Arrays

Insertion

Program to Insert an element in an Array

Given an array **arr** of size **n**,

Scenario 1: **arr** is unsorted

Scenario 2: **arr** is sorted

Program to Insert an element in an Array

Problem: Given an array **arr** of size **n**, For **scenario 1**:

Case 1: insert an element **x** in the array **arr** at the end of the array **arr**

Case 2: insert an element **x** in the array **arr** at specific position **pos.**

Case 3: insert an element **x** in the array **arr** at first position (Beginning of the Array **arr**).

Program to Insert an element in an Unsorted Array

- Case 1: insert an element **x** in the array **arr** at the end of the array **arr**

Program to Insert an element in an Array

- Case 1: insert an element **x** in the array **arr** at the end of the array **arr**

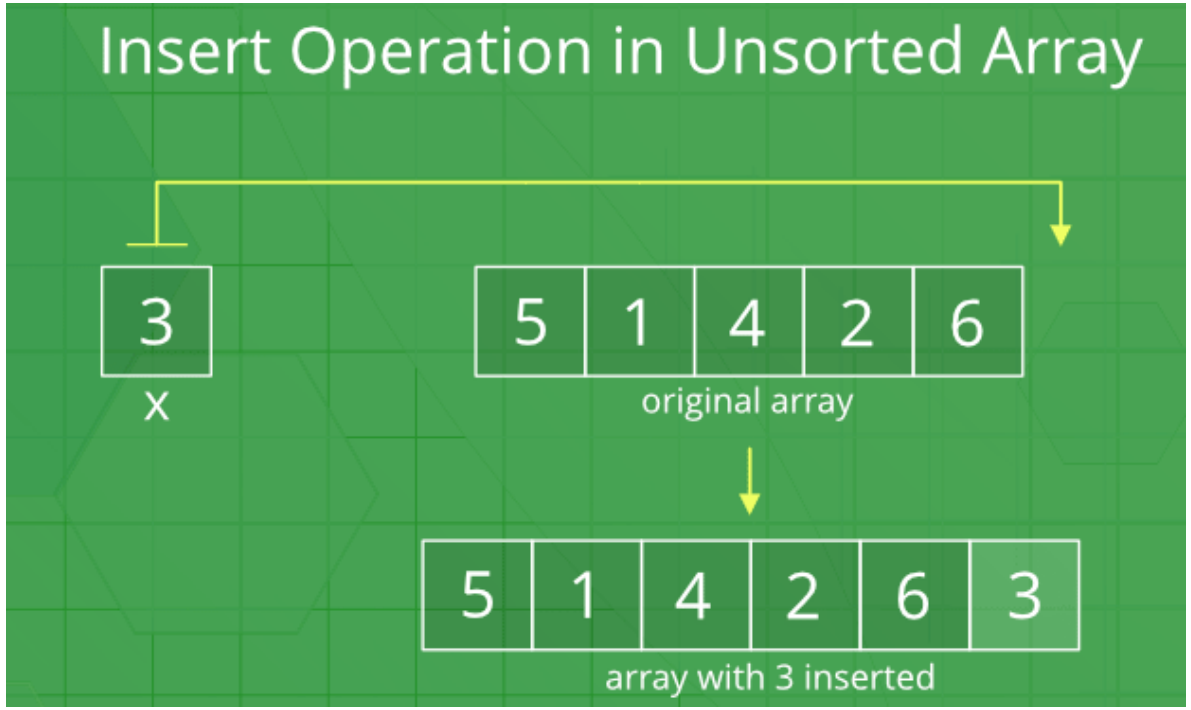
Insert at the end

In an unsorted array, the insert operation is faster as compared to a sorted array because we don’t have to care about the position at which the element is to be placed.

