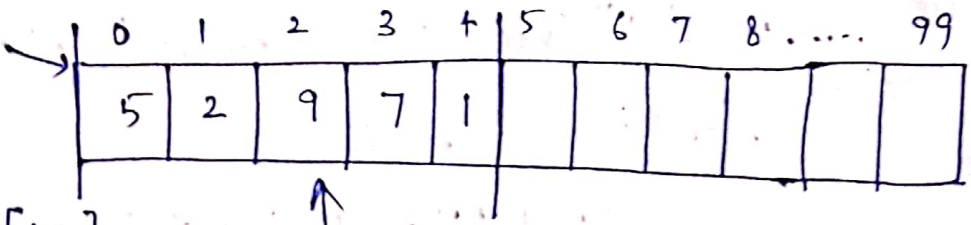
Array operations :-

1.) Searching of an element in an Array.
2.) Sorting elements of an Array (Ascending or descending)
3.) Insertion of an element in an Array.
4.) Deletion " " " from " " .

(I.) Insertion of an element in An Array :-

Ex:- A



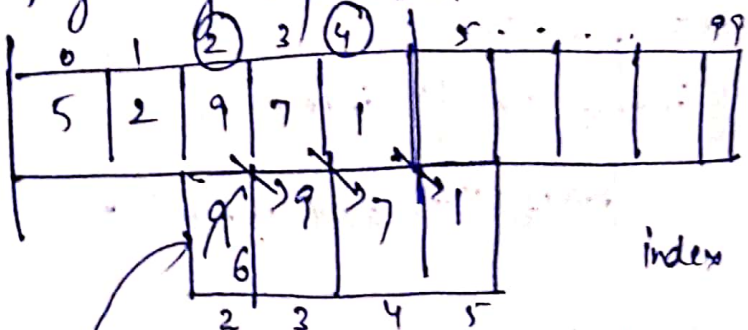
int A[100].

n = 5 (elements)

ele = 6

position = 3 ⟶ index would be 3-1 = 2.

Process:- We need to insert element 6 at index 2 & we will shift element 9, 7 & 1 to the right by 1 position



Shifting from index 4 to 2

index 4 → 5
3 → 4
2 → 3

then copy element at index 2

ele = 6

Now size = n+1 = 5+1 = 6

final Array would be

0	1	2	3	4	5
5	2	6	9	7	1

Code:-

```
#include <stdio.h>
Void main()
{
    int A[100], n, i, ele, pos, index;
    printf("Enter size");
    scanf("%d", &n);
    printf("Enter %d elements", n);
    for(i=0; i<n; i++)
        scanf("%d", &A[i]);
    printf("Enter element you want to insert");
    scanf("%d", &ele);
    printf("Enter position where you want to insert");
    scanf("%d", &pos);
    index = pos-1;
    if( index < 0 || index > n)
        printf("Invalid index");
    else
    {   for( i= n-1; i >= index; i--)
            A[i+1] = A[i];
        A[index] = ele;
```

Input size, Array elements, element & position

→ maximum size

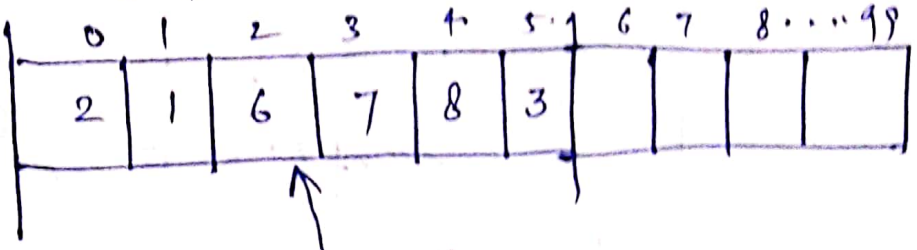
→ shifting from left to right

```
        n= n+1;
        printf(" Array after Insertion of element \n");
        for(i=0; i<n; i++)
            printf("%d\t", A[i]);
    }
}
```

