

DSA-Numerical **Prepare By Dilbar Yadav**

Clz_NMSS_Bharatpur-chitwan

* Infix to Postfix conversion

→ Operands - A, B, x, y, P, Q, ...

→ Operators - +, -, /, *, (,) ...

⇒ A + B = Infix

Rules

① Priorities of operators

^ → Highest priority

*, / → Next priority

+, - → Lowest priority

(operator between)

⇒ AB+ = Postfix

(operator after
operand)

② No two operators of same priority can
stay together in stack column.

③ Lowest priority can't be placed before highest
priority.

* operand - Postfix
* operator - stack

Eg. (A+B/C)*(D+C)-F)

Ans Here,

Steps	Symbol	Stack	Postfix
1	(C	
2.	A	(A
3	+	(+	A
4	B	(+)	AB
5	/	(+/	AB
6	C	(+/	ABC
7.	*	(+*)	ABC/
8.)	(+*)	ABC/

steps	symbol	stack	Postfix
9.	D	L * (A B C D
10	+	(+ * (+	A B C D
11	((+ * (+	A B C D C
12)	(+ * (+)	A B C D C
		(+ *	A B C D C +
13	-	(-	A B C D C + * +
14	F	(-	A B C D C + * +
			F
15)	(F)	A B C D C + * +
			F -

∴ A B C | D C + * + F - is the reqd postfix expression.

INPPx to P~~o~~SSPx

Operators	Associativity
^	Right to Left
*, /	Left to Right
+, -	Left to Right

Steps

- ① Reverse the infix expression.
- ② Obtain the "nearly" postfix expression of the modified expression.

3. Reverse the postfix expression.

Q. Convert the following infix expression into prefix expression.

$K + L - M * N + (O^P) * W / U \vee * T + Q$.

Ans Here,

Reverse the expression

$Q + T * \vee / U / W * (P^O) + N * M - L + K$

steps	symbol	stack	Prefix expression
1	+	Q	Q
2	+	+ +	Q
3	T	+ + T	Q T
4	*	+ * + T	Q T
5	\vee	+ * + T \vee	Q T \vee
6	/	+ * / + T \vee	Q T \vee
7	U	+ * / U + T \vee	Q T U \vee
8	/	+ * / / U + T \vee	Q T U \vee
9	W	+ * / / U W + T \vee	Q T U W \vee
10	*	+ * / / * U W + T \vee	Q T U W \vee
11	(+ * / / * (U W + T \vee	Q T U W \vee
12	P	+ * / / * (P U W + T \vee	Q T U W P \vee
13	^	+ * / / * (^ P U W + T \vee	Q T U W P \vee
14	O	+ * / / * (^ O P U W + T \vee	Q T U W P O \vee
15)	+ * / / * (^) P U W + T \vee	Q T U W P O \vee
16	+	+ + + + + + + +	Q T U W P O \vee * / / *

steps	symbol	stack	prefix expression
17	N	++	QTVUWPO^*//KN
18	*	++*	QTVUWPO^*//KN
19	M	++*	QTVUWPO^*//KNM
20	-	++-	QTVUWPO^*//KNM
			*
21	L	++-	QTVUWPO^*//KNM
			*L
22	+	++-+	QTVUWPO^*//KNM
			*L
23	K	++-+	QTVUWPO^*//KNM
			*LK
			QTVUWPO^*//KNM
			*LK+-++

Again,

Reverse the expression

AC

++-+KL*MN*//*^OPWUVTA | 95

the reqd prefix expression.

CTEVET (2030)

2. convert following expression to prefix and postfix.

$$A^2B * C - D + E / F / (G + H)$$

Ans Here,

converting expression into prefix

$$A^2B * C - D + E / F / (G + H)$$

Reverse the expression

$$(H + G) . / F / E + D - C * B ^ A$$

steps	symbol	stack	prefix expression
1.	(A C	C
2.	H	(H
3.	+	(+	H
4.	G	(+ G	H G
5.)	(+)	H G +
6.	/	/	H G +
7.	F	/	H G + F
8.	/	//	H G + F
9.	E	//	H G + F E
10	+	+ //	H G + F E //
11	D	+	H G + F E / D
12	-	+ -	H G + F E / / D
13	C	+ -	H G + F E / / D C
14	*	+ - *	H G + F E / / D C

Gurans NoteBook

Date: 20 / /

Steps	Symbol	Stack	Prefix expression
15	B	+-*	HG+FE//DCB
16	A	+-*^	HG+FE//DCB
17	A	+-*^	HG+FE//DCBA
			<u>HG+FE//DCBA^*-+</u>

Again,

Reverse the expression

[+-*^ABCD//EF+GH] is the reqd prefix expression.

Again,

Converting infix into postfix expression

A ^ B * C - D + E / F / (G + H)

Steps	Symbol	Stack	Postfix expression
1	A		A
2	^	^	A
3	B	^	AB
4	*	*	AB^
5	C	*	AB^C
6	-	-	AB^C*
7	D	-	AB^C*D
8	+	+	AB^C*D-
9	E	+	AB^C*D-E
10	/	+/	AB^C*D-E

Gurans Note Book

Date: 20 / /

steps	symbol	stack	Postfix expression
11	F	+/	AB^C*D-EF
12	/	+/	$AB^C*D-EF/$
13	(+/(C	$AB^C*D-EF/$
14	G	+/(C	AB^C*D-EF/G
15	+	+/(+	AB^C*D-EF/GH
16	H	+/(+	AB^C*D-EF/GH
17)	+/(+)	$AB^C*D-EF/GH+$
			<u>$AB^C*D-EF/GH+I$</u>

∴ $AB^C*D-EF/GH+I$ is the reqd postfix expression.

CBSEET (2080)

AC

Q. Convert the following infix expression to postfix and prefix expression.

$(A+B)/(C+D)*E-F$

Ans Here, Converting into Postfix

steps	symbol	stack	Postfix expression
1	(C	
2	A	(A
3	+	(+	A
4	B	(+)	AB
5	/	(+)	AB
6)	(+/(C	AB

Steps	Symbol	Stack	Postfix Expression
7	C	(+ / C	ABC
8	+	(+ / C +	ABC
9	D	(+ / C + D	ABCD
10	\$	(+ / C + \$	ABCD
11	E	(+ / C + \$ E	ABCDE
12	*	(+ / C + \$ * E	ABCDE
13	#	(+ / C + \$ * F	ABCDEF
14)	(+ / C + \$ * F)	ABCDEF*\$ +
			ABCDEF*\$ + / + (
$\therefore \boxed{ABCDEF*$ + / + (}$ is the reqd postfix expression.			

Converting Infix expression into prefix

$(A+B/(C+D\$E*F))$

Now,

Reverse the expression

$(F*E\$D+C)/B+A)$

Gurans Note Book

Date: 20 / /

Steps symbol stack Prefix expression

1 C C

2 F C F

3 * (* F

4 E (*E FE

5 \$ (*\$ FE

6 D (*\$D FED

7 + (+ FED\$*

8 C (+C FED\$*C

9) (+) FED\$*C+

10 / / FED\$*C+

11 B / FED\$*C+B

12 + + FED\$*C+B/

13 A + FED\$*C+B/A

14) +) FED\$*C+B/A

Again, FED\$*C+B/A+)

Reverse the expression

|+A/B+C*\$DEF| is the reqd prefix expression

Ans //

Gurans Note Book

Date: 20 / /

~~QUESTION 20~~ convert the following infix expression
into prefix and postfix expression.

$$A \times B / C + D - E \times F + G$$

Ans Here,

Converting infix into prefix

$$A \times B / C + D - E \times F + G$$

Reverse the expression

$$G + F - (E A D) * C / B S A$$

Step	symbol	Stack	Prefix expression
1	G		G
2	+	+	GA
3	F	+F	GAF
4	-	+F-	GAF-
5	(+F-(GAF-
6	(+F-(()	GAF-
7	E	+F-(()E	GAF-E
8	A	+F-(()EA	GAF-EA
9	D	+F-(()EA	GAF-EAD
10)	+F-(()	GAF-EAD
11	*	+F-(*)	GAF-EAD
12	C	+F-(*)C	GAF-EADC
13)	+F-(*)C)	GAF-EADC*
14	/	+F-(*)C/	GAF-EADC*
15	B	+F-(*)C/B	GAF-EADC*B
16	S	+F-(*)C/B S	GAF-EADC*B S

Gurans Note Book

Date: 20 / /

Step	Symbol	Stack	Prefix expression
1	A	+ /	^FEADC*B3A
			<u>^FEADC*B3A/-+ </u>

Reverse the expression

$+ / A3B * C(DAEF) - F + G$ is the reqd prefix expression.

again,

Converting infix into postfix

$A S B / C C * (D A E) - F + G$

Step	Symbol	Stack	Postfix expression
1	A		A
2	S		AS
3	B		ASB
4	/	/	ASB
5	C	/C	ASB
6	C	/C	ASBC
7	*	/C*	ASBC
8	(/(*()	ASBC
9	D	/(*()	ASBCD
10	A	/(*()	ASBCDA
11	E	/(*()	ASBCDAE
12)	/(*()	ASBCDAE
13)	/(*)	ASBCDAE*

Step	Symbol	Stack	Postfix expression
14	-	-	A B C D A E * /
15	F	-	A B C D A E * / F
16	+	+	A S B C D A E * / F -
17	G	+	A S B C D A E * / F - G
			A S B C D A E * / F - G +]

∴ [A S B C D A E * / F - G +] is the reqd postfix expression.

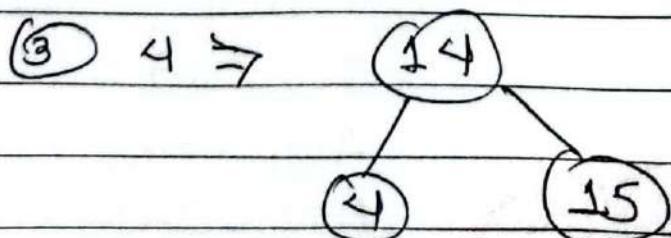
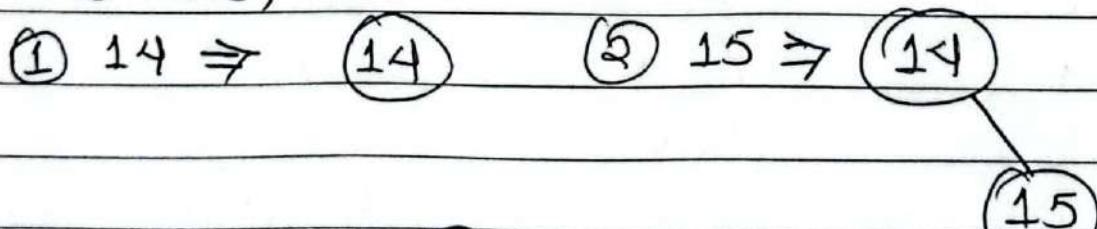
Binary Search Tree (BST)

- * Element less than root → left side
- * Element greater than root → right side

LTEVER (2079)

