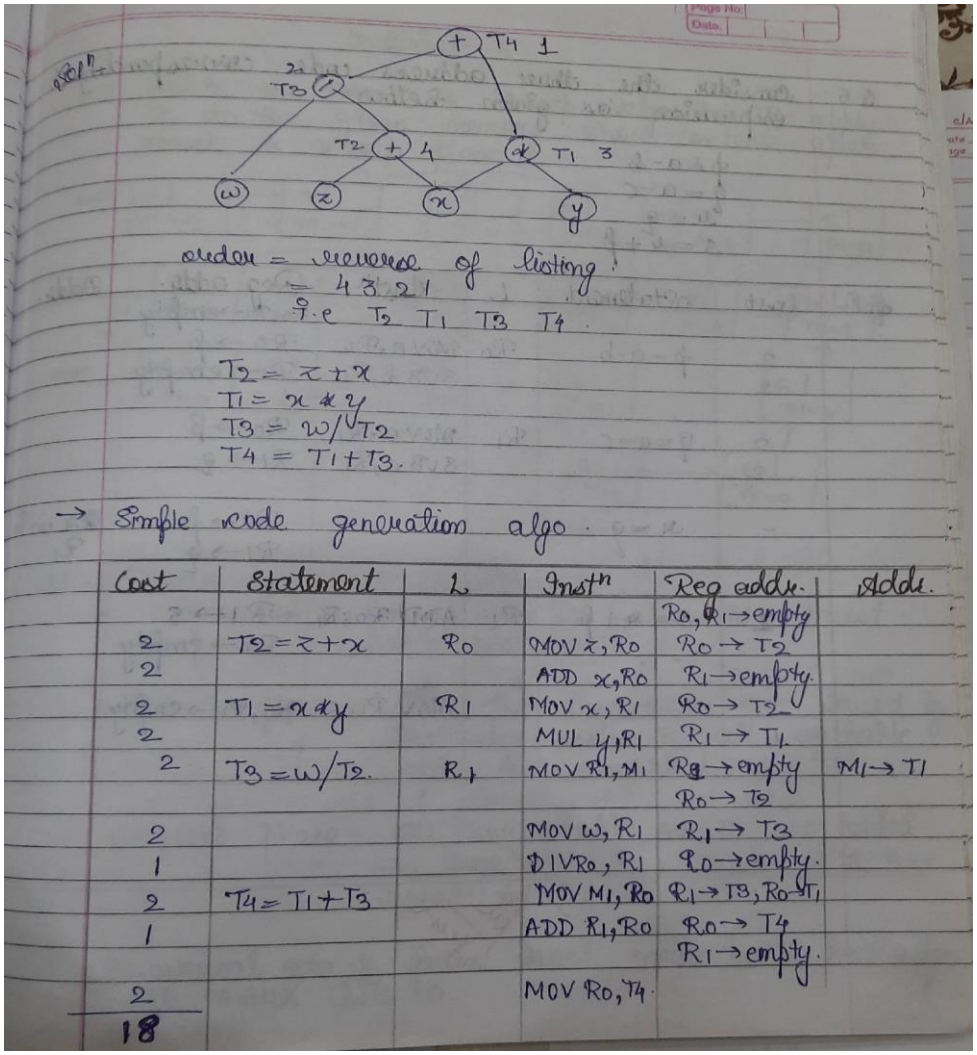
Instructions: Give proper assumptions wherever necessary.

Que		Marks	CO	EO

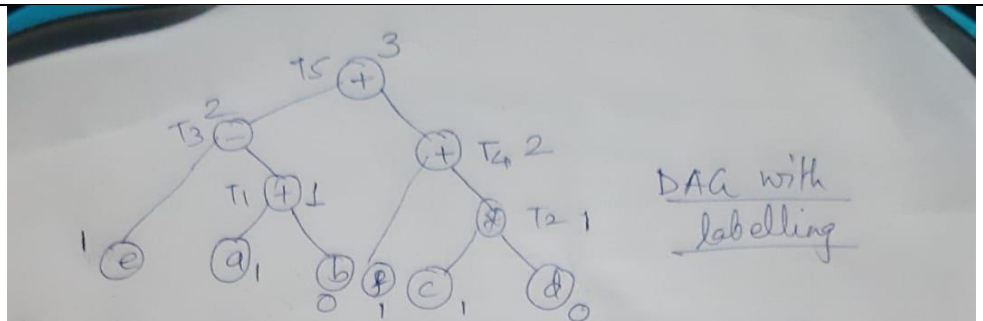
Q.1	Construct the intermediate Three Address Code (TAC) using SDTS for the given language construct. Show the annotated parse tree and the TAC generated. Also write the SDTS of while loop.
	while(not(a>b)) do
	begin
	if(c>5) then
	c=c-1;
	end
	What is the significance of S.next . Also give the value of target of S.next.
		5 M	CO3	L3

