Selection Sort

```
int main()
{
    int a[10], i, j, temp;
for (i=0; i<10; i++)
{
  scanf ("%d", &a[i]);
}
for (i=0; i<10; i++)
{
for (j=0; j<10; j++)
{
    if (a[i] > a[j])
    {
        temp = a[i];
        a[i] = a[j];
        a[j] = temp;
    }
}
}
for (i=0; i<10; i++)
{
  printf ("%d", a[i]);
}
return 0;
```

1st pass

i=0

4 — i
5 — j

4 — i
5
3 — j Swap

3 — i
5
4
2 — j

2 — i
5
4
3
1 — j

1 —
5
4
3
2

Smallest element comes at 1st position in 1st pass

i=1

5 — i
4 — j >

4 — i
5
3 — j >

1
3 — i
5
4
2 — j

1
2
5
4
3

2nd smallest element

i=2

1
2
5 — i
4 — j >

1
2
4 — i
5
3 — j

1
2
3
5
4

—

3rd smallest element

i=3

1
2
3
5 — i
4 — j >

1
2
3
4
5

Sorted Array (Selection Sort)

- Simple and efficient algo
- Works by repeatedly selecting the smallest (or largest) element from the unsorted portion of the list and moving it to the sorted position of list.

Bubble sort

```
main()
{
int a[10], i, j, temp;
for(i=0; i<10; i++)
{
scanf("%d", &a[i]);
}
for(i=0; i<n; i++)
{
for(j=i+1; j<n-i-1; j++)
{
if(a[j]>a[j+1])
{
temp= a[j];
a[j]= a[j+1];
a[j+1]= temp;
}
}
}
for(i=0; i<10; i++)
{
printf("%d", a[i]);
}
```

Bubble sort.

i = 0

1st pass

4	— j	<	4		4		4		4		4
5	— j+1		5	— j >	1		1		1		1
1			1	— j+1	5	— j >	3		3		3
3			3		3	— j+1	5	— j >	2		2
2			2		2		2	— j+1	5		5

Swap

Largest element will come at last position

i = 1

4	— j		1		1		1		1		1
1	— j+1	>	4	— j	>	3		3		3	
3			3	— j+1		4	— j	>	2		2
2			2			2	— j+1		4		4
5			5			5			5		5

i = 2

1	— j		1		1		1		1		1
3	— j+1	<	3	— j		2		2		2	
2			2	— j+1		3		3		3	
4			4			4		4		4	
5			5			5		5		5	

i = 3

1	— j		1		1		1		1		1
2	— j+1		2		2		2		2		2
3			3		3		3		3		3
4			4		4		4		4		4
5			5		5		5		5		5

→ Sorted Array

(Bubble sort)

Concept: Works by repeatedly swapping the adjacent elements if they are in wrong order.

Insertion Sort

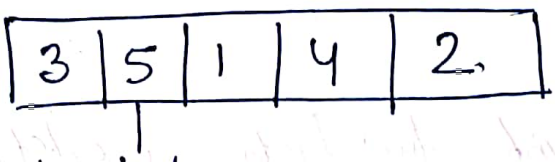
→ Idea is to divide the list into two parts: a sorted part and unsorted part

→ Initially sorted part contains only 1st element of list. while rest of list is in unsorted part.

→ The algo then iterates through each element in the unsorted part, picking one at a time and insert into its correct position in sorted part.

```
main()
{
  int a[10], temp, j, i;
  for(i=1; i<n; i++)
  {
    temp=a[i];
    j=i-1;
    while(j>=0 && a[j]>temp)
    {
      a[j+1]=a[j];
      j=j-1;
    }
    a[j+1]=temp;
  }
```

(1)



(6)

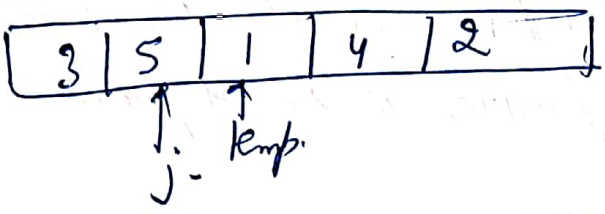
a[j] > temp, False

So no change

a[j+1] = temp; 5 will remain in same position

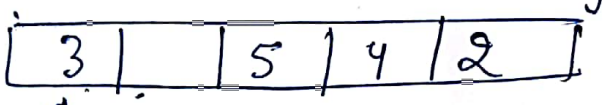
---

(2)



a[j] > temp, True

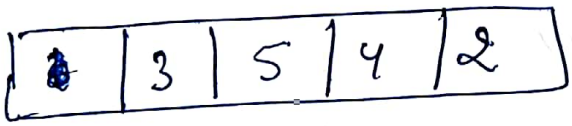
So a[j+1] = a[j]:



a[j] > temp

3 > 1

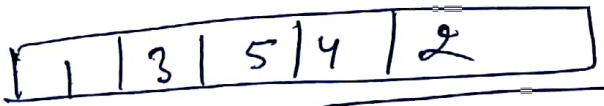
3 will move right



j-1 = False

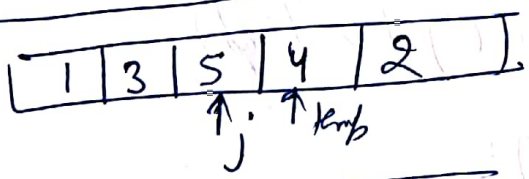
Loop stop

a[j+1] = temp



---

(3)



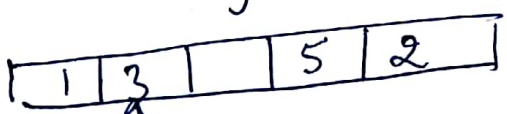
5 > 4

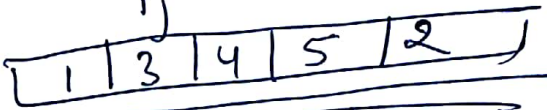
5 will move right

a[j] < temp

3 < 4 = Loop stop

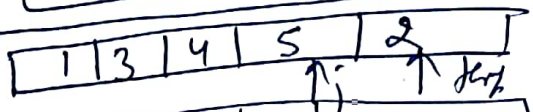
a[j+1] = temp





---

(4)



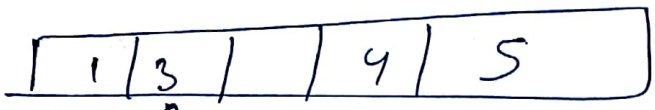
a[j] > temp 5 > 2

4 > 2

4 will move right.

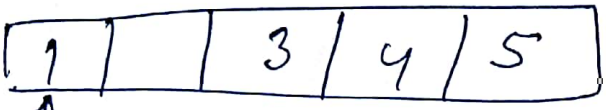
a[j]+1 temp

temp
2



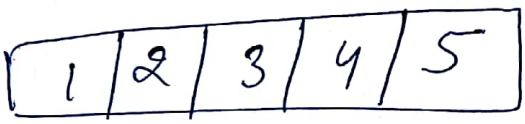
7 3 > 2

3 will move right



temp
2

Loop all stop.



and a[j+1] = temp;

Sorted Array.