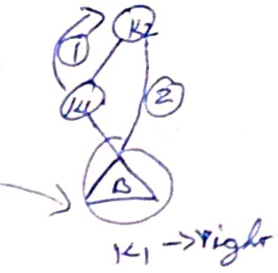
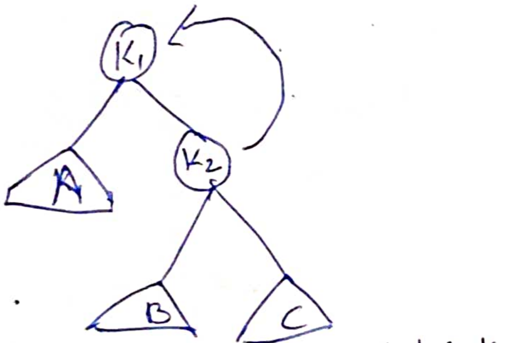
```
Singlerotationswithleft ( Struct avltree * K2)
{
    struct avltree *K1;
①   K1 = K2 -> left;
②   K2 -> left = K1 -> right;
    K1 -> right = K2;
   // return K1
    K2 -> height = Max ( height (K2 -> left), h(K2 -> right ))+1
    K1 -> height = max (height (K1 -> left), h(K1 -> right))+1
    return K1
}
```

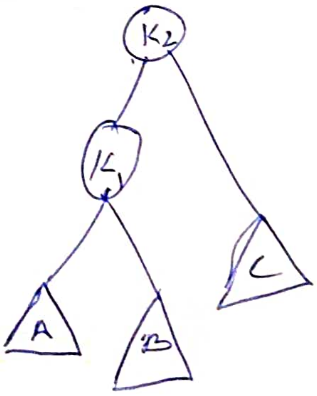


Case ② RS of RC of A

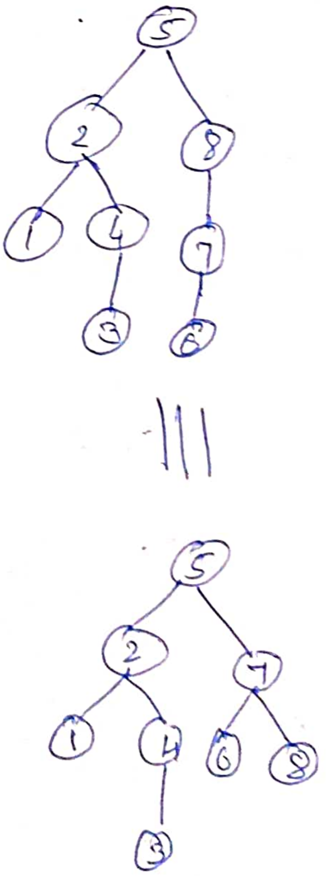


Now B is < K2 but > K1

So change to K1



Eg :



```
Singl rotation with Right ( Avl tree * K2)
{
        struct avltree *K1;
         K1 = K2 -> right;
         K2 -> right = K1 -> left;
         K1 -> left = K2;
```

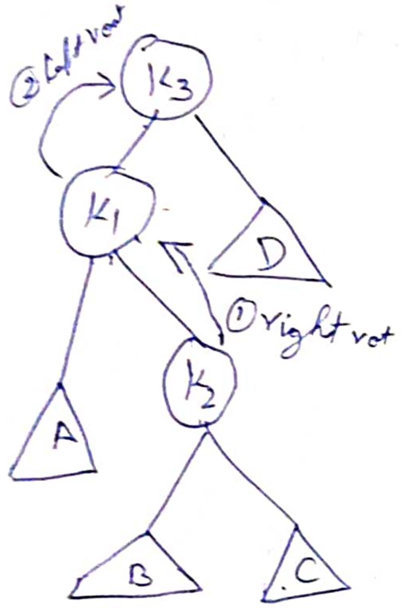
$K_2 \to height = max (h (K_2 \to left) ; h (K_2 \to Right)) + 1$