Program to Insert an element in an Array

- Case 1: insert an element **x** in the array **arr** at the end of the array **arr**

Insert at the end



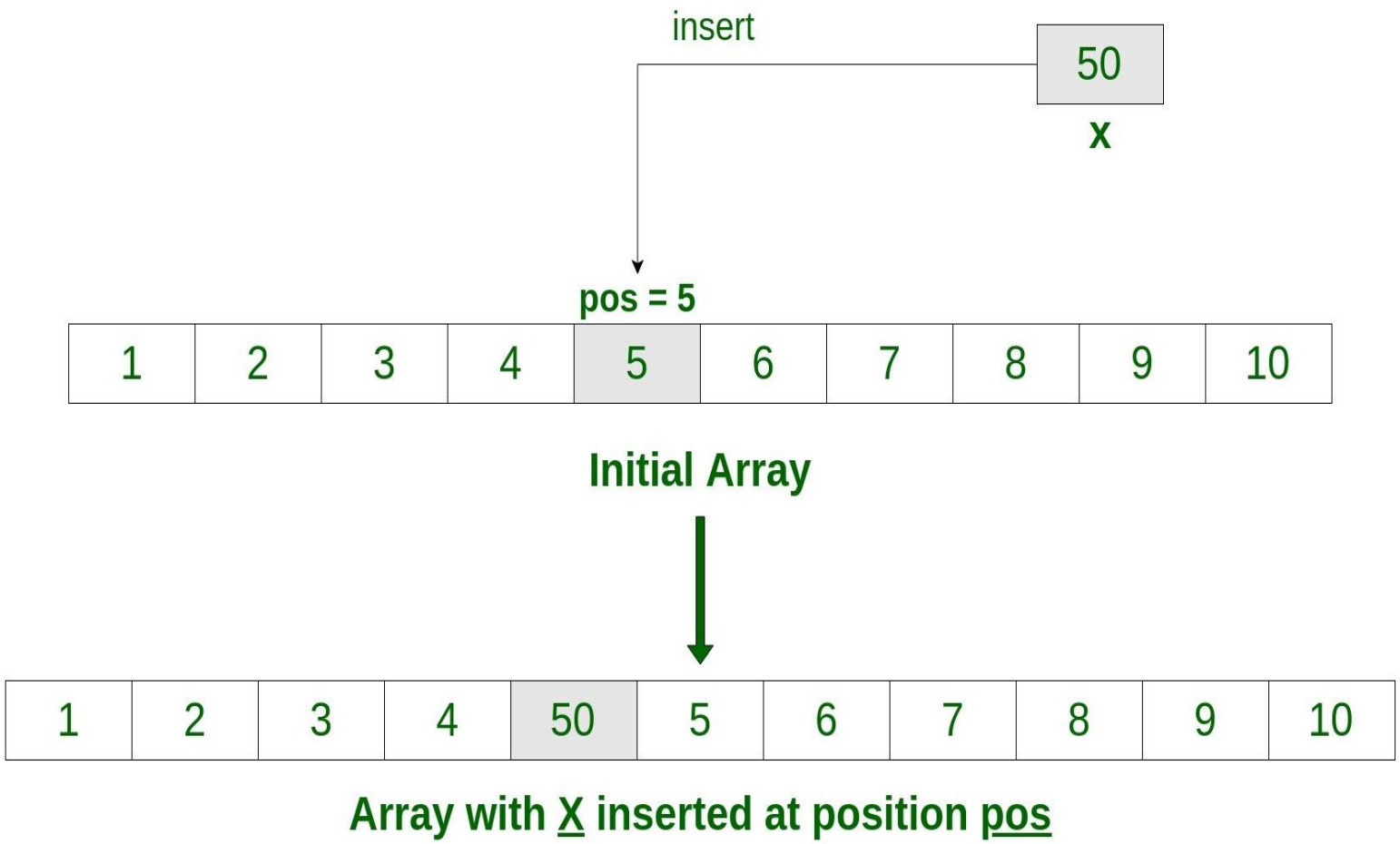
Prog nd in an Unsc

```
// C program to implement insert
// operation in an unsorted array.
#include<stdio.h>
int main()
{
    int arr[20] = {12, 16, 20, 40, 50, 70};
    int totalele = 6;
    int i, key;

 printf("\n Enter the number to be Inserted at the end of
the array: ");
  scanf(“%d“, &key);

  printf("\n Before Insertion: ");
    for (i = 0; i < totalele; i++)
        printf("%d  ", arr[i]);
  // Inserting key
    arr[totalele]=key;

   printf("\n After Insertion: ");
    for (i = 0; i <= totalele; i++)
        printf("%d  ",arr[i]);

    return 0;
```

Program to Insert an element in an Array

Case 2: Given an array **arr** of size **n**, insert an element **x** in the array **arr** at a specific position **pos**.