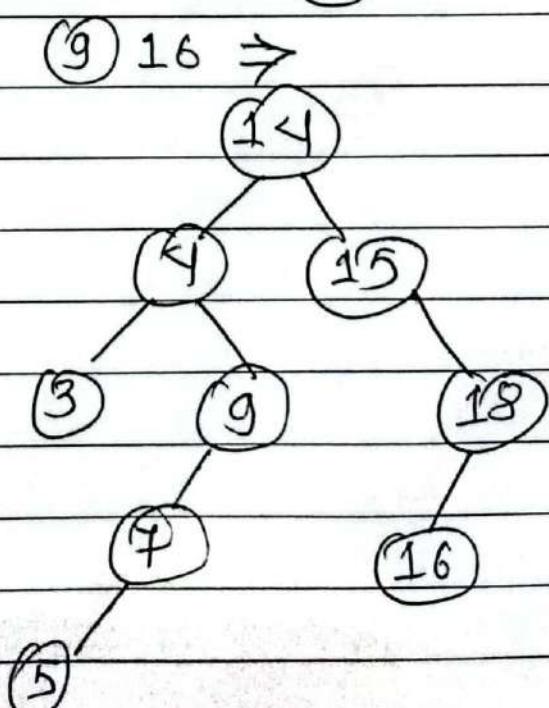
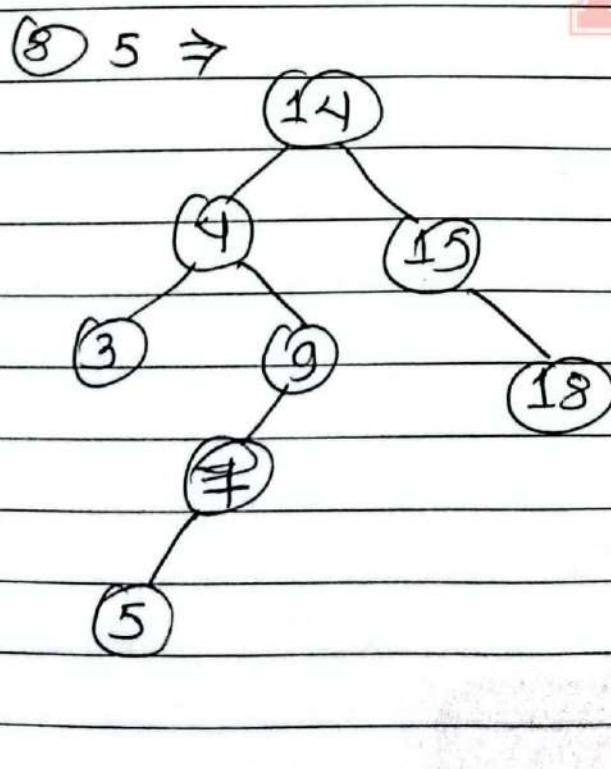
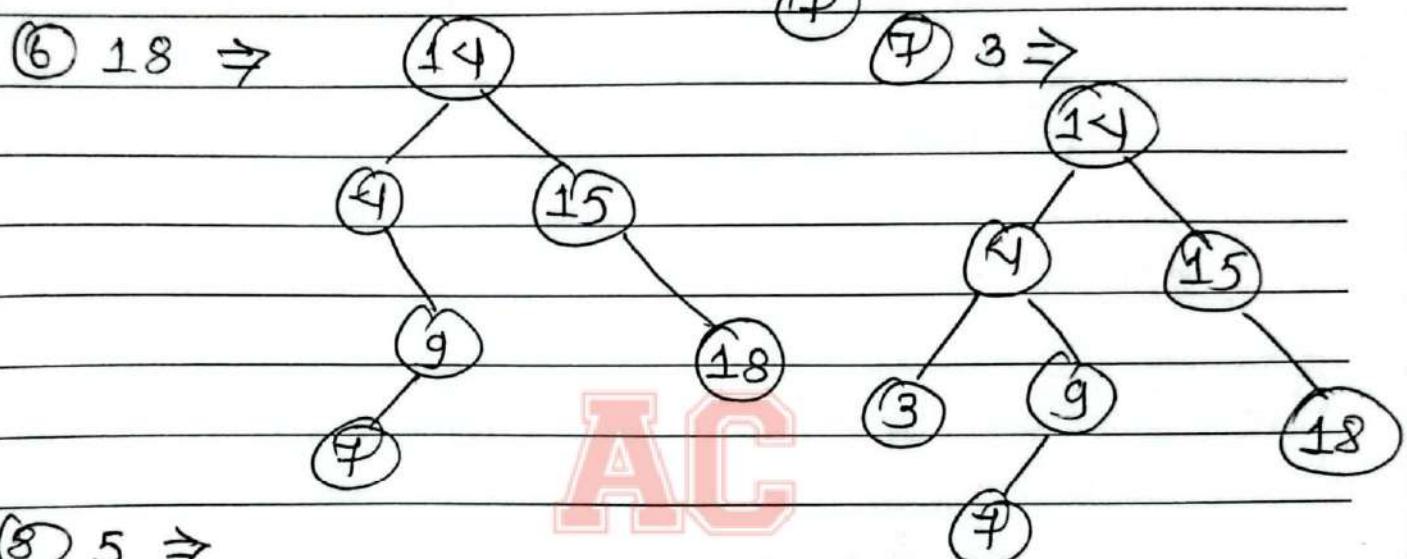
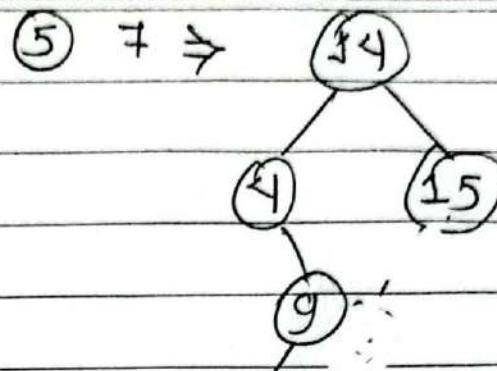
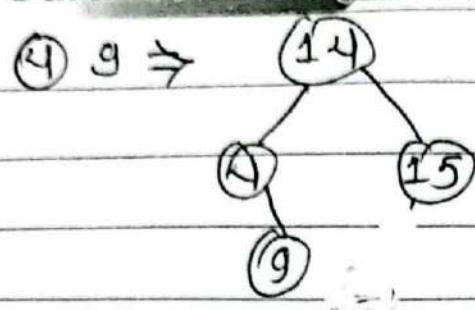
Q. 1. Draw BST for 14, 15, 4, 9, 7, 18, 3, 5, 16, 11, 20, 17, 9, 14, 5.

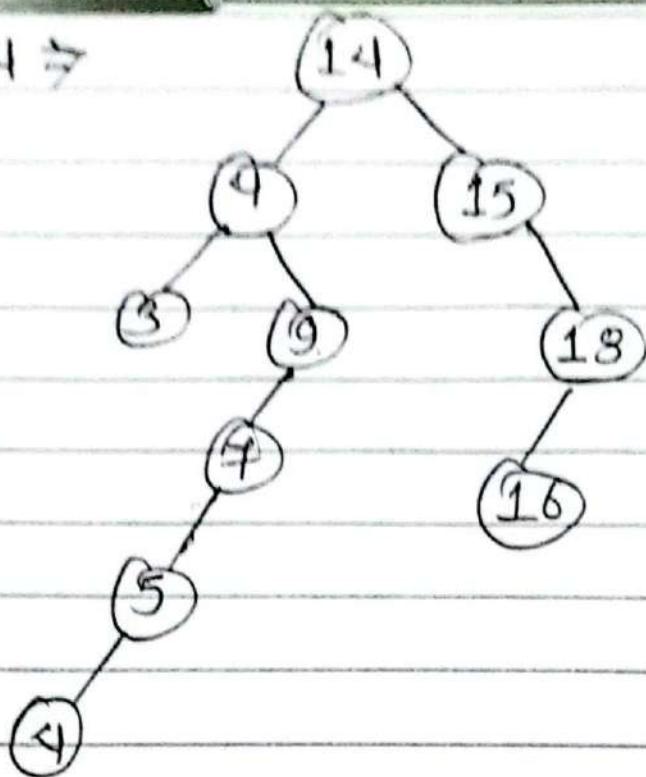
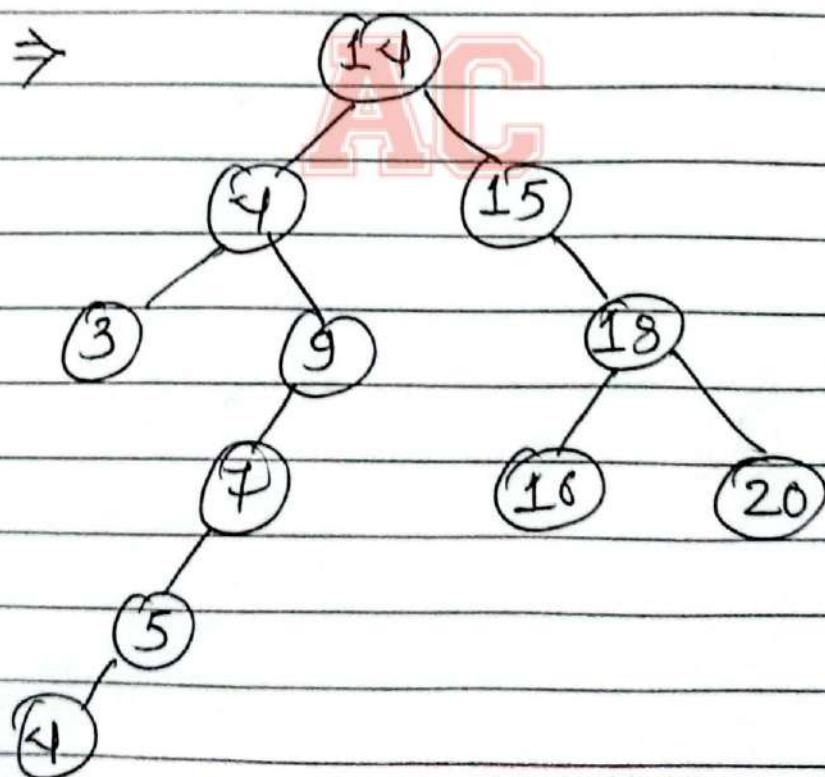
Ans Here,