$K_1 \to height = max (h (K_1 - left) ; h (K_1 \to Right)) + 1$

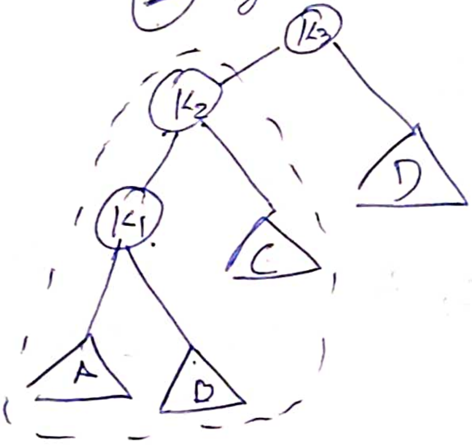
return K_1

}

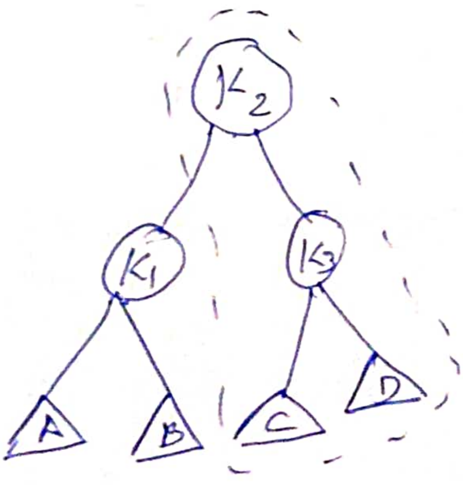
Case ③: LS of RC of A



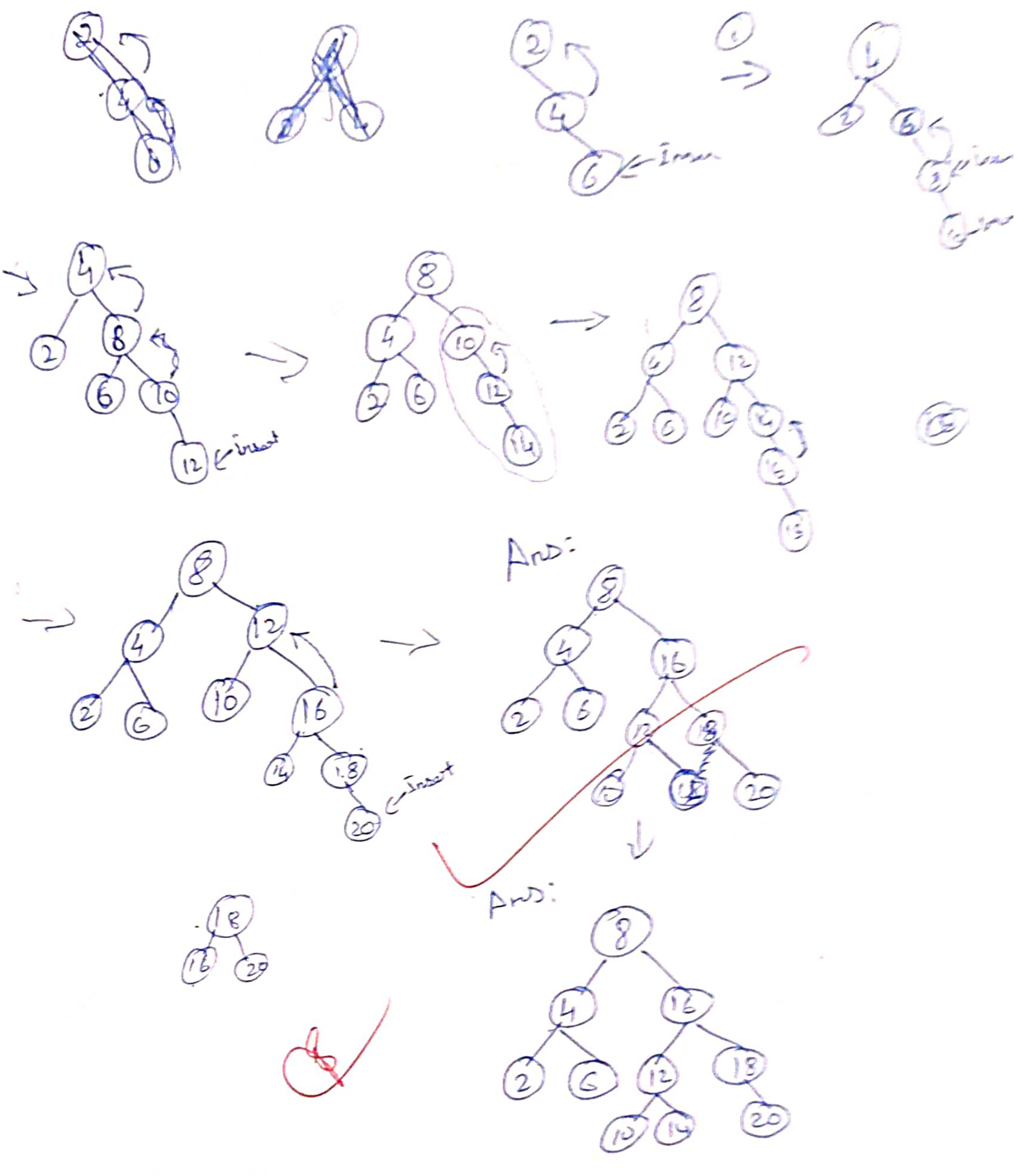
① right rotation



② Left rotation

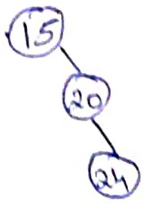


2, 4, 6, 8, 10, 12, 14, 16, 18, 20.

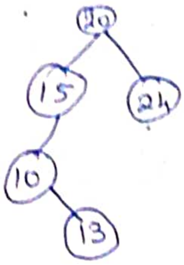


2) 15, 20, 24, 10, 13, 7, 30, 36, 25

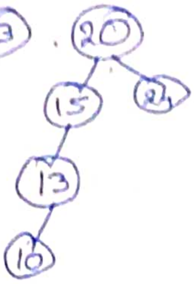
1



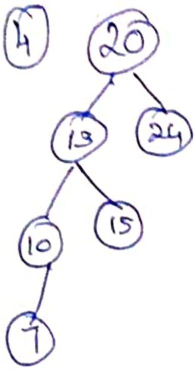
2