Insert an element at a specific position in an Array



Approach:

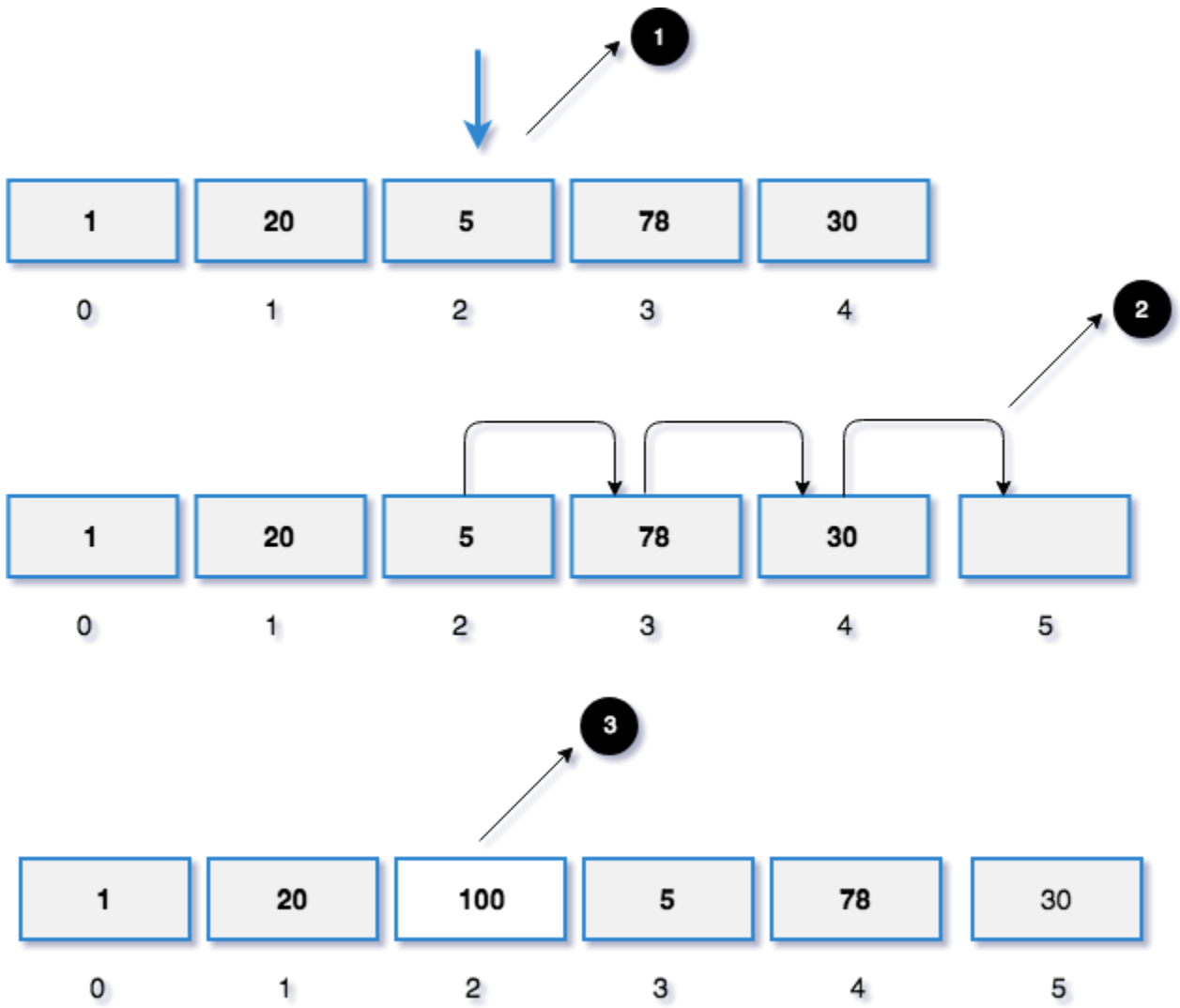
- Here’s how to do it.

1. First get the element to be inserted, say x
2. Then get the position at which this element is to be inserted, say **pos**
3. Then shift the array elements from this position to one position forward, and do this for all the other elements next to pos.
4. Insert the element x now at the position pos, as this is now empty.