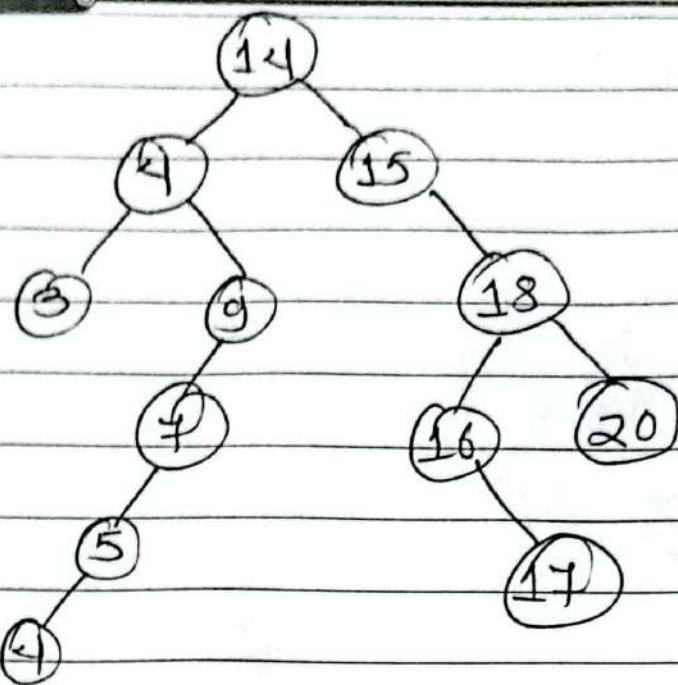
Gurans Note Book

Date: 20 / /

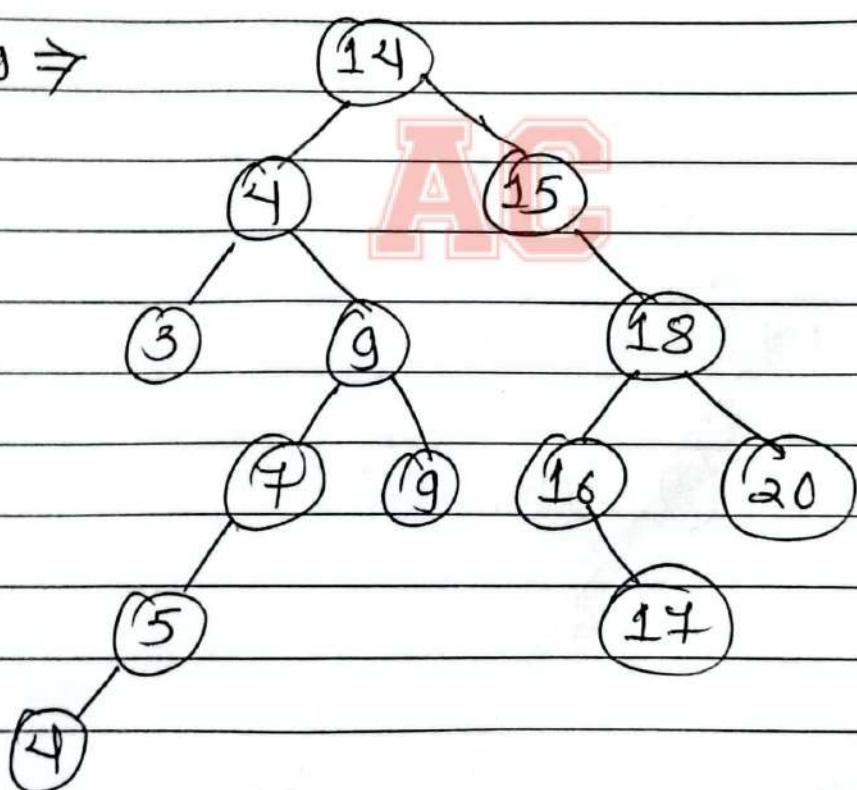


(10) $\leftarrow \Rightarrow$ (11) 20 \Rightarrow 

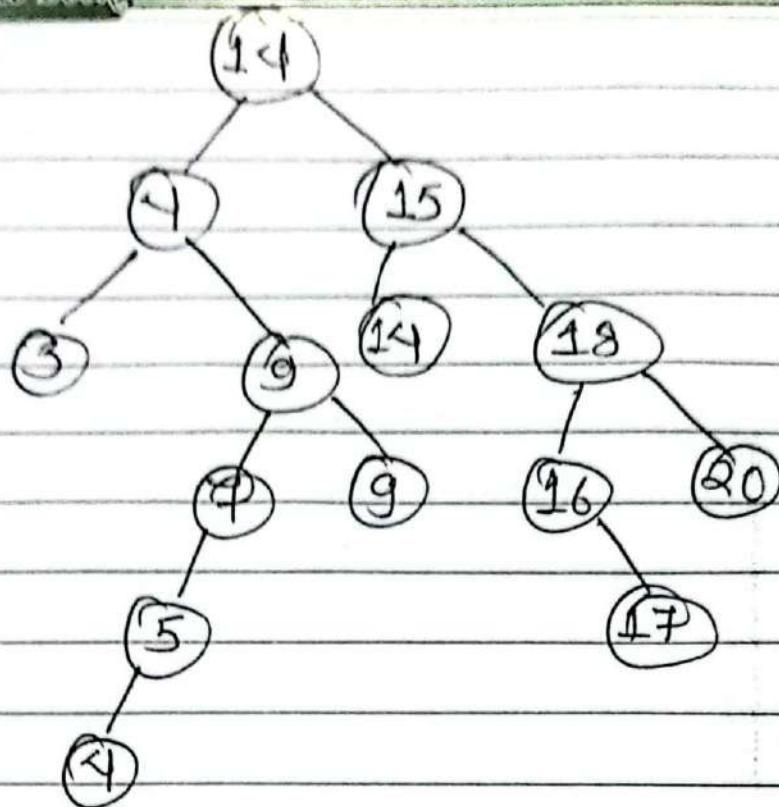
(12) 14 ⇒



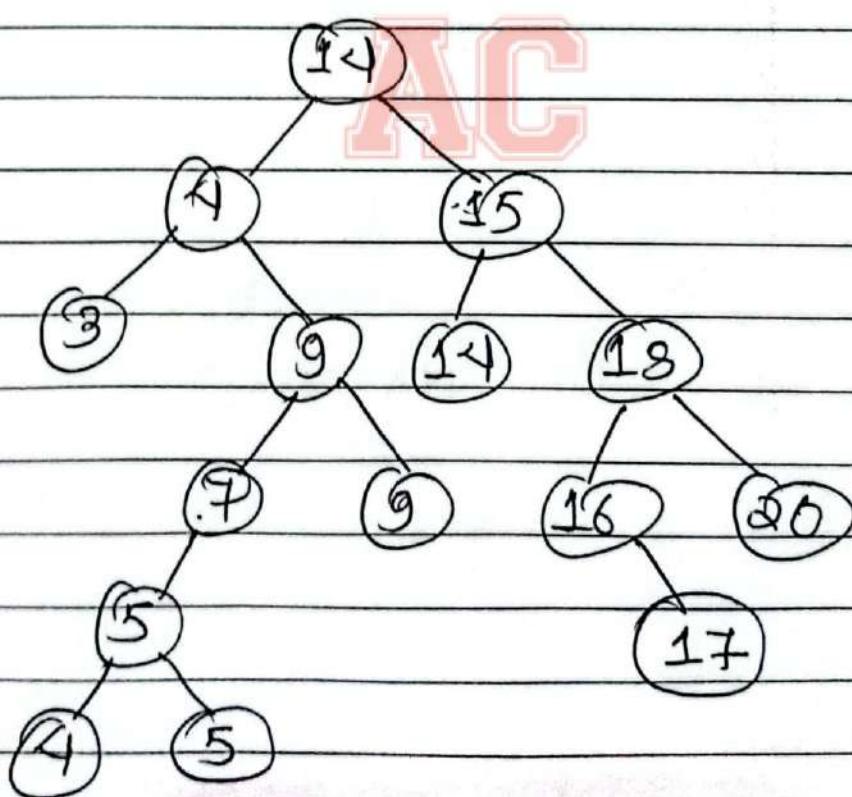
(13) 9 ⇒

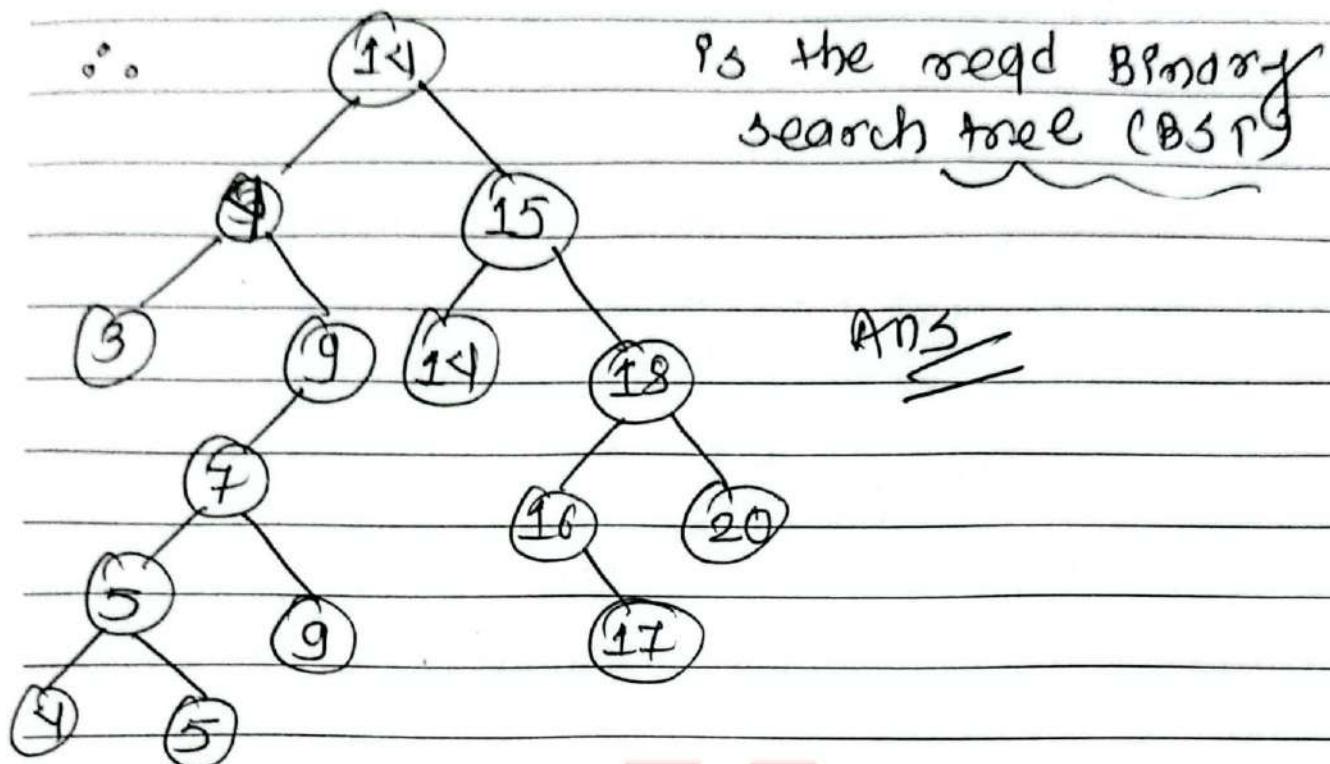


(14) 14 ⇒



(15) 5 ⇒





AVL Tree (Self Balancing Tree)

* Balance Factor (B.F) = Height of left subtree - Height of right subtree

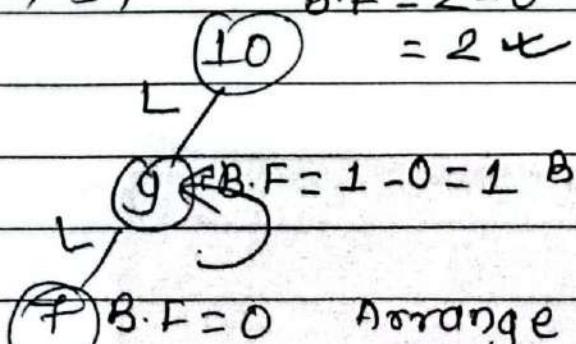
* Conditions to be satisfied

$$= \{-1, 0, 1\} \Rightarrow \text{value of B.F}$$

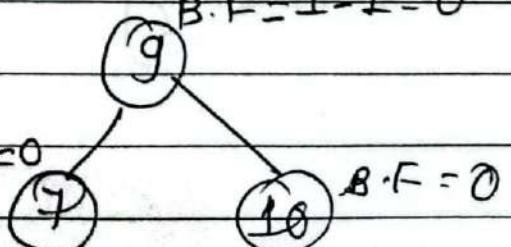
Types of Rotation

1. Left to Left Rotation (LL Rotation)

$$10, 9, 7 \quad B.F = 2 - 0 = 2 \leftarrow$$



$$B.F = 1 - 1 = 0$$



B.F = 0 Arrange in

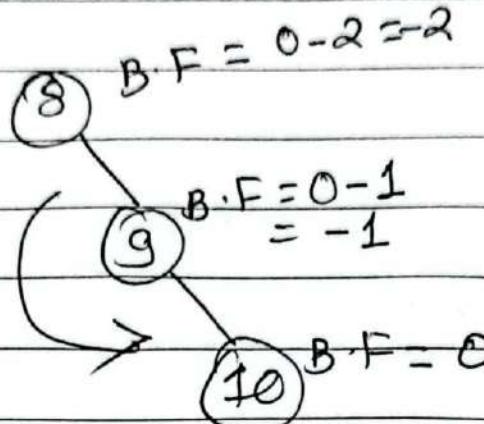
∴ clockwise
rotation.