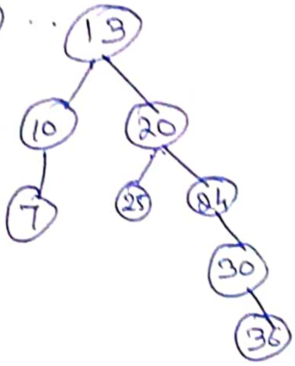
3



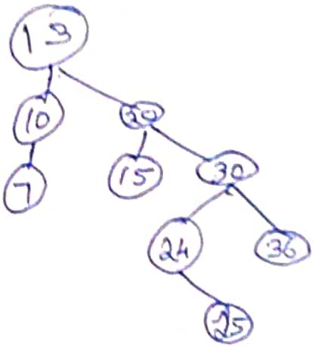
4



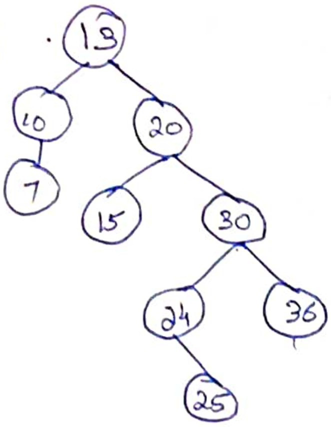
5



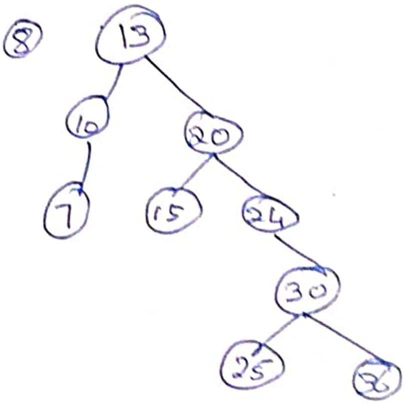
6



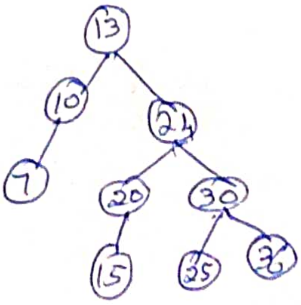
7



8



9

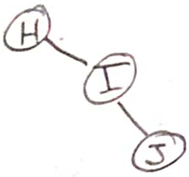


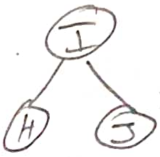
Karthikeyan - A

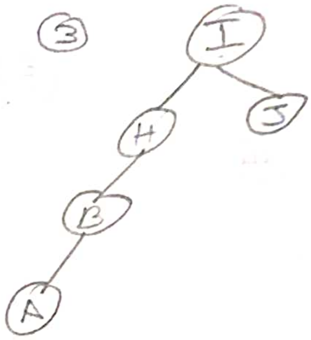
Date : 10/11/22

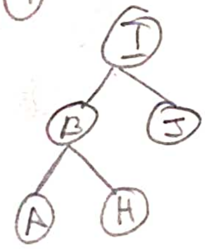
Tutorial - 2

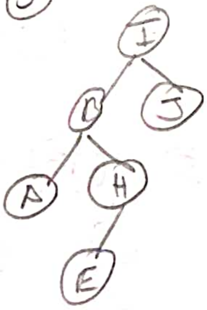
Insert the element H, I, J, B, A, E ; C, F, D, G, K, L