(II.) Deletion of an element from an Array :-

(i) Deletion by position :-

EX:-



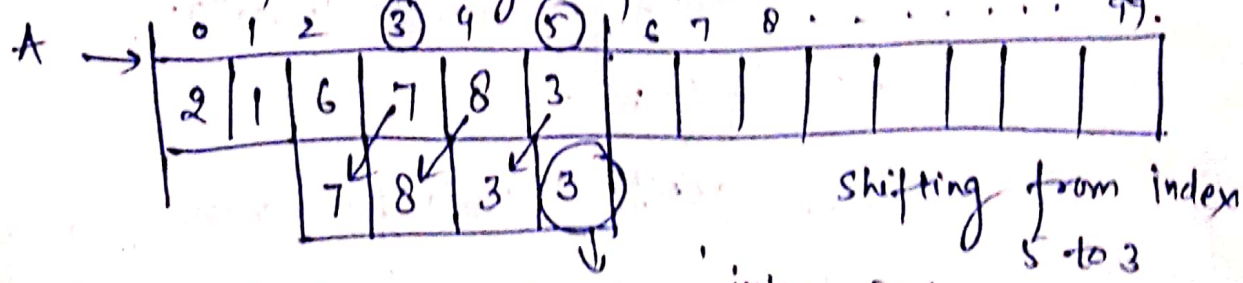
int A[100]

n = 6 (no. of elements)

~~ele~~ position = 3 → index would be 3-1 = 2

need to be deleted.

Process:- After deleting element at index 2 which is 6, we will shift all elements after 6 i.e. 7, 8 & 3 to left side by 1 position.



New size = n-1 = 6-1 = 5

would be garbage now

index 5 → 4
4 → 3
3 → 2

final Array would be

0	1	2	3	4
2	1	7	8	3

code :-

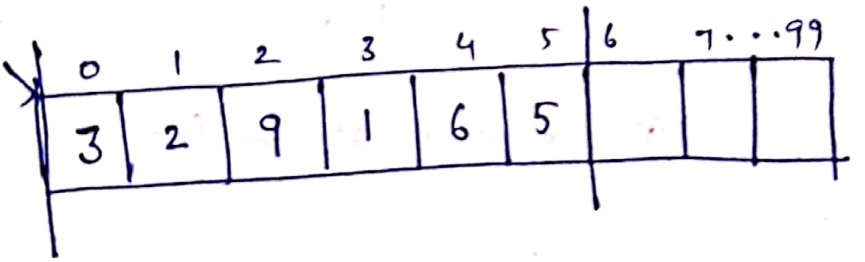
```
#include <stdio.h>
Void main()
{
    int A[100], n, i, pos, index, ele;
    printf(" Enter size");
    scanf("%d", &n);
    printf("Enter %d elements", n);
    for(i=0; i<n; i++)
        scanf("%d", &A[i]);
    printf(" Enter at what position element you
            want to delete ");
    scanf("%d", &pos);
    index = pos-1;
    if (index < 0 || index > n-1)
        printf(" Invalid index");
    else if( n==0)
        printf(" Underflow, nothing to delete)
    else
    {
```

Input

```
ele = A[index];
for( i = pos; i <= n-1; i++)
    A[i-1] = A[i];        ⟶ Shifting from Right to left
n = n-1;
printf("Element deleted = %d", ele);
printf("Array after deletion of element\n");
for(i=0; i<n; i++)
    printf("%d\t", A[i]);
}
}
```

(III.) Searching of an element in an Array:- (Linear Search)

Ex:- A



Searching element 1 by 1 from starting to end.

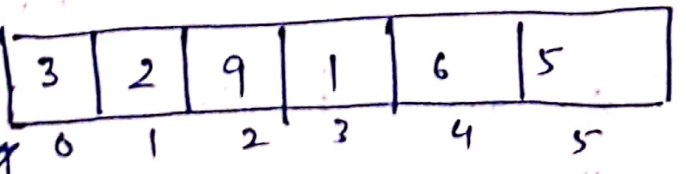
int A[100]

n = 6 (No. of elements)

ele to search (ele) = 1

Process:-

Step(1)