Gurans Note Book

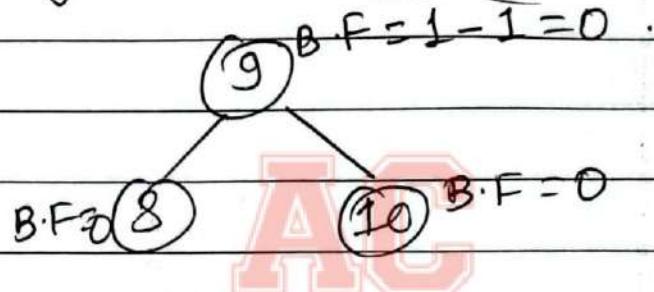
Date: 20 / /

997 RR Rotation

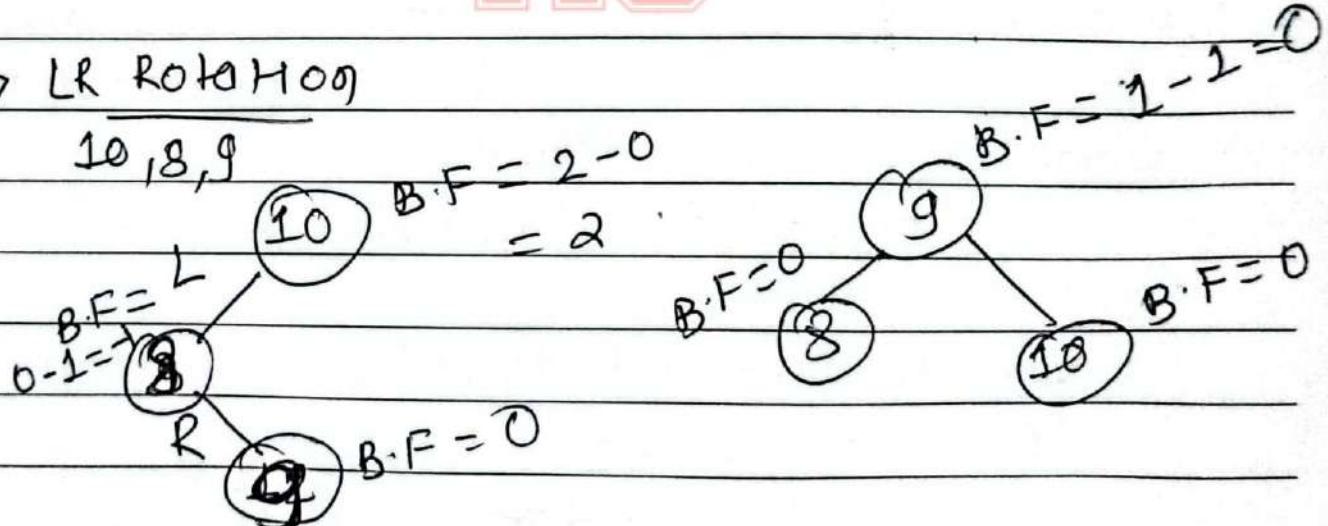
8, 9, 10



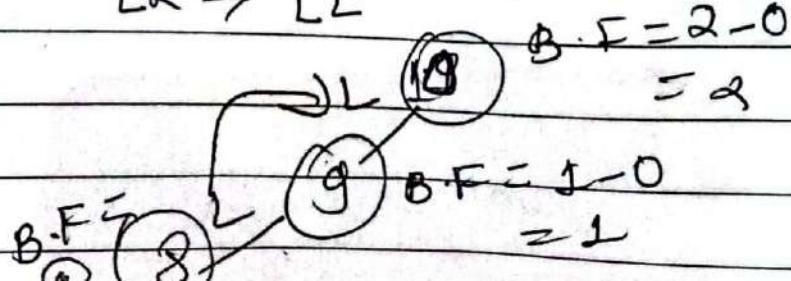
Arrange in anti-clockwise

999 LR Rotation

10, 8, 9



Convert first into LL

 $LR \rightarrow LL$ 

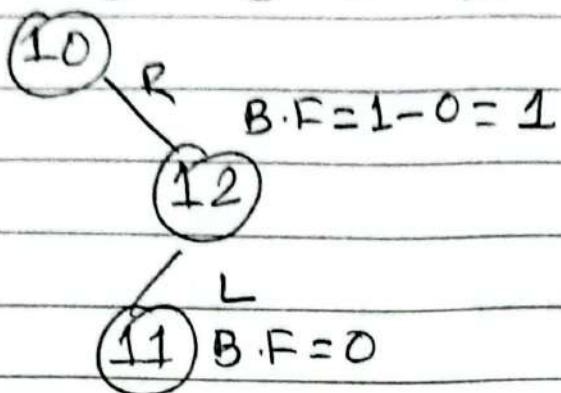
Gurans Note Book

Date: 20 / /

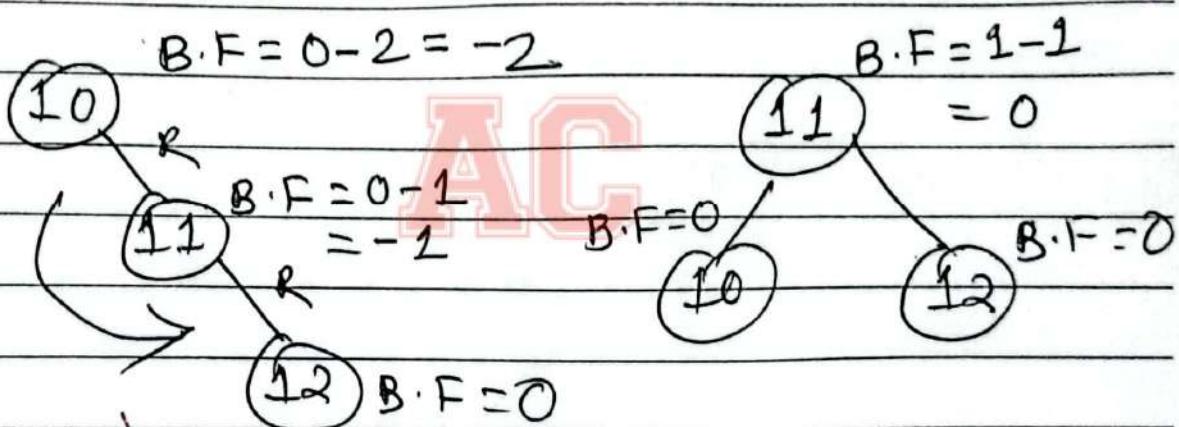
Q. RL Rotation

10, 12, 11

$$B.F = 0 - 2 = -2$$



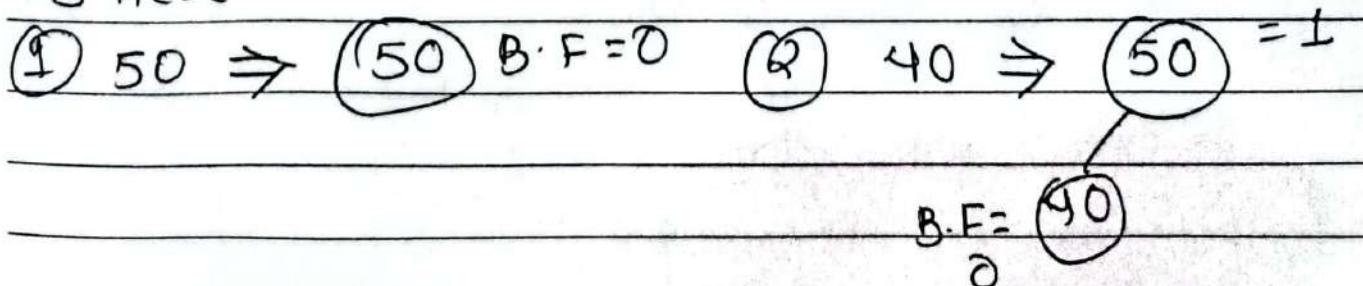
$RL \rightarrow RR$



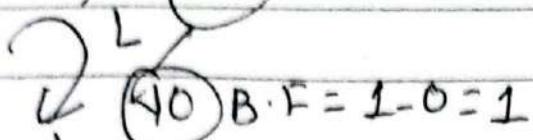
QUESTION (2080)

Q. Draw a AVL tree for: 50, 40, 35, 58, 48, 42, 60, 30, 33, 22.

Ans Here .

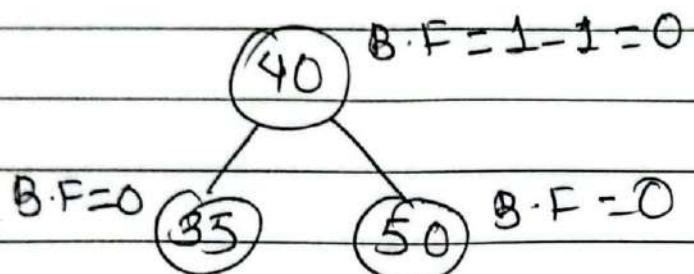


$$\textcircled{3} \quad 35 \Rightarrow \textcircled{50} \quad B.F = 2 - 0 = 2$$

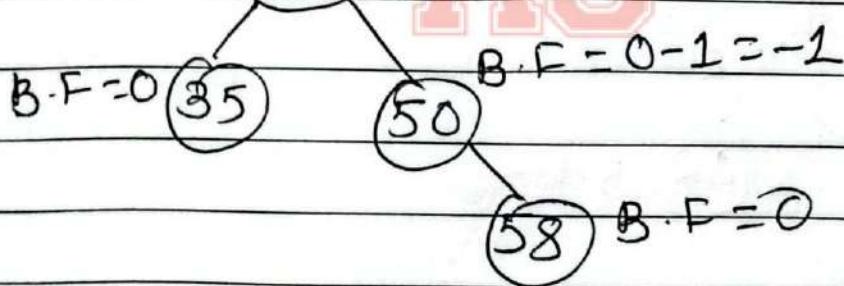


$$\textcircled{35} \quad B.F = 0$$

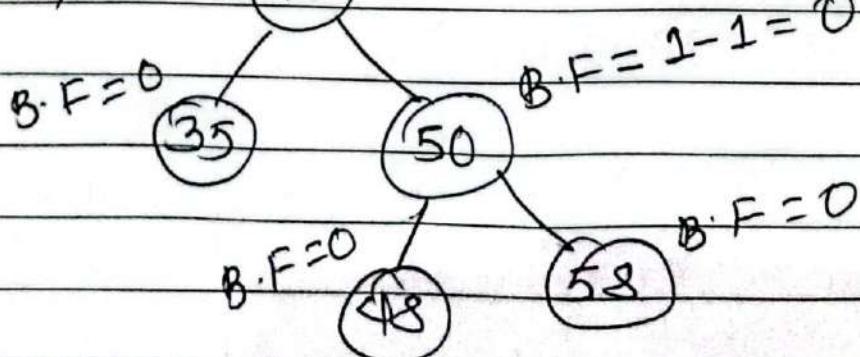
LL Rotation



$$\textcircled{4} \quad 58 \Rightarrow \textcircled{40} \quad B.F = 1 - 2 = -1$$

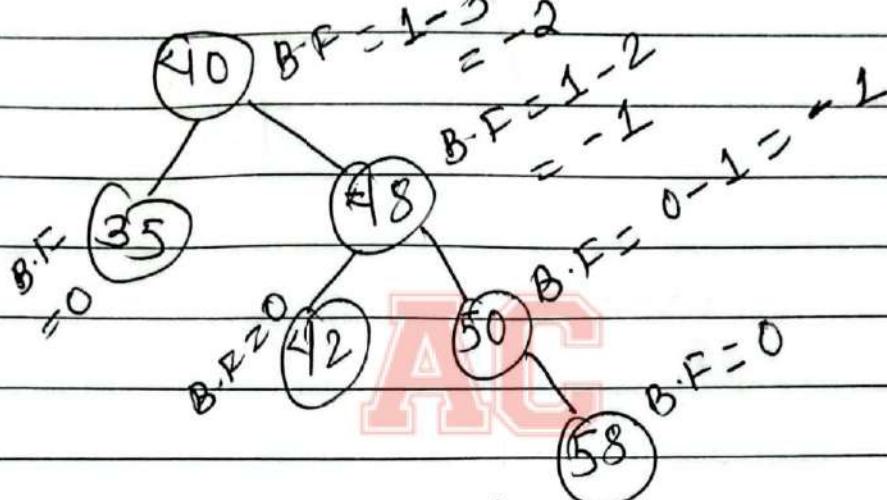
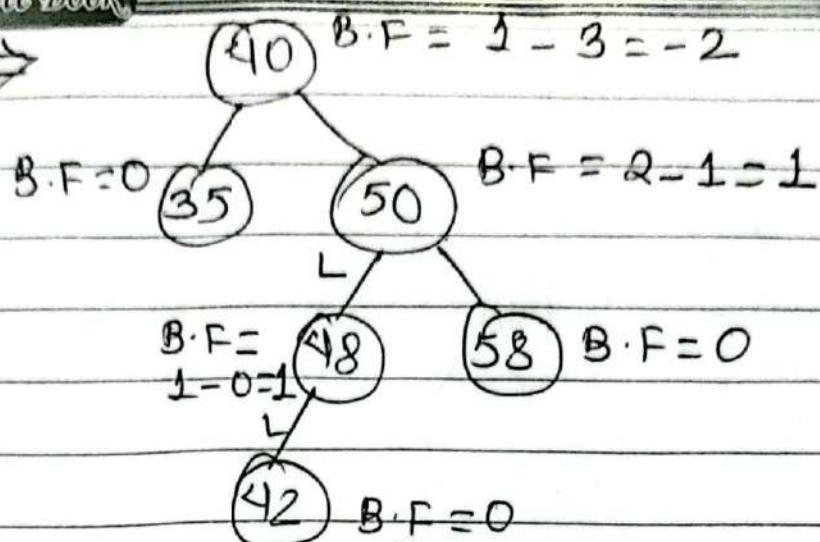
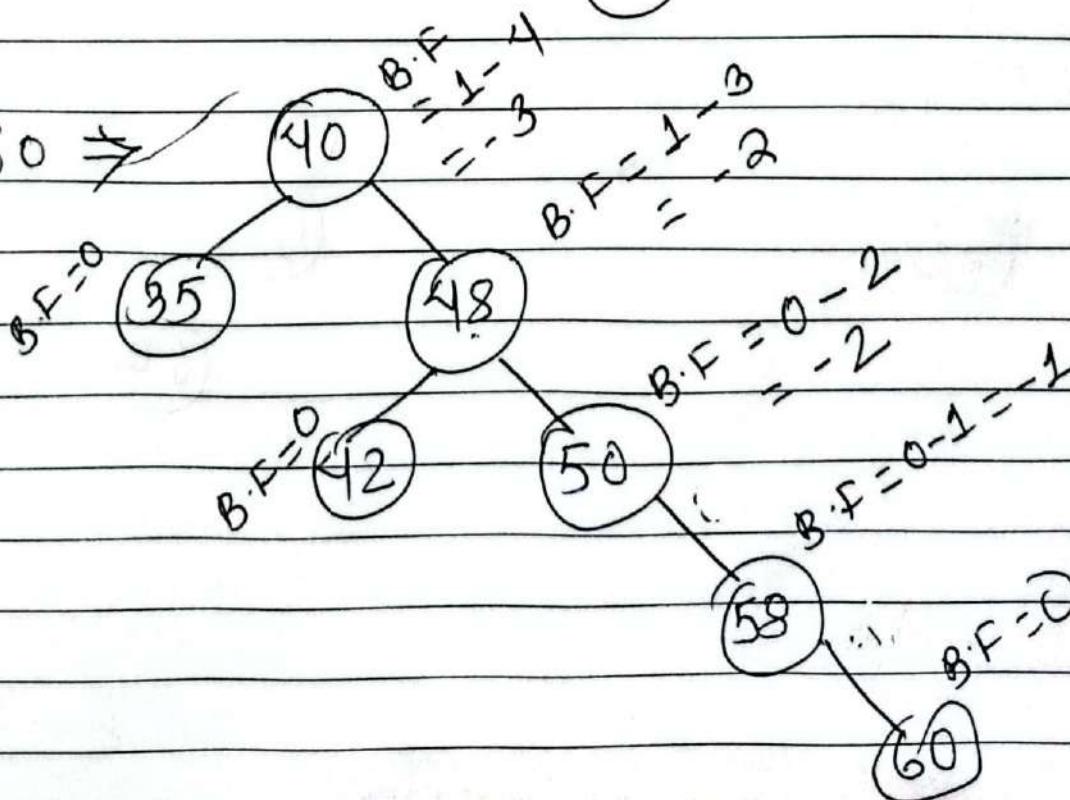