Approach:

- Consider an array a[10] having three elements in it initially and a[0] = 1, a[1] = 2 and a[2] = 3 and you want to insert a number 45 at location 1 i.e. a[0] = 45,
- so we have to move elements one step below so after insertion a[1] = 1 which was a[0] initially, and a[2] = 2 and a[3] = 3.
- Array insertion does not mean increasing its size, i.e., array will not contain 11 elements.

For example: Insert 100 at position 3



A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11			

Pos =3
Value = 99

Pos =3 means loc is a[2]

Step 1: A[7] =A[6]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11	**11**		

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11			

Pos =3
Value = 99

Pos =3 means loc is a[2]

Step 1: A[7] =A[6]
Step 2: A[6] =a[5]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	**10**	11		

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11			

Pos =3
Value = 99

Pos =3 means loc is a[2]

Step 1: A[7] =A[6]
Step 2: A[6] =A[5]
Step 3: A[5] = A[4]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	**9**	10	11		

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11			

Pos =3
Value = 99

Pos =3 means loc is a[2]

Step 1: A[7] =A[6]
Step 2: A[6] =A[5]
Step 3: A[5] = A[4]
Step 4: A[4] = A[3]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	**8**	9	10	11		

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11			

Pos =3
Value = 99

Pos =3 means loc is a[2]

Step 1: A[7] =A[6]
Step 2: A[6] =A[5]
Step 3: A[5] = A[4]
Step 4: A[4] = A[3]
Step 5: A[3] = A[2]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	**7**	8	9	10	11		

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11			

Pos =3
Value = 99

Pos =3 means loc is a[2]

Step 1: A[7] =A[6]
Step 2: A[6] =A[5]
Step 3: A[5] = A[4]
Step 4: A[4] = A[3]
Step 5: A[3] = A[2]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	**99**	7	8	9	10	11		

A[2]= 99 where insertion will be done

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	7	8	9	10	11			

Pos =3
Value = 99

Pos =3 means loc is a[2]

Step 1: A[7] =A[6]
Step 2: A[6] =A[5]
Step 3: A[5] = A[4]
Step 4: A[4] = A[3]
Step 5: A[3] = A[2]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	6	**99**	7	8	9	10	11		

A[2]= 99 where insertion will be done

Points to Ponder:

1. If Array had n elements then last element is at n-1 position
2. Steps 1 to 5 can be written using the for loop
3. The loop will be initialized to n-1. say c=n-1
4. The loop condition would be executed till c >= pos-1
5. After the processing of for loop c would be decremented by 1
6. What step we are doing here
7. After exiting the for loop we are at correct position to make insertion

```c
#include <stdio.h>
 int main()
{   int array[100], pos, c, n, value;
   printf("Enter number of elements in array\n");
   scanf("%d", &n);
   printf("Enter %d elements\n", n);

   for (c = 0; c < n; c++)
      scanf("%d", &array[c]);

   printf("Enter the location where you wish to insert an element\n");
   scanf("%d", &pos);

   printf("Enter the value to insert\n");
   scanf("%d", &value);

   for (c = n - 1; c >= pos - 1; c--)
     { array[c+1] = array[c]; }
   array[pos-1] = value;

   printf("Resultant array is\n");

   for (c = 0; c <= n; c++)
      printf("%d\n", array[c]);
   return 0;  }
```

```
Enter number of elements in array
5
Enter 5 elements
2
5
4
3
8
Enter the location where you wish to insert an element
4
Enter the value to insert
10
Resultant array is
2
5
4
10
3
8
```

Program to Insert an element in an Array

- **Case 3:** insert an element **x** in the array **arr** at the beginning of the array **arr**

Insert at the beginning

For example consider an array n[10] having four elements:

n[0] = 1, n[1] = 2, n[2] = 3 and n[3] = 4

And suppose you want to insert a new value 60 at first position of array. i.e. n[0] = 60, so we have to move elements one step below so after insertion

n[1] = 1 which was n[0] initially, n[2] = 2, n[3] = 3 and n[4] = 4.

```c
#include <stdio.h>
#define MAX 5
void main() {
int array[MAX] = {27, 23, 14, 25};
int N = 4; // number of elements in array
int i = 0; // loop variable
int value = 10; // new data element to be stored in array
// print array before insertion
 printf("Printing array before insertion -\n");
for(i = 0; i < N; i++)
 { printf("array[%d] = %d \n", i, array[i]); }
// now shift rest of the elements downwards
for(i = N-1; i >= 0; i--)
{ array[i+1] = array[i]; }
// add new element at first position
array[0] = value;
// increase N to reflect number of elements
N++;
// print to confirm
printf("Printing array after insertion -\n");
 for(i = 0; i < N; i++)
{ printf("array[%d] = %d\n", i, array[i]);
}
}
```

