

Insertion sort

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

9	5	1	4	3
---	---	---	---	---

```
insertionsort(data);
```

we call the insertionsort function

```
void insertionsort(int array[])
```

```
{
```

```
    for(int k=1; k<5; k++)
```

```
        int key = array[k]
```

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

```
        int j = k-1 = 0
```

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

```
        {
```

```
            array[j+1] = array[j]
```

```
            --j
```

```
        }
```

5

key

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

∴ Again go back to while loop

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

Now the condition is false

so we skip the while loop.

```
array[j+1] = key
```

5	9	1	4	3
0	1	2	3	4

Again repeat for loop

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

int key = array[k]

int j = k-1

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

array[j+1] = array[j]

--j;

}

Again repeat while loop

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

array[j+1] = array[j]

--j;

}

j = -1

5	9	1	4	3
0	1	2	3	4

key

5	9		4	3
0	1	2	3	4

5		9	4	3
0	1	2	3	4

5		9	4	3
0	1	2	3	4

	5	9	4	3
0	1	2	3	4

	5	9	4	3
0	1	2	3	4

Again repeat while loop

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

Now skip while loop

arr[j+1] = key

1	5	9	4	3
0	1	2	3	4

Again repeat for loop

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

if key = array[k]

4
key

int j = k - 1

j=2

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

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

{

arr[j+1] = arr[j]

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

j--



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

Again repeat while loop

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

{
array[j+1] = array[j]

1		5	9	3
0	1	2	3	4

j-1 = 0

1		5	9	3
0	1	2	3	4

Again repeat while

condition:

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

Now we skip false while loop;

array[j+1] = key

1	4	5	9	3
0	1	2	3	4

Now this process repeat again and again until the array is sorted.

1	3	4	5	9
0	1	2	3	4