$$\textcircled{5} \quad 48 \Rightarrow \textcircled{40} \quad B.F = 1 - 2 = -1$$

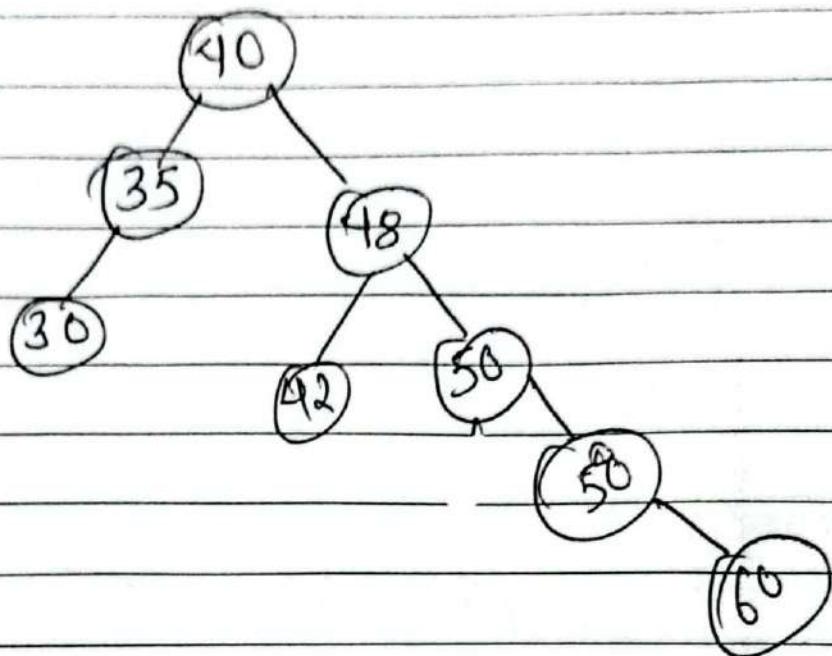


Gurans Note Book

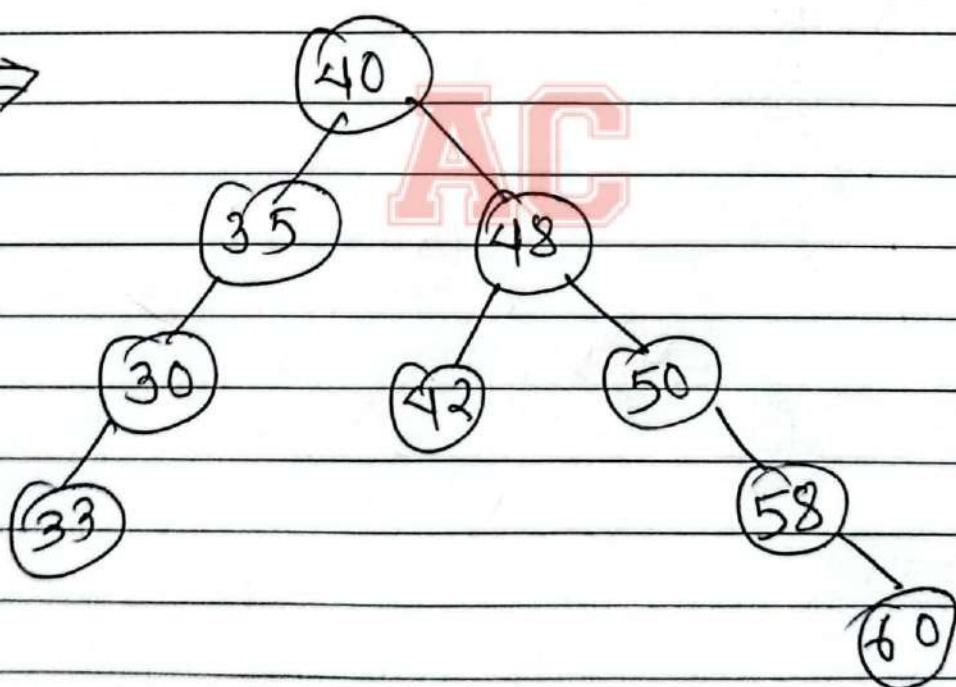
Date: 20 / /

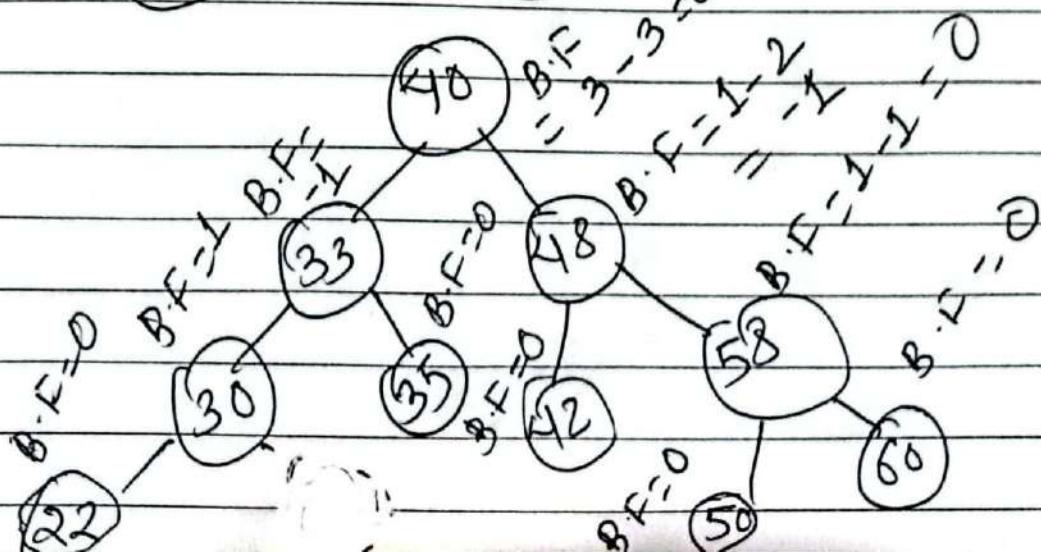
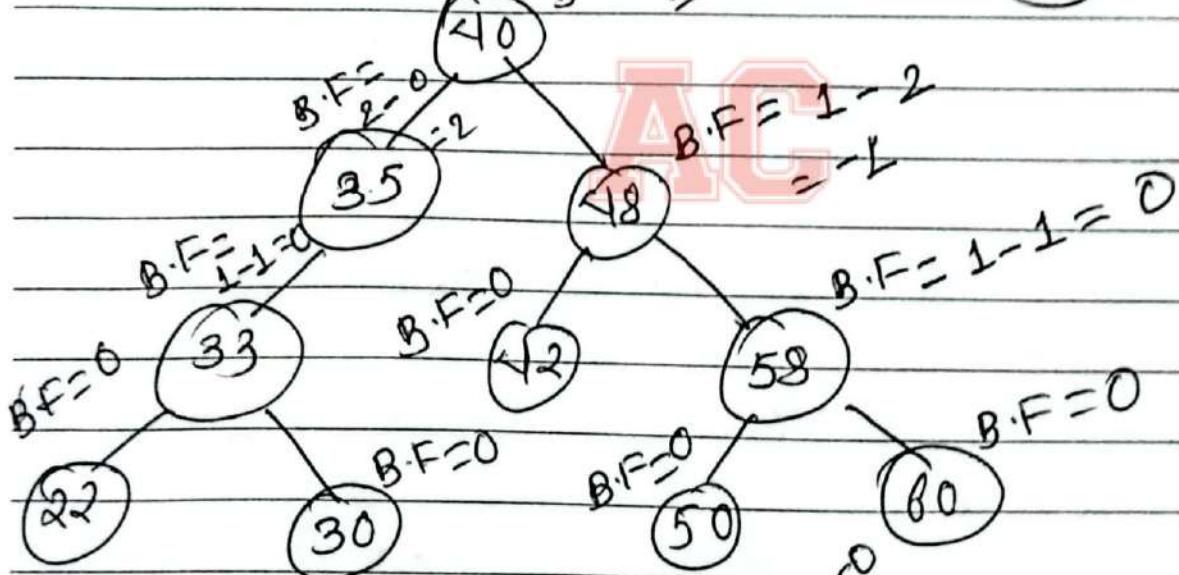
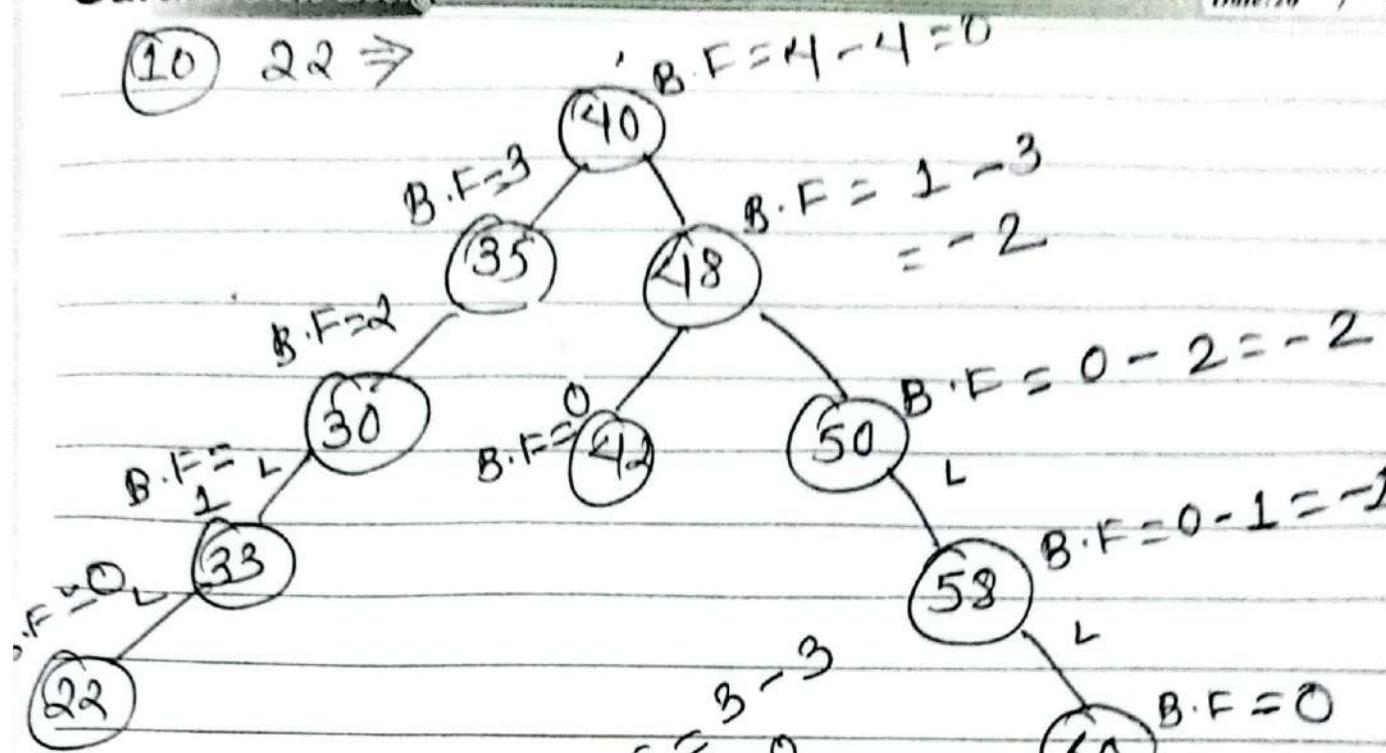
(6) $42 \Rightarrow$ (7) $60 \Rightarrow$ 

87 30 ⇒



⑨ 33 ⇒

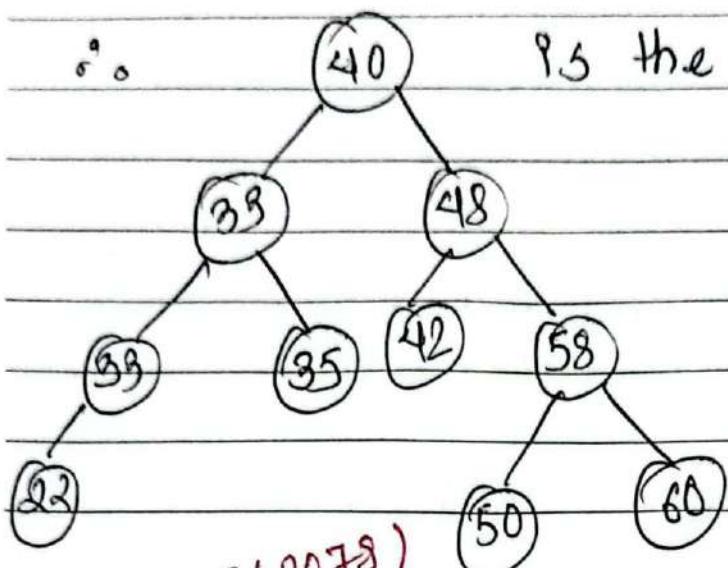


(10) $22 \Rightarrow$ 

Q.