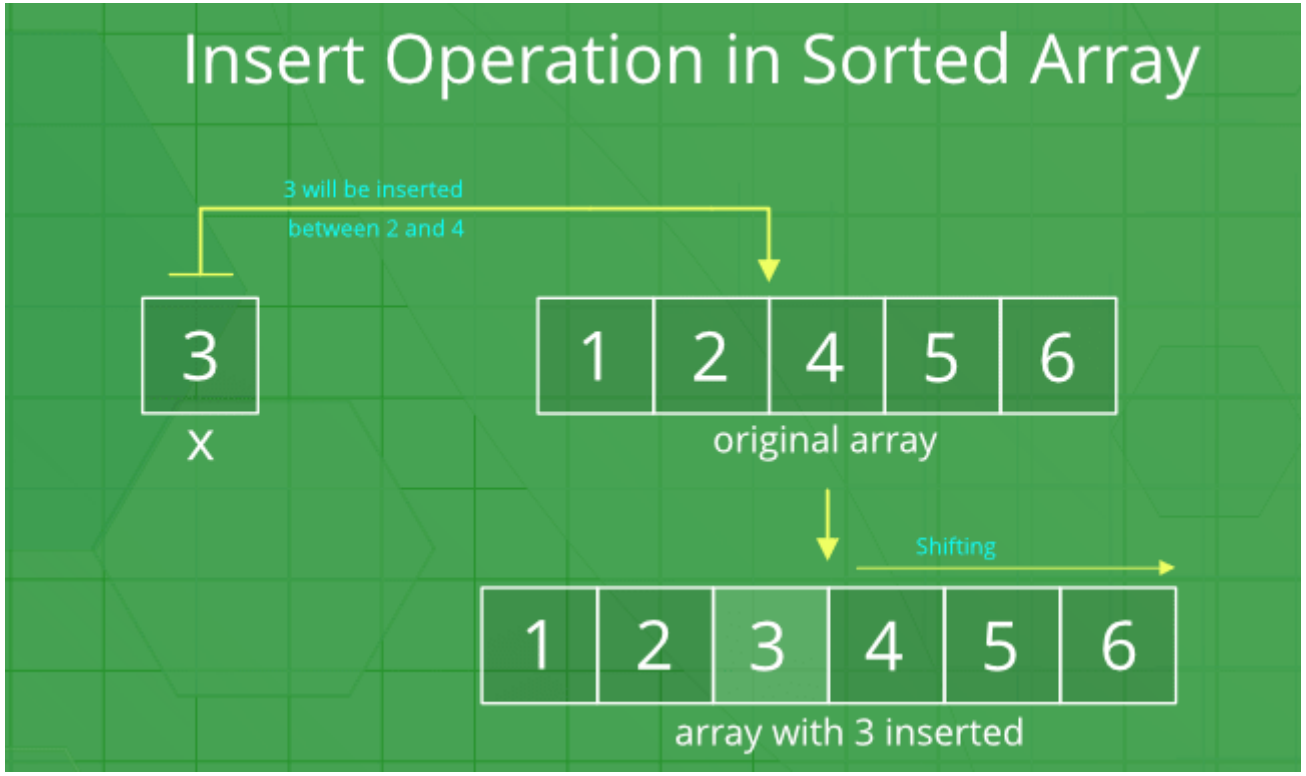
DIY

- Write a program to:
  - Insert a value Before the Given Index of an Array
  - Insert a value After the Given Index of an Array

Program to Insert an element in an Array

scenario 2: **arr** is sorted

Problem: Given a sorted array **arr** of size **n**, Insert a new element at the right position such that the array is sorted even after the insertion.



Program to Insert an element in an Array [Sorted List]

Write a program in C to insert New value in the array (sorted list).

In a sorted array, Firstly, a search operation is performed for the possible position of the given element and then insert operation is performed followed by shifting the elements.