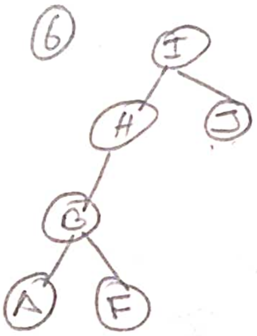
1


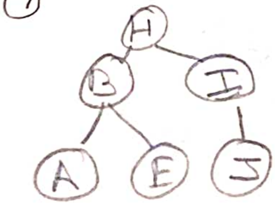
2


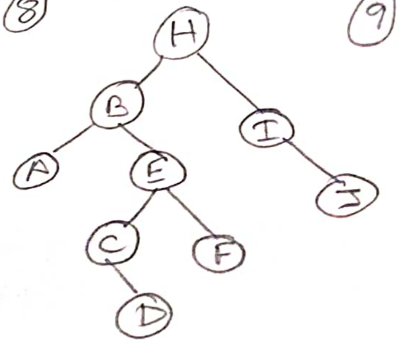
3


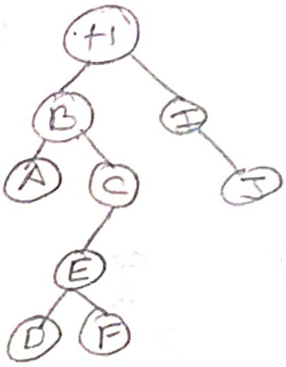
4


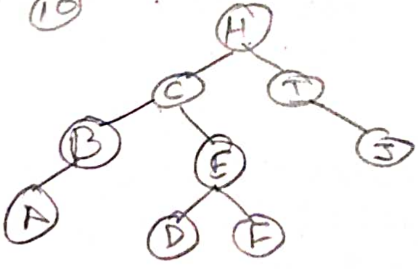
5


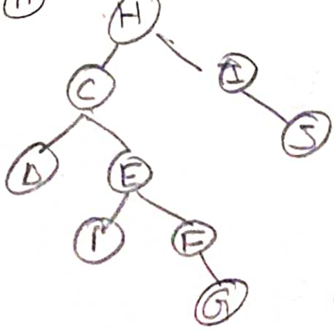
6


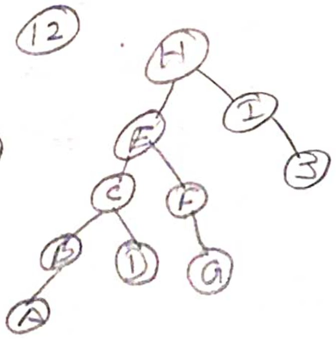
7


8


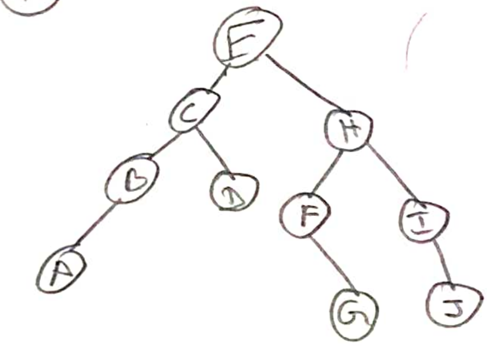
9


10