40

Is the reqd AVL tree.



Ans

CTEET (2018)

Q. Draw the AVL tree for the following sequence of data:

2, 7, 6, 4, 9, 10, 12, 8, 5

AC

Ans Here,

(1) 2 \Rightarrow (2) B.F = 0(2) 7 \Rightarrow (2) B.F = -1

$$\begin{aligned} B.F &= 0 - 1 \\ &= -1 \end{aligned}$$

$$B.F = 0$$

(7)

(3) 6 \Rightarrow

(2)

$$\begin{aligned} B.F &= 0 - 2 \\ &= -2 \end{aligned}$$

RR Rotation

$$\begin{aligned} B.F &= 1 - 0 \\ &= 1 \end{aligned}$$

RL \rightarrow RR

(2)

$$B.F = 0 - 2$$

$$B.F = 0$$

(2)

(6)

$$\begin{aligned} B.F &= 1 - 1 \\ &= 0 \end{aligned}$$

$$\begin{aligned} B.F &= 0 \\ &= 0 \end{aligned}$$

(7)

R

$$B.F = 0 - 1$$

(6)

R

$$B.F = -1$$

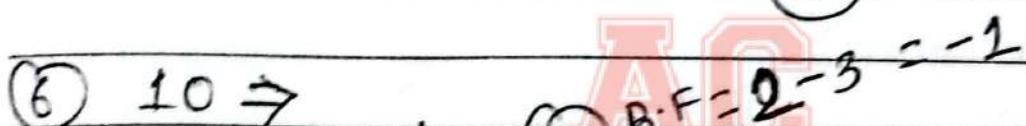
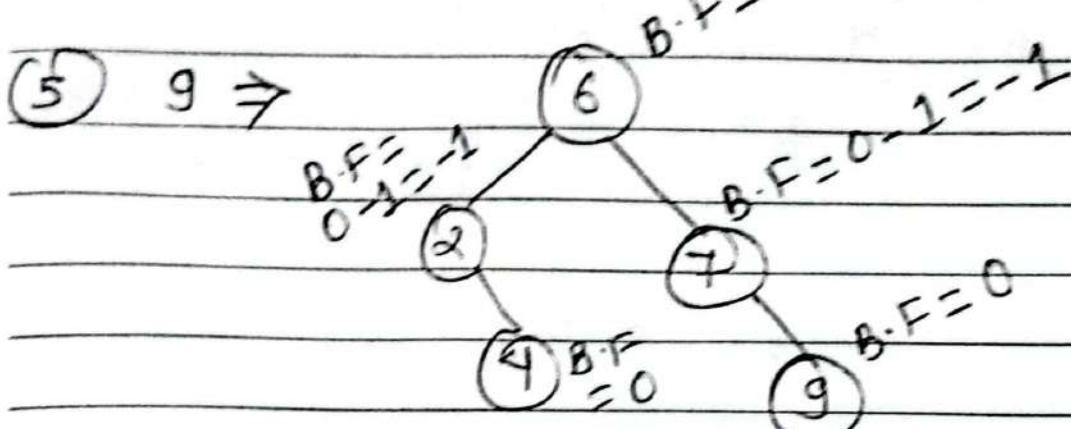
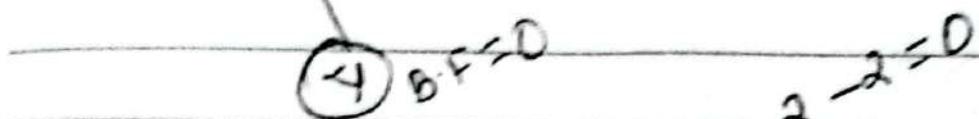
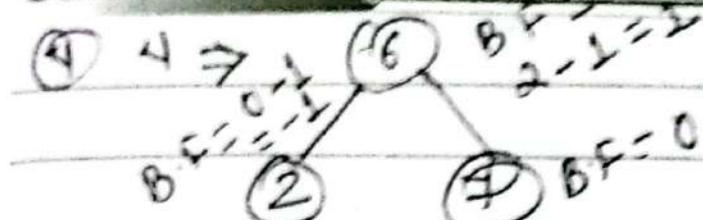
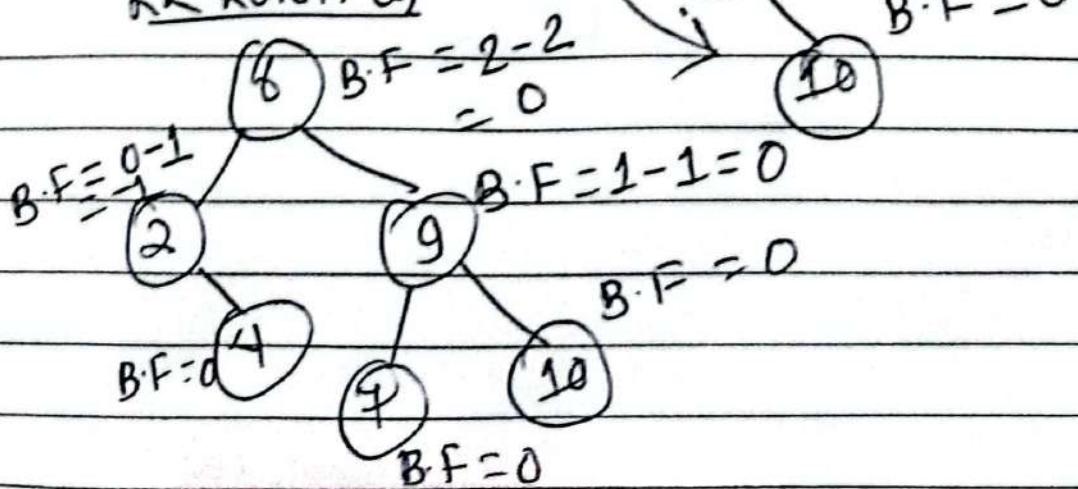
(7)

R

$$B.F = 0$$

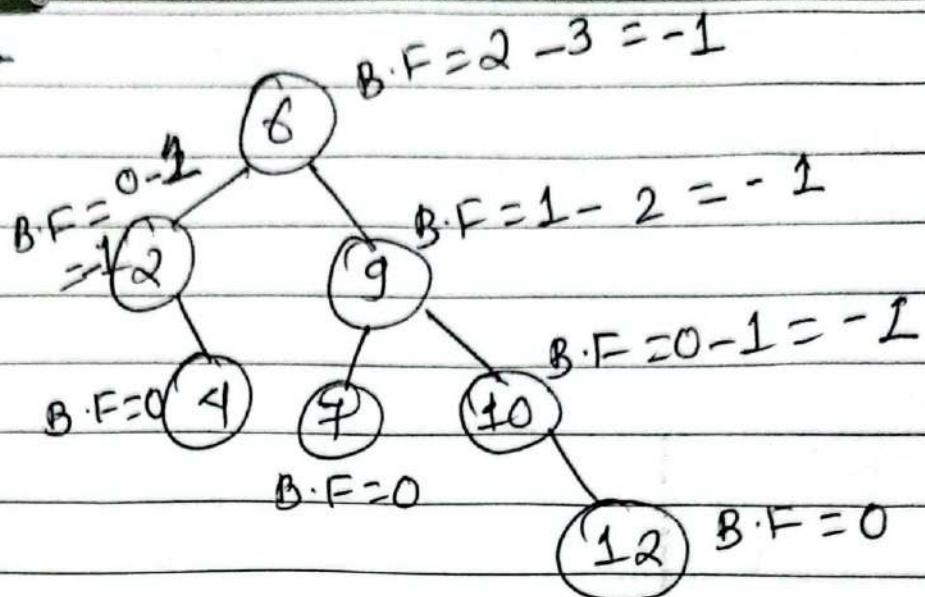
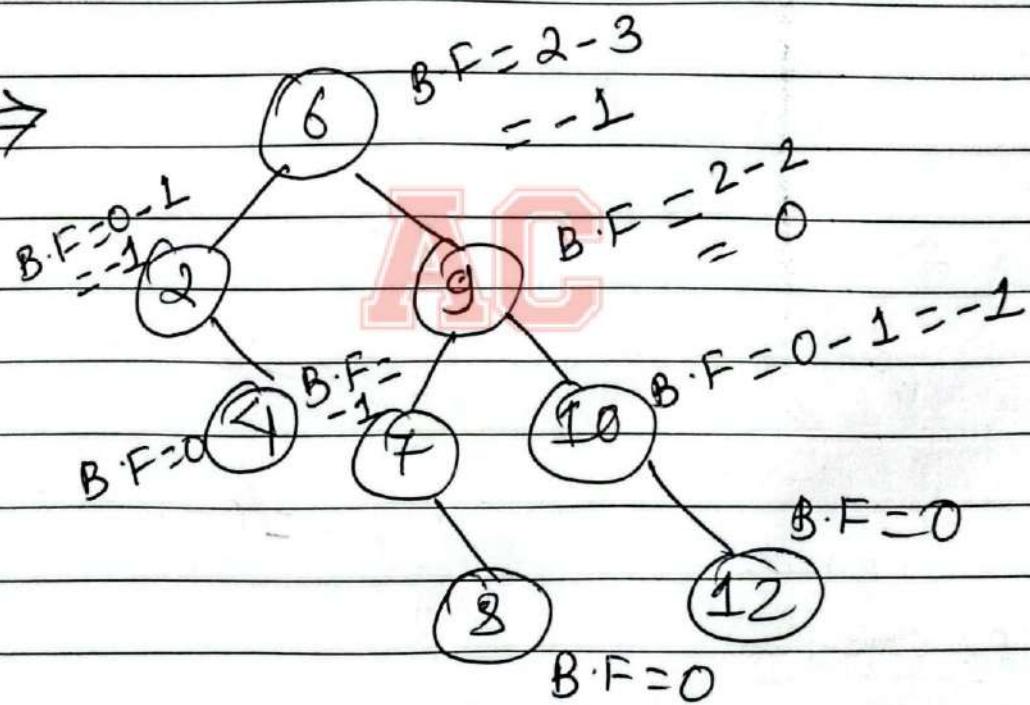
Gurans *Hand Book*