Solution				

Q.2	Perform IN OUT analysis of the following program flow graph.  Compute the UD chain for the statement c = a + 2 in block B2 and c=c + 1 in block B4 and find whether it is loop variant computation or not.	5 M	CO4	L3, L4
Solution	 			

	The loop for the given PFG is: B2-B3-B4-B2 So definition 1 is outside the loop, hence c = a + 2 is loop invariant computation Now consider c=c + 1, UD chain(c)= IN (B4) = {1, 2, 3, 4, 5, 7} = {3 } Since definition 3 is inside the loop, so c=c + 1 is not loop invariant computation			
3 (a) **Solution**	Generate the optimal order of execution for the following 3 address code using Heuristic Algorithm. Now generate target code for this optimal order using simple code generation algorithm. Calculate the total cost. T1=x * y T2=z + x T3=w / T2 T4=T3 + T1  order = reverse of listing = 4 3 2 1 i.e T2 T1 T3 T4 T2 = z + x T1 = x * y T3 = w / T2 T4 = T1 + T3 → Simple code generation algo Cost | Statement | L | Instn | Reg addr. | Addr. Regs R0, R1 → empty 2 | T2=z+x | R0 | MOV z, R0 | R0 → T2 2 | | | ADD x, R0 | R1 → empty 2 | T1=x*y | R1 | MOV x, R1 | R0 → T2 2 | | | MUL y, R1 | R1 → T1 2 | T3=w/T2 | R1 | MOV R1, M1 | R1 → empty | M1 → T1 R0 → T2 2 | | | MOV w, R1 | R1 → T3 1 | | | DIV R0, R1 | R0 → empty 2 | T4=T1+T3 | | MOV M1, R0 | R1 → T3, R0 → T1 1 | | | ADD R1, R0 | R0 → T4 R1 → empty 2 | | | MOV R0, T4 18	5 M	CO4	L4
	OR			
3 (b)	Determine the optimal number of registers for the computation: T1=a + b T2=c * d T3=e - T1 T4=f + t2 T5=T3 + T4 Also generate the target code using gencode() procedure.	5 M	CO4	L5

Solution



Call to gencode	Cases	procedure	RSTACK	code generation
			R0 R1 R2	MOV e, R0
gencode(T5)	Case 3	1) gencode (T3)		MOV a, R1
		P1) R = R0		ADD b, R1
		12) gencode (T4)	R1 R2	SUB R1, R0
		21) ADD R1, R0		MOV f, R1
		22) PUSH R0	R0 R1 R2	MOV C, R2
				MUL d, R2
				ADD R2, R1
gencode(T3)	Case 3	2) gencode (e)	R0 R1 R2	MOV e, R0
		4) R = R0	R1 R2	MOV a, R1
		5) gencode (T1)		ADD b, R1
		9) SUB R1, R0		SUB R1, R0
		10) PUSH R0	R0 R1 R2	
gencode(e)	case 0	3) MOV e, R0	R0 R1 R2	MOV e, R0
gencode(T1)	case 1	name = b	R1 R2	MOV a, R1
		6) gencode (a)		ADD b, R1
		8) ADD b, R1		
gencode(a)	case 0	7) MOV a, R1	R1 R2	MOV a, R1
gencode(T4)	case 3	13) gencode (f)	R1 R2	MOV f, R1
		15) R = R1	R2	MOV C, R2
		16) gencode (T2)		MUL d, R2
		19) ADD R2, R1	R1 R2	ADD R2, R1
		20) PUSH R1		
gencode(f)	case 0	14) MOV f, R1	R1 R2	MOV f, R1
gencode(T2)	case 1	17) name = d	R2	MOV C, R2
		19) gencode (c)		MUL d, R2
		19) MUL d, R2		
gencode(c)	case 0	18) MOV C, R2	R2	MOV C, R2