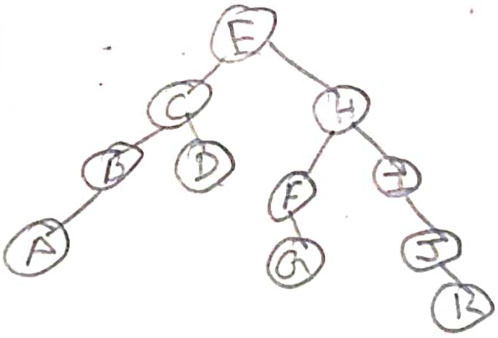
11


12


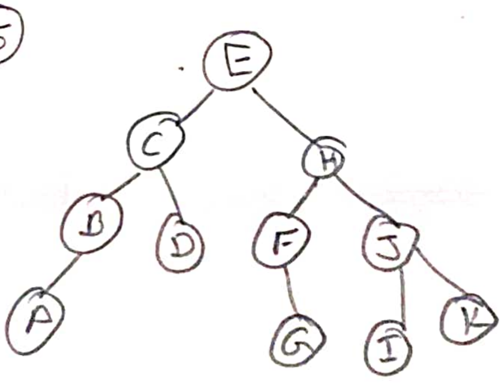
13



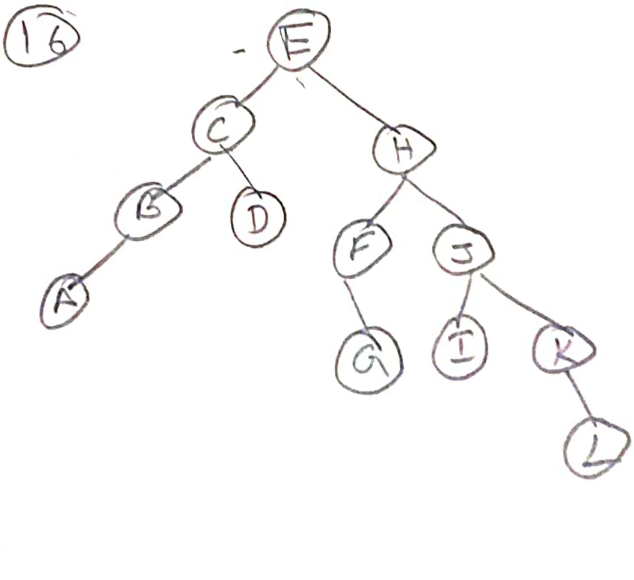
14



15

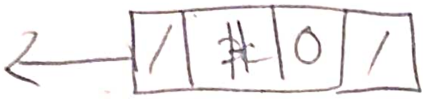


16

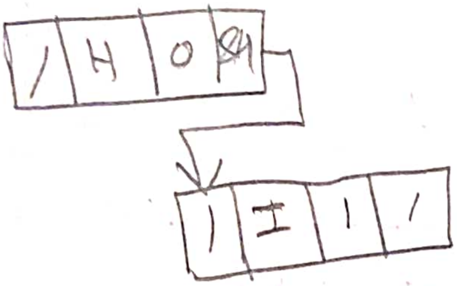


while inserting the elements H, I, J, B, A, E, C, F, D, G, K, L into AVL tree, trace the insertion algorithm for element insertion.

① if (T==NULL) ✓ x = 'H'