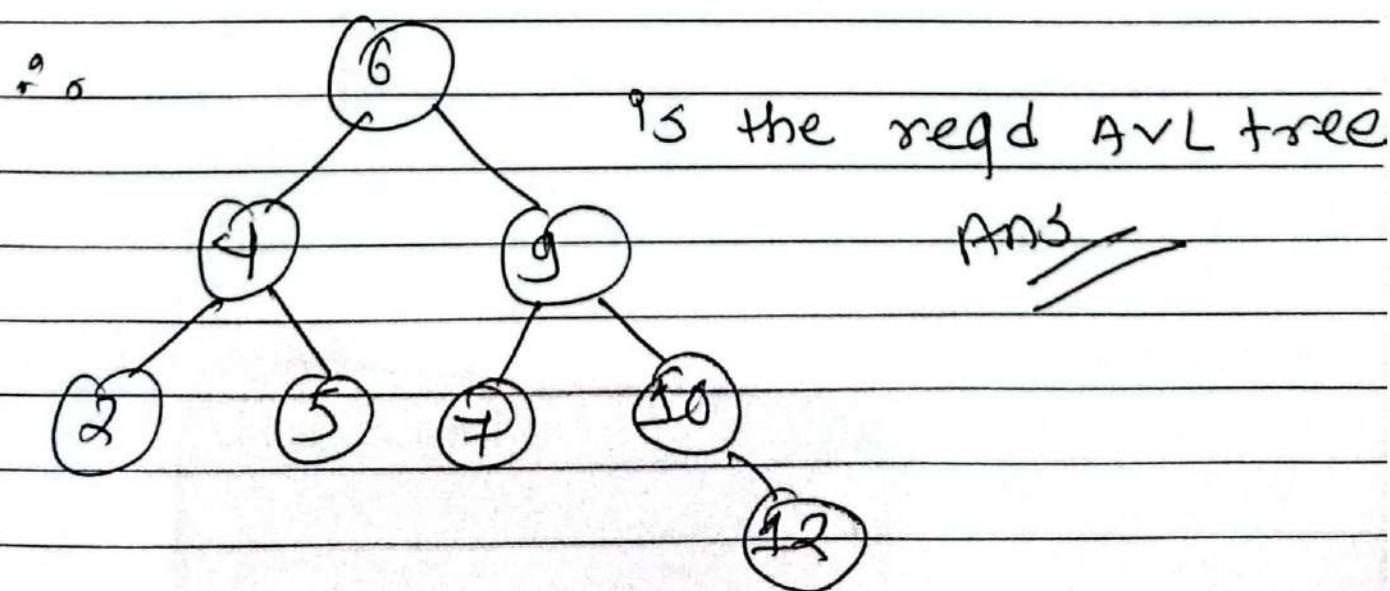
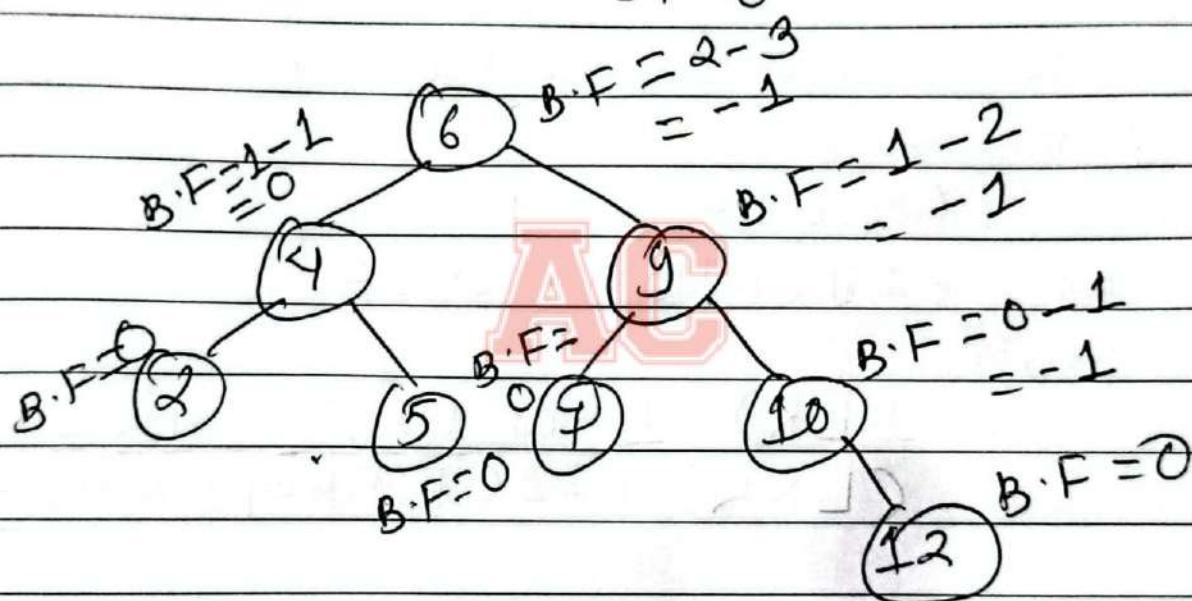
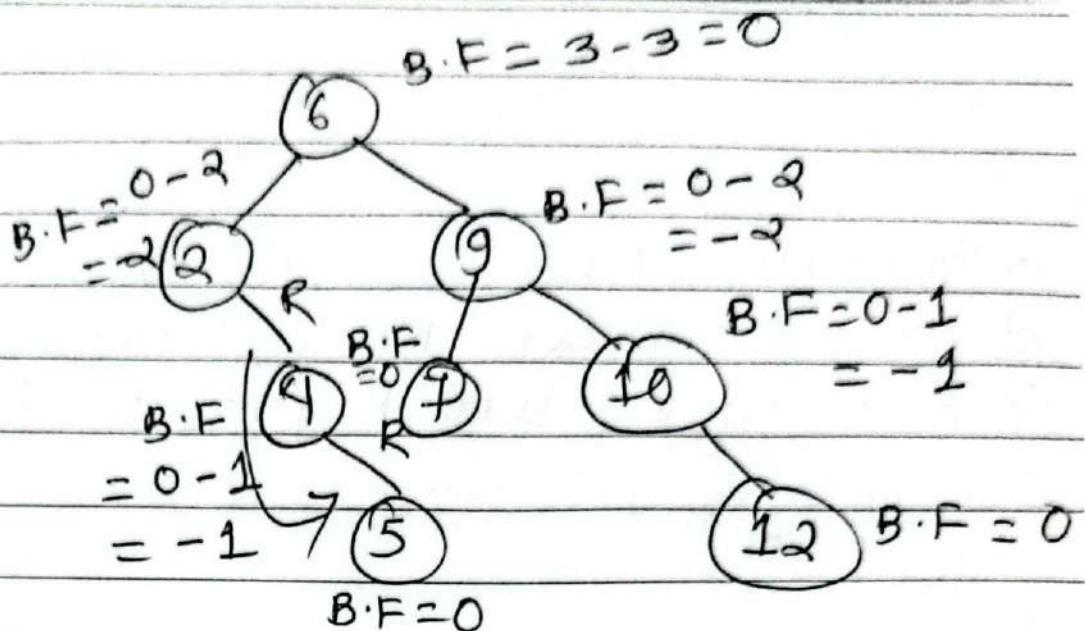
Date: 20 / /

RR Rotation

Gurans Note Book

Date: 20 / /

(7) 12 \Rightarrow (8) 8 \Rightarrow 

(9) $5 \Rightarrow$ 

Preorder, Inorder and Postorder (Creating Binary tree)

- * Preorder (Node, left, right) $\Rightarrow (N, L, R)$
- * Inorder (left, Node, right) $\Rightarrow (LNR)$
- * Postorder (left, right, Node) $\Rightarrow (LRN)$

CREAT (2080)

Q. What is binary search tree? Draw the binary search tree for

Preorder: ABCEIFJDUHKI

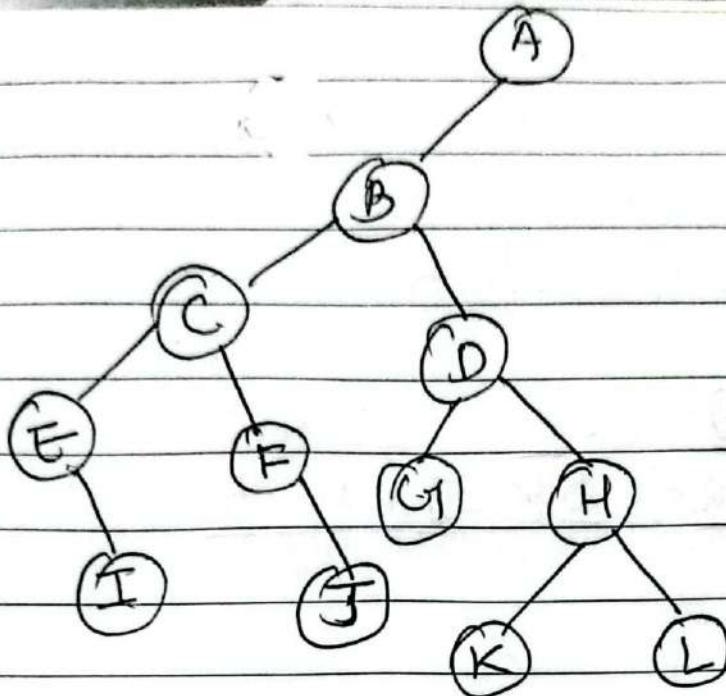
Inorder : EI¹CFJBUDKHLA

Ans Here,

Preorder (NLR) : ABC EIFJDU HKL

Inorder (LNR) : EI¹ CFJBUDKHLA

Now,



Is the reqd binary search tree.

~~GATEET (2020)~~

AC

- Q. Construct a binary tree from given Preorder and Inorder sequence.

Preorder: AB D C E H I F

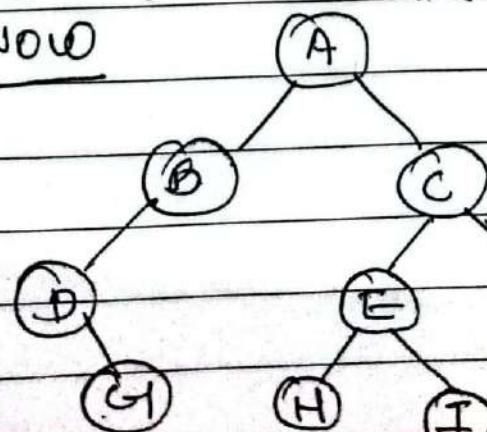
Inorder: D G B A H E I C F

Ans Here,

Preorder (NLR) : A B D C E H I F

Inorder (LNR) : D G B A H E I C F

NOW



Is the reqd
Binary search
tree (BST)

Ans //

Q. construct a binary tree from given inorder and postorder traversal.

Inorder: DBFEAGCLJHK

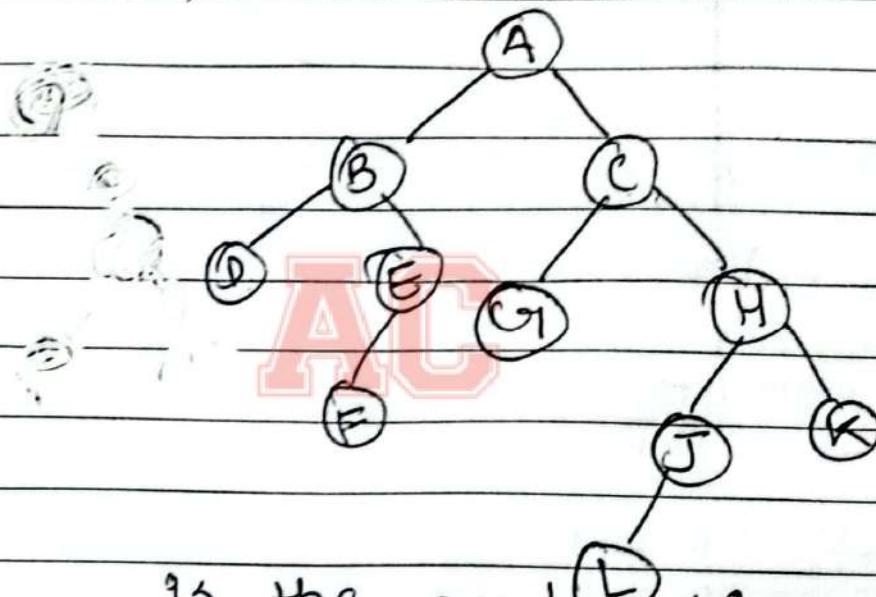
Postorder: DFEBA^HGLIKCA

Ans Here,

Inorder (LNR) : DBFEAGCLJHK

Postorder (LRN) : DFEBA^HGLIKCA

NOW,



is the reqd binary tree.

~~Ans~~

Sorting
CREVER (2081)
BATCH

- Q. Define sorting and searching sort the following numbers using merge and bubble sort: 32, 8, 3, 15, 11, 21, 6, 5.
- Ans Here,

Using Merge sort

Elements 32, 8, 3, 15, 11, 21, 6, 5
Now,

Arranging elements in pairs

32, 8 , 3 , 15 , 11 , 21 , 6 , 5
 [] [] [] []

8, 32 , 3, 15 , 11, 21 , 5, 6
 [] [] [] []

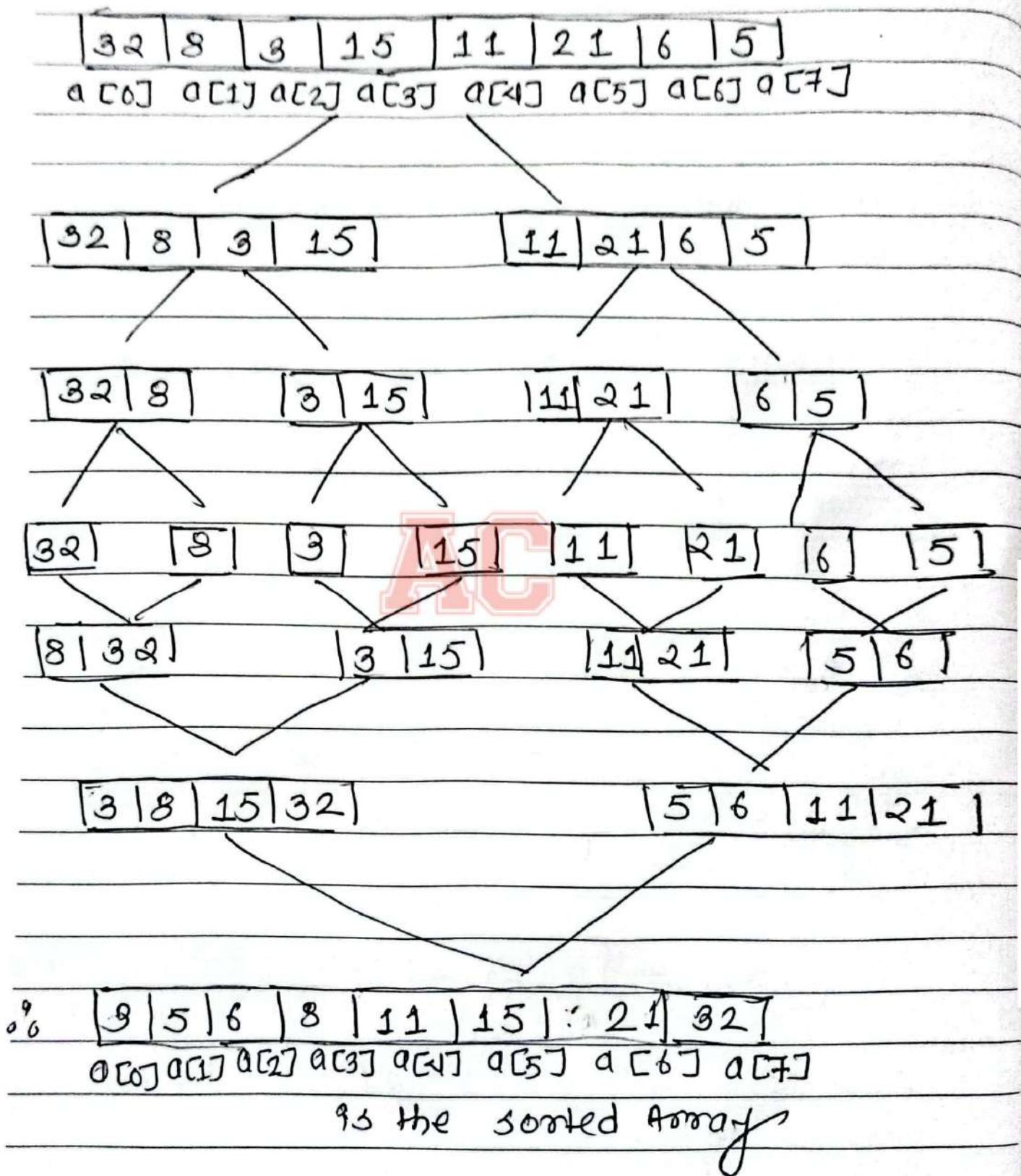
3, 8, 15, 32 . 5, 6, 11, 21
 [] []
 } }

3, 5, 6, 8, 11, 15, 21, 32 ,

∴ 3, 5, 6, 8, 11, 15, 21, 32 is the sorted arranged elements

Ans

Another Method . (Merge Sort)