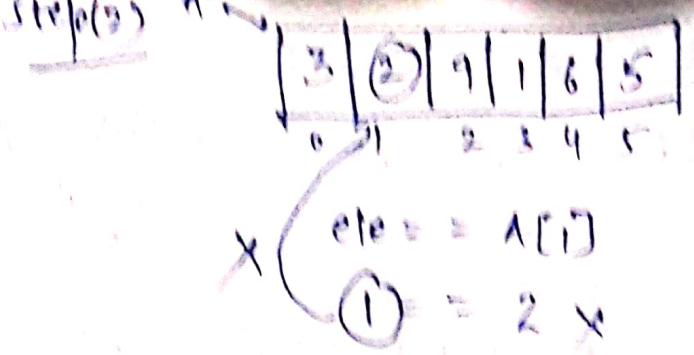


ele = = A[0]

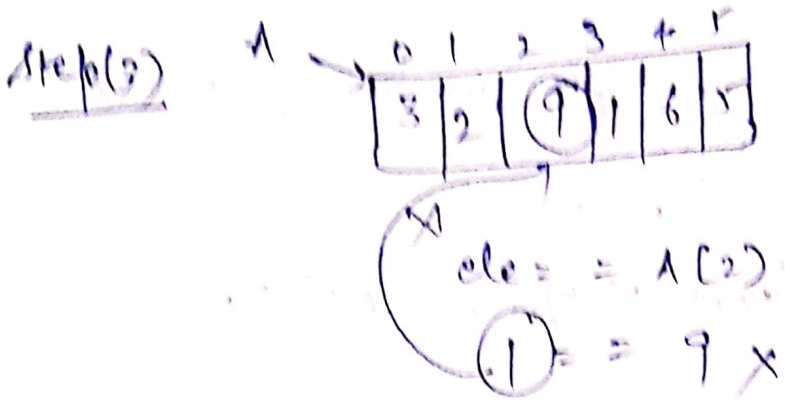
1 = = 3 X

X

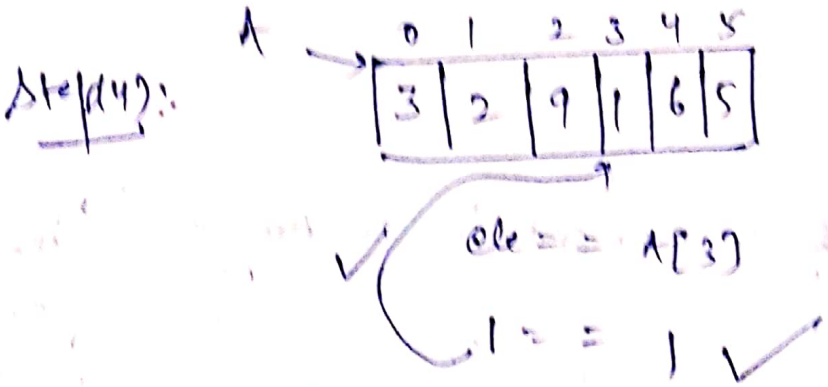
Step(3)



Step(3)



Step(4):



Yes, found at index 3, position 3+1=4

Stop

① Code:

```
#include <stdio.h>
void main()
{
    int A[100], n, i, ele
    printf("Enter size");
    scanf("%d", &n);
```

Assuming all elements are distinct. (different)

Input

```
printf("Enter %d elements", n);
for (i=0; i<n; i++)
    scanf("%d", &A[i]);
printf("Element element you want to search? ");
scanf("%d", &ele);
```

Searching Linearly

```
for (i=0; i<n; i++)
{
    if (A[i] == ele)
    {
        printf("%d found at position %d ",
                ele, i+1);
        break;
    }
}
```

if not found

```
if (i == n)
    printf("Element Not found");
}
```

Code (2):- Assuming elements are Repeating

```
f=0;
for (i=0; i<n; i++)
{
    if (A[i] == ele)
    {
        f=1;
        printf("%d found at position %d", ele, i+1);
    }
}
```

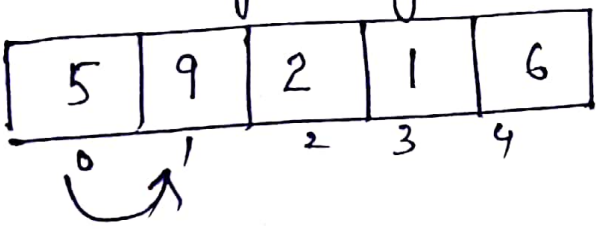
```
if( f==0)
    printf(" Element Not found");
```

(IV) **Sorting :- (Bubble sort)**

To arrange the data either in ascending or decending order.