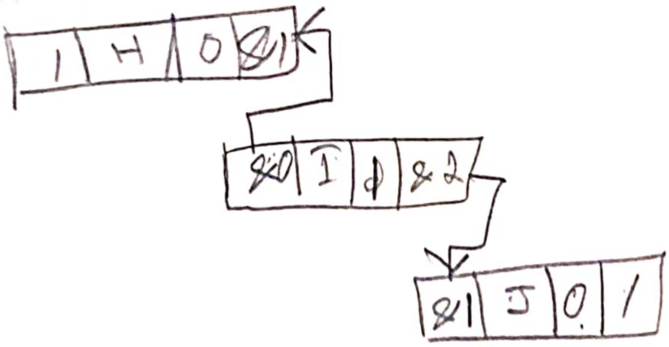


② x = I

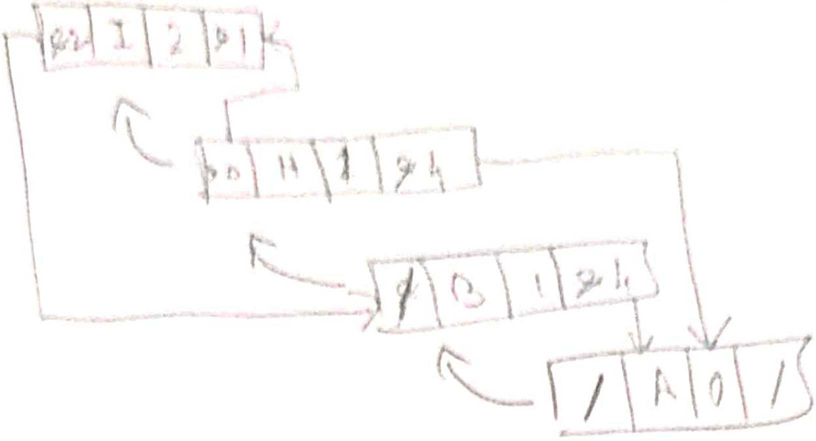


0≠2

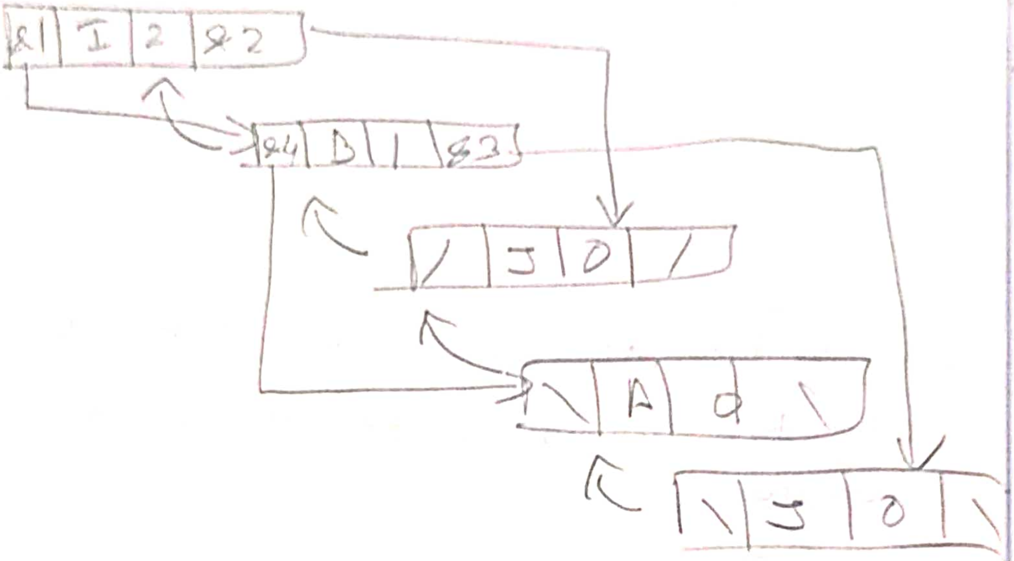
③



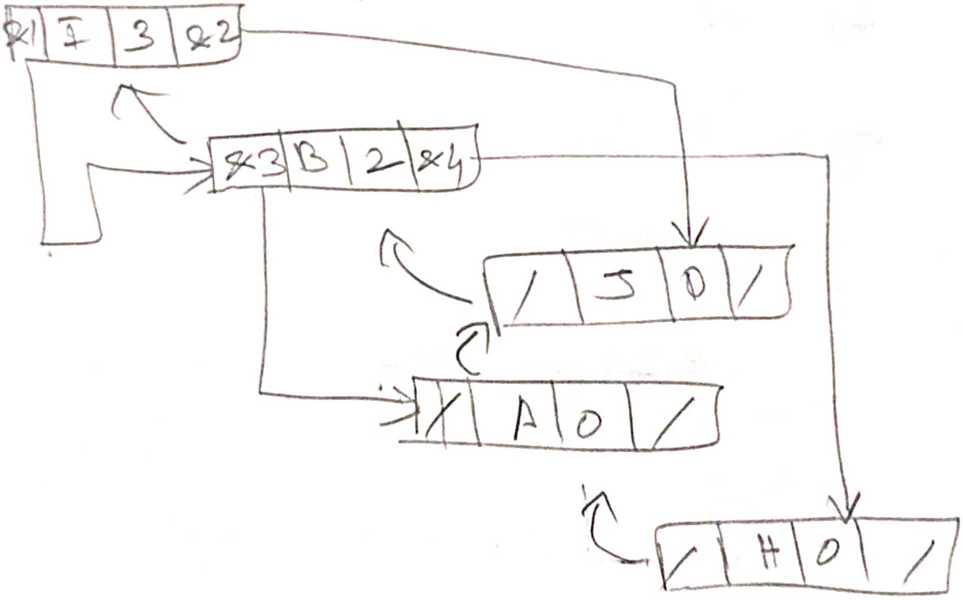
4



5



6


&1 | I | 3 | &2
&3 | B | 2 | &4
/ | J | 0 | /
/ | A | 0 | /
/ | H | 0 | /