is the sorted array

using Bubble Sort

32, 8, 3, 15, 11, 21, 6, 5

Now,

Step 1: 32 8 3 15 11 21 6 5

8	32	3	15	11	21	6	5
8	3	32	15	11	21	6	5
8	3	15	32	11	21	6	5
8	3	15	11	32	21	6	5
8	3	15	11	21	32	6	5
8	3	15	11	21	6	32	5
8	3	15	11	21	6	5	32

→ 32

Step 2: 8 3 15 11 21 6 5

3	8	15	11	21	6	5
---	---	----	----	----	---	---

3	8	15	11	21	6	5
---	---	----	----	----	---	---

3	8	11	15	21	6	5
---	---	----	----	----	---	---

3	8	11	15	21	6	5
---	---	----	----	----	---	---

3	8	11	15	6	21	5
---	---	----	----	---	----	---

3	8	11	15	6	5	21
---	---	----	----	---	---	----

Step 3: 3 8 11 15 6 5

3	8	11	15	6	5
---	---	----	----	---	---

3	8	11	15	6	5
---	---	----	----	---	---

3	8	11	15	6	5
---	---	----	----	---	---

3	8	11	6	15	5
---	---	----	---	----	---

3	8	11	6	5	15
---	---	----	---	---	----

Gurans Note Book

Date: 20 / /

Step 4 : $\begin{array}{r} 3 \quad 8 \quad 11 \quad 6 \quad 5 \\ 3 \quad 8 \quad \underline{11} \quad 6 \quad 5 \\ 3 \quad 8 \quad \underline{11} \quad 6 \quad 5 \\ 3 \quad 8 \quad 6 \quad \underline{11} \quad 5 \\ 3 \quad 8 \quad 6 \quad 5 \quad \boxed{11} \rightarrow 11 \end{array}$

Step 5 : $\begin{array}{r} 3 \quad 8 \quad 6 \quad 5 \\ 3 \quad \underline{8} \quad 6 \quad 5 \\ 3 \quad 6 \quad \underline{8} \quad 5 \\ 3 \quad 6 \quad 5 \quad \boxed{8} \rightarrow 8 \end{array}$

Step 6 : $\begin{array}{r} 3 \quad 6 \quad 5 \\ \underline{3} \quad 6 \quad 5 \\ 3 \quad 5 \quad \boxed{6} \rightarrow 6 \end{array}$

Step 7 : $\begin{array}{r} 3 \quad 5 \\ \underline{3} \quad \boxed{5} \rightarrow 5 \end{array}$

Step 8 : $\boxed{3} \rightarrow 3$

∴ $\boxed{3} \boxed{5} \boxed{6} \boxed{8} \boxed{11} \boxed{15} \boxed{21} \boxed{32}$
 $a[0] \quad a[1] \quad a[2] \quad a[3] \quad a[4] \quad a[5] \quad a[6] \quad a[7]$
 is the sorted array.

~~CREVER (2080)~~

Q. Let us consider we have following data in array as 44, 33, 11, 55, 77, 90, 40, 60, 99, 22, 88. Now sort them using quick sort.

Ans Here, Using Quick sort

44	33	11	55	77	90	40	(60)	99	22	88
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]

$$\text{Pivot} = 60$$

44	(33)	11	55	20	22	60	77	90	99	88
0	1	2	3	4	5	6	7	8	9	10
Left of 60										Right of 60

$$\text{Pivot} = 33$$

AC

11	22	33	44	55	40	60	77	90	99	88
0	1	2	3	4	5	6	7	8	9	10
Left of 33 sorted			Right of 33							Right of 60

Left of 60

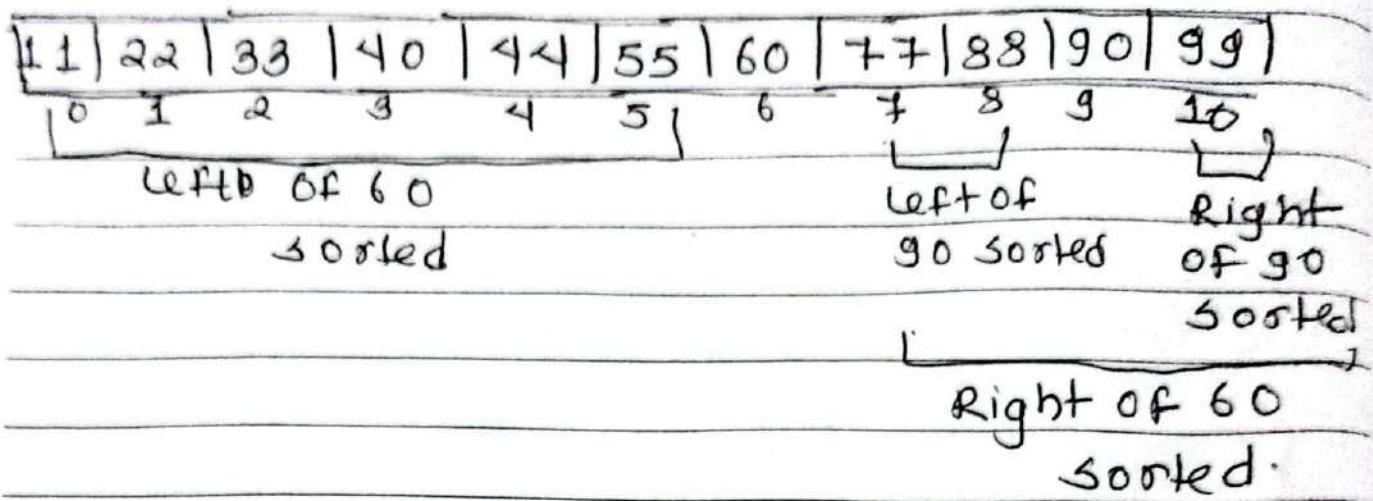
$$\text{Pivot} = 44$$

11	22	33	40	44	55	60	77	90	99	88
Left of 33		Left of 44		Right of 44						Right of 60
										Right of 60

Right of 44

$$\text{Pivot} = 90$$

Left of 60 sorted



∴ 11 | 22 | 33 | 40 | 44 | 55 | 60 | 77 | 88 | 90 | 99
 0 1 2 3 4 5 6 7 8 9 10

is the sorted array.

~~QUESTION (2080)~~

Q. Sort the following data using Selection sort : 64, 25, 12, 22, 11.

Ans Here,

64 | 25 | 12 | 22 | 11
 A[0] A[1] A[2] A[3] A[4]

↑
min

↑
loc

swap

PASS 1. 11 | 25 | 12 | 22 | 64
 A[0] A[1] A[2] A[3] A[4]

↑ ↑
min loc

swap

Pass 2 :

11	12	25	22	64
A[0]	A[1]	A[2]	A[3]	A[4]

↑ ↑
min loc
SWAP

Pass 3 :

11	12	22	25	64
A[0]	A[1]	A[2]	A[3]	A[4]

↑
min loc NO SWAP

∴ 11 12 22 25 64 is the sorted array.

~~CTEVEP(2079)~~ Q. Sort the following set of numbers using bubble sort and insertion sort.
 27, 38, 39, 0, 22, 18, 7, 15.

Ans Here,

Using insertion sort

27	38	39	0	22	18	7	15
A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]

~~27 < 38 27 < 39 27 > 0 (shift)~~

[0] | 27 | 38 | 39 | 22 | 18 | 7 | 15]

27 > 22 (shift)

[0] | 22 | 27 | 38 | 39 | 18 | 7 | 15]

22 > 18 (shift)

[0] | 18 | 22 | 27 | 38 | 39 | 7 | 15]

18 > 7 (shift)

[0] | 7 | 18 | 22 | 27 | 38 | 39 | 15]

7 < 15

18 > 15 (shift)

∴ [0] | 7 | 15 | 18 | 22 | 27 | 38 | 39] is the

A[0] A[1] A[2] A[3] A[4] A[5] A[6] A[7]

reqd sorted array.

using bubble sort

Pass 1:	27	<u>38</u>	39	0	22	18	7	15
	27	<u>38</u>	<u>39</u>	0	22	18	7	15
	27	<u>38</u>	<u>39</u>	0	22	18	7	15
	27	38	0	<u>39</u>	<u>22</u>	18	7	15
	27	38	0	22	<u>39</u>	18	7	15
	27	38	0	22	18	<u>39</u>	7	15
	27	38	0	22	18	7	<u>39</u>	15
	27	38	0	22	18	7	15	<u>39</u> → 39

Pass 2:	<u>27</u>	<u>38</u>	0	22	18	7	15
	<u>27</u>	<u>38</u>	0	22	18	7	15
	27	0	<u>38</u>	<u>22</u>	18	7	15
	27	0	22	<u>38</u>	18	7	15
	27	0	22	18	<u>38</u>	7	15
	27	0	22	18	7	<u>38</u>	15
	27	0	22	18	7	15	<u>38</u> → 38

Pass 3:	<u>27</u>	0	22	18	7	15
	0	<u>27</u>	22	18	7	15
	0	22	<u>27</u>	<u>18</u>	7	15
	0	22	18	<u>27</u>	7	15
	0	22	18	7	<u>27</u>	15
	0	22	18	7	15	<u>27</u> → 27

Gurans Note Book

Date: 20 / /

Pass 4: 0 22 18 7 15
 0 22 18 7 15
 0 18 22 7 15
 0 18 7 22 15
 0 18 7 15 22 → 22

Pass 5: 0 18 7 15
 0 18 7 15
 0 7 18 15
 0 7 15 18 → 18

Pass 6: 0 7 15
 0 7 15
 0 7 15 → 15

Pass 7: 0 7 → 7

Pass 8: 0 → 0

∴ [0 7 15 18 22 27 38 39]
 A[0] A[1] A[2] A[3] A[4] A[5] A[6] A[7]

is the sorted array.

Ans
✓

~~CTEVE 2078~~ Q. What is sorting? Sort the following list of numbers using insertion sort.

44, 33, 55, 77, 90, 40, 60, 99, 22, 88, 66

Ans Here,

using insertion sort

44	33	55	77	90	40	60	99	22	88	66
0	1	2	3	4	5	6	7	8	9	10
44 > 33 (shift)										

33	44	55	77	90	40	60	99	22	88	66
0	1	2	3	4	5	6	7	8	9	10
44 < 55										
44 < 77										
44 < 90										
44 > 40 (shift)										

33	40	44	55	77	90	60	99	22	88	66
0	1	2	3	4	5	6	7	8	9	10
55 < 60										

33	40	44	55	60	77	90	99	22	88	66
0	1	2	3	4	5	6	7	8	9	10
33 > 22 (shift)										

Gurans Note Book

Date: 20 / /

22	33	40	44	55	60	77	90	99	88	66
0	1	2	3	4	5	6	7	8	9	10

90 > 88 (shift)

22	33	40	44	55	60	77	88	90	99	66
0	1	2	3	4	5	6	7	8	9	10

77 > 66 (shift)

22	33	40	44	55	60	66	77	88	90	99
0	1	2	3	4	5	6	7	8	9	10

20	22	33	40	44	55	60	66	77	88	90	99
0	1	2	3	4	5	6	7	8	9	10	

are the sorted array.

Ans //