Bubble sort:- In this sorting we will compare 2 elements at a time (adjacent to each other) & shift larger element to left & smaller to right likewise for whole array (Bubbling largest element at last position).

Ex:-



→ Actually have to (Bubble) sort 4 elements out of 5, rest 1 element will automatically be at its position.

Pass I

(i) 5 9 2 1 6

(ii) 5 2 9 1 6

(iii) 5 2 1 9 6

(iv) 5 2 1 6 9 → Bubbled/sort at its position.

4 comparisons for 5 elements Array

Pass II

(i) 2 5 1 6 9

(ii) 2 1 5 6 9

(iii) 2 1 5 6 9

3 comparisions

↳ Bubbled / sorted at its position

Pass III

(i) 1 2 5 6 9

(ii) 1 2 5 6 9

2 comparisions

↳ Bubbled / sorted

Pass IV

(i) 1 2 5 6 9 → 1 comparison

↳ Sorted

↓

No need to sort, already at its place.

final array

1	2	5	6	9

(Ascending order)

→ As we can see we have to sort only n-1 No. of elements out of n elements, 1 (last) will automatically gets sorted.

→ So we have to run n-1 passes to sort / Bubble them.

→ Each pass takes some No. of comparisions & are decreasing by 1 when pass increases

~~fata~~ ~~comp~~

Pass 1 → 4 comparisions
" 2 → 3 "
" 3 → 2 "
" 4 → 1 "

Comparisions decreasing by 1 for every pass.

code:-

```
#include <stdio.h>
Void main()
{
    int A[100], n, i, j, temp;
    printf("Enter size");
    scanf("%d", &n);
    printf("Enter %d elements", n);
    for(i=0; i<n; i++)
        scanf("%d", &A[i]);
    for(i=1; i<=n-1; i++)
    {
        for(j=0; j<=n-i-1; j++)
        {
            if(A[j] > A[j+1])
            {
                temp = A[j];
                A[j] = A[j+1];
                A[j+1] = temp;
            }
        }
    }
}
```

(Ascending order)

Input size & Array elements

No. of passes

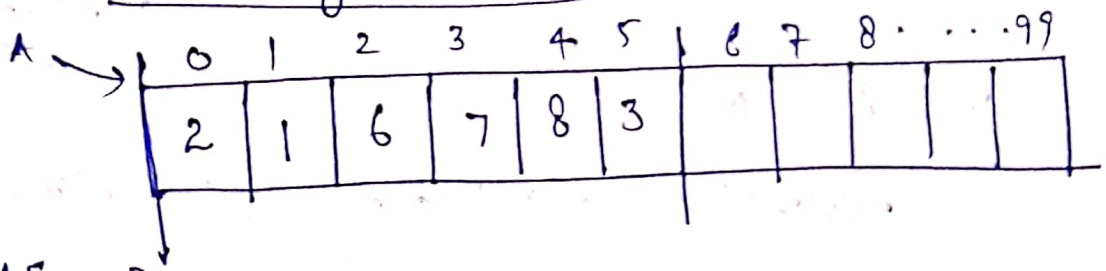
No. of comparisions for that pass

swap.

```
printf(" Array after Sorting ");
for ( i=0; i<n; i++)
    printf("%d", A[i]);
}
```

(II.) Deletion of an element from an Array :-

(iii) Deletion by element :-



int A[100]

n = 6 (No. of elements)

element to be (ele) = 6 deleted

Process :- we will first seach that element & then will delete it, if not found then array will remain same.

Code :- you can do it

```
{ Input size, elements of array & element
  to be deleted
for ( i=0; i<n; i++)
{
if (A[i] == ele)
```

P.T.O.

```
f=0;
for(i=0; i<n; i++)
{
    if(A[i]==ele)
    {
        f=1;
        for(j=i; j<=n-2; j++)
            A[j]=A[j+1];
        n=n-1;
        //break;
    }
}
if(f==0)
    printf("element Not found, so can't delete it");
printf("Final Array");
for(i=0; i<n; i++)
    printf("%d\t", A[i]);
```

if use then only delete 1st occurence

if not use, then delete All occurence