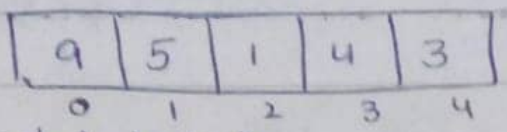
int data[5] = {9, 5, 1, 4, 3}

data →



insertionsort (data);

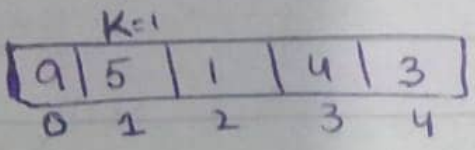
we call the insertionsort function

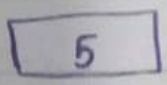
void insertionsort (int array[])

{

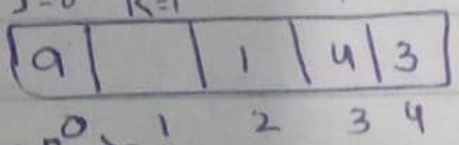
for (int K = 1; K < 5; K++)

int key = array[K] (1)





int j = K - 1 = 0 (1 - 1)

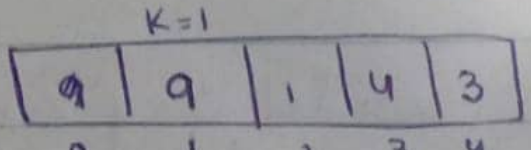


while (key < array[j] && j >= 0) (5 < 9, 0 >= 0)

{ array[j+1] = array[j] (0+1=1)

--j (-1)

→ condition true



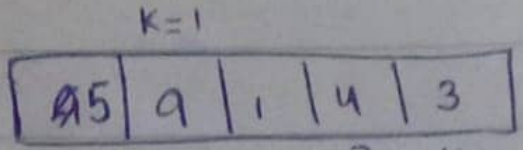
}

: again go back to while loop

while (key < array[j] && j >= 0) (5 < NULL, -1 >= 0) → false

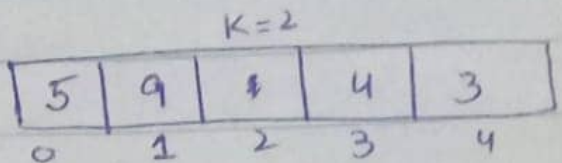
skip while loop

array[j+1] = key (0+1=1)

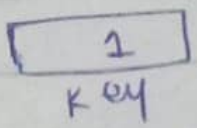


again repeat for loop.

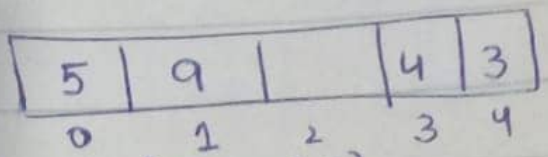
for (int K = 1; K < 5; K++) (2 < 5, 2)



int key = array[2]



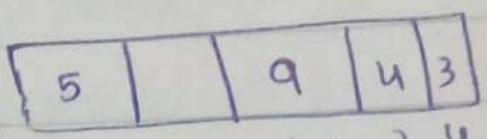
int j = 2 - 1 = 1



while (key < array(1) && j >= 0)

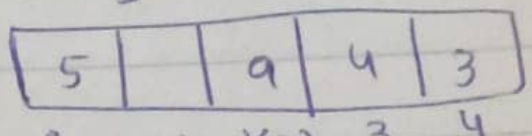
1 < 9

array [1+1] = array [1]



- -j ⇒ 0

}



again repeat while loop

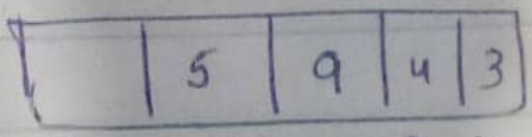
while (key < array [j] && j >= 0)

1 < 5 0

{

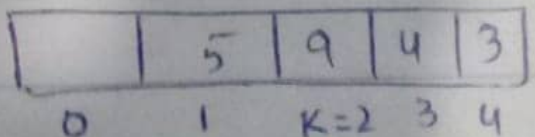
array [0+1] = array [j]

1 5 0



- -j => -1

}



again repeat while loop

while(key < array[j] && j>=0)

1 Null -1 -1>=0 → false

we skip the while loop

go to the outher loop

arr[j+1] = key

-1+1

1	5	9	4	3
0	1	K=2	3	4

j=-1

again repeat for loop

for (int K=1 ; K<5 ; K++)

3 <5 3

int key = array[K]

1	5	9	4	3
0	1	2	K=3	

3 -1

int j = K-1 = 2

4

key

1	5	9	4	3
0	1	2	K=3	4

j=2

4 9 2 j=2

while(key < array[j] && j>=0)

4

2+1 2

arr[j+1] = array[j]

--j ;

1

1	5	9	9	3
0	1	2	K=3	4

j=2

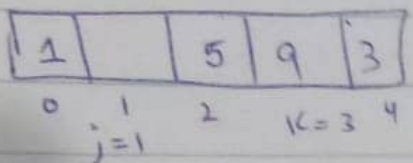
1	5		9	3
0	j=1	2	K=3	4

again repeat the while loop

while (key < array[j] && j>=0)

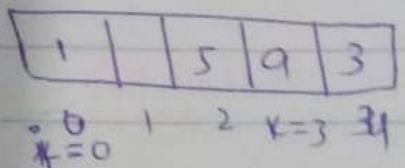
{

array[j+1] = array[j]



--j => 0

}



again repeat while condition

while (key < array[j] && j>=0)

→ false

skip while part.

array[j+1] = key