Program to Insert an element in an Array [Sorted List]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	7	9	10	11			

Value to be inserted is 2 .
Position of insertion

Step 1: 2 <11
A[7]=A[6]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	7	9	10	11	11		

Program to Insert an element in an Array [Sorted List]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	7	9	10	11			

Value to be inserted is 2 .
Position of insertion

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	7	9	10	10	11		

Step 1: 2 <11
A[7]=A[6]
Step 2: 2< 10
A[6] =A[5]

Program to Insert an element in an Array [Sorted List]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	7	9	10	11			

Value to be inserted is 2 .
Position of insertion

Step 1: 2 <11, A[7]=A[6]
Step 2: 2< 10, A[6] =A[5]
Step 3: 2<9, A[5]=A[4]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	7	9	9	10	11		

Program to Insert an element in an Array [Sorted List]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	7	9	10	11			

Value to be inserted is 2 .
Position of insertion

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	7	7	9	10	11		

Step 1: 2 <11, A[7]=A[6]
Step 2: 2< 10, A[6] =A[5]
Step 3: 2<9, A[5]=A[4]
Step 4: 2<7, A[4] =A[3]

Program to Insert an element in an Array [Sorted List]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	7	9	10	11			

Value to be inserted is 2 .
Position of insertion

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	5	7	9	10	11		

Step 1: 2 <11, A[7]=A[6]
Step 2: 2< 10, A[6] =A[5]
Step 3: 2<9, A[5]=A[4]
Step 4: 2<7, A[4] =A[3]
Step 5: 2< 5, A[3]=A[2]

Program to Insert an element in an Array [Sorted List]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	7	9	10	11			

Value to be inserted is 2 .

Position of insertion

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	3	5	7	9	10	11		

Step 1: 2 <11, A[7]=A[6]
Step 2: 2< 10, A[6] =A[5]
Step 3: 2<9, A[5]=A[4]
Step 4: 2<7, A[4] =A[3]
Step 5: 2< 5, A[3]=A[2]
Step 6: 2<3, A[2]=A[1]

Program to Insert an element in an Array [Sorted List]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	7	9	10	11			

Value to be inserted is 2 .

Position of insertion

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	3	5	7	9	10	11		

Step 1: 2 <11, A[7]=A[6]
Step 2: 2< 10, A[6] =A[5]
Step 3: 2<9, A[5]=A[4]
Step 4: 2<7, A[4] =A[3]
Step 5: 2< 5, A[3]=A[2]
Step 6: 2<3, A[2]=A[1]
Step 7: 2>1, Exit the loop

Program to Insert an element in an Array [Sorted List]

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	3	5	7	9	10	11			

Value to be inserted is 2 .
Position of insertion ✖

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
1	2	3	5	7	9	10	11		

Step 1: 2 <11, A[7]=A[6]
Step 2: 2< 10, A[6] =A[5]
Step 3: 2<9, A[5]=A[4]
Step 4: 2<7, A[4] =A[3]
Step 5: 2< 5, A[3]=A[2]
Step 6: 2<3, A[2]=A[1]
Step 7: 2>1, Exit the loop

A[1]= 2 where insertion will be done

Points to Ponder:

1. If Array had n elements then last element is at n-1 position
2. Steps 1 to 5 can be written using the for loop
3. The loop will be initialized to n-1. say c=n-1
4. The loop condition would be executed till value< A[c] && c>=0
5. After the processing of for loop c would be decremented by 1
6. When value becomes greater then value at A[c] exit the loop

```c
#include<stdio.h>
void main( )
{
	int a[20],n,item,i;

	printf("Enter the size of the array");
	scanf("%d",&n);

	printf("Enter elements of the array in the sorted order");
	for(i=0; i<n; i++)
	{
		scanf("%d", &a[i]);
	}

	printf("\nEnter ITEM to be inserted : ");
	scanf("%d", &item);

	i = n-1;
	while(item<a[i] && i>=0)
	{
		a[i+1] = a[i];
		i--;
	}
	a[i+1] = item;
	n++;

	printf("\n\nAfter insertion array is :\n");
	for(i=0; i<n; i++)
	{
		printf("\n%d", a[i]);
	}
	getch();
}
```

Note that the array elements to be entered in the array should be in the **ascending order**.

As the program to insert element is made according to it.

After that enter the item to be inserted and the while loop will check where the item should be inserted.

It will insert the item according to the sorted ascending order.