newNode (val) {	
note = new nodel)	
node => key = val	
node->left = NVIL	_
node = right = NULL	
node -> height = 1	
veturn node	
	-
right Rotate (node)	ħ.
l = node > left	
r = node ->right	
7 - 100 ac - 7/10 m	
leright = node	
nodes left = v	
hode sheight = max (hoight (node sleft), height (node-right)) +1	
made la height = max (height (la left), height (laright)) + 1	
return x;	
leftRotate (node) {	
Kto v = node-sright	
l= node -> left	
r->left = node	
node right = l	
node > height = mare height (node > beft), height (mode snight) +1	
x = height = max (height (node > left) height (smode > right))+1	
	, \
return node:	
	-
	13/5

insert (node, key) ? if (node == NULL) node slift = insent (node slift, key) dhe if (key > node > key) node -> right = insert (node > right, ky) node -> height = 14 max (height (node sleft), height (node sright)) balance - getBalance (node) if (bolance >1 and key & node -sleft -> Key) return right Rotate (node) else if (balance 4-1 and key > node > right >key) return left Rotate (node) ele if (balance >1 and key > node > left > key)

node -> left = left Rotate (node -> left) retion RightRotate (mode) else if (bolance 2-1 and key 2 node -> right -> tey) node = right = rightRotate (node = right) return aftrotate (node) return node get Balance (node) [if (mode = = NULL) return 0 lie return (heightboode-sleft) - helght (mode-> right)

delebald root, key)
if (root == NULL) return voot
elv if (kay & rootskey)
root -> left = deleteNode (root -> left, key)
elic if (key > root > key)
root->right = deleteNode (root->right, key)
die
if (froot > left == NULL 11 Yout > right == NULL)
temp = voot > left? voot > left : root > right
if (temp = = Will)
temp = voot
roof = NUL
alse voot = temp
free (semp)
else
temp = minvalue Node (root -> right)
root > key - temp > key
root >right = delete Node (root->right, temp > ky)
if (root == NULL) votuan root
roof - sheight = 12 max (height (root-slift), height(root -> vight),
bolance = gel Bolance (root)
if (bolance > 1 & t gelfalance (root ->left) >= 0) return vight Rotate (root)
root -s left = leftRotate (root->left) 20)
return rightkotate (root)
else if (balance < -1 4& get Balance (1006 -> right) x=0)
retion self-Rotate (voot)
else if (balance c. 1 44 get Palance Wood Sright) >0)
roof > right = rightPotate(roof > right)
. vetran. left Rotate (root)
return root