Operations ON Array.

We can perform different operations on array as listed below:-

(i) Traversing
(ii) Inserting
(iii) Deleting
(iv) Searching
(v) Sorting
(vi) Merging

(i) Traversing:- Means visiting each element.

- Traversing means visiting and accessing each element of array exactly once.
- To access the elements in the array the index is varied from lower bound to upper bound.
- To traverse an array, we use loop that iterates from lower bound to upper bound.
- For example:- we want to print the elements of the array. we will start with first element. Print this. Then we will increment the

__/__/____

index value and will print the next element and so on till we reach the last element.

→ So here we are visiting each element exactly once. This is the case of traversal.

- Algorithm :- Traverse (A, LB, UB).
- A linear array A is declared with Lower bound LB and Upper bound UB.
- This algorithm traverse an array.

```
Step 1 : start
Step 2 : for i = LB to UB
             process A[i]
Step 3 ; end
```

- C Program to Traverse array.

```
#include <stdio.h>
#include <conio.h>
void main()
{
 int a[5], i, n;
printf(" Enter elements of array");
 for(i=0; i<n ; i++)
    {
        scanf("%d", &a[i]);
    }
```

```
printf("\n Elements of Array are \n");
for(i=0; i<n; i++)
  {
   printf("%d \n", a[i]);
  }
getch();
}
```

OUTPUT:-

```
Enter elements of array:
10
25
15
13
30
Elements of Array are:
10
25
15
13
30
```

- Traversal in 2-D Array

(Algorithm) (a, m, n) :- 'a' is 2-D array with 'm' rows and 'n' columns.

1. Start
2. for i = 1 to m
3. for j = 1 to n
4. process a[i][j]
5. end for
6. end for
7. end

(ii) Insertion :-

- Insertion means adding a new element to an array.
- Insertion of an element of linear array is at the end.
- Element can be inserted at any position in the array.
- When element is to be inserted at the end of array, the array size is incremented by 1 and element is inserted at the last position.

Before Insertion :-

1	2	3	4	5	6
10	70	50	40		

After Insertion :-

1	2	3	4	5	6
10	70	50	40	60	

→ Now suppose we want to add a value 30 at 3rd position. Then 3rd, 4th and 5th elements will be shifted down thereby creating space at 3rd location.

__/__/____

- Before Insertion at k^{th} position:-

1	2	3	4	5	6
10	70		50	40	60

- Array After Insertion at k^{th} position:-

1	2	3	4	5	6
10	70	30	50	40	60

* Algorithm Insert (a, n, max, k, item) :-

a → array, n → elements, max → max size, k → Position, item → value to be inserted

1. Start
2. if n = max then
3. print "Max array size reached. Can't Insert value".
4. exit
5. end if
6. for i = n to k step -1
   a[i+1] = a[i]
7. a[k] = item
8. n = n+1
9. end

__/__/____

C Program to Insert element an array.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[10], i, n, k, item;
printf("Enter number of elements");
scanf("%d",&n);
printf("Enter elements");
for(i=0; i<n; i++)
  {
    scanf("%d", &a[i]);
  }
printf("Enter element to be inserted");
scanf("%d", &item);
printf("Enter position where element is to be inserted");
scanf("%d",&k);
for(i=n; i>=k; i--)
  {
    a[i+1] = a[i];
  }
a[k] = item;
n++;
printf("Array elements after Insertion :\n");
for(i=0; i<n; i++)
  {
    printf("%d", a[i]);
  }
getch();
}
```

__/__/____

Output:-

```
Enter Number of elements 5
Enter elements
12
65
70
42
35
Enter element to be inserted 55
Enter position where element is to be inserted 3
Array elements After Insertion
12
65
55
70
42
35
```

3. Deletion:-

- Deletion means removing an element from the list.
- To delete an element, that element is to be searched in the list.
- An array named, A is a Linear Array having n elements in it.

Example:- Suppose that we have the following array with 5 elements and we want to delete 5th element

(Before deletion)

1	2	3	4	5	
10	70	50	40	60	

(After deletion)

1	2	3	4	5	
10	70	50	40		

→ Now Array size is 4.
Now suppose we want to delete second item as:-

1	2	3	4		
10		50	40		

→ Now 3rd element will be shifted to 2nd place and 4th element will be shifted to 3rd place to keep the array in order.

1	2	3	4		
10	50	40			

(After deletion operation)

(Array deleting an element from begining).

1	2	3	4		
50	40				

* Algorithm Delete (a, n, k, item) :-

1. Start
2. if n = 0 then
3. print" there is no element to delete in the lis
4. exit
5. end if
6. item = a[k] (Save value to be deleted)
7. [Shift elements]
   for i = k to n-1
   a[i] = a[i+1]
8. end for
9. n = n-1 (Decrement n)
10. End

C Program to delete element from an Array.

```
#include <stdio.h>
#include <conio.h>
void main()
{
int a[20], i, n, p, item;
printf("Enter number of elements");
scanf("%d", &n);
printf("Enter elements");
for(i=0; i<n; i++)
{
scanf("%d", &a[i]);
}
printf("Enter Position from where element
        is to be deleted");
scanf("%d", &p);

item = a[p];
for(i=p; i<n-1; i++)
{
    a[i] = a[i+1];
}
    n = n-1;

Printf("Array elements after deletion :\n");
```

```
for (i=0; i<n; i++)
  {
    printf("%d", a[i]);
  }
getch();
}
```

(OUTPUT):-

Enter No. of elements 5
Enter elements:
12
65
95
42
35

Enter position from where element is to be ~~inserted~~ deleted 3

Array elements after deletion:

12
65
42
35

4. Searching :-

This operation which finds the location of a given element in a list is called Searching.

Searching can be done by traversing the array until the element to be searched is found.

Types of Searching Methods :-
1. Linear Search
2. Binary Search.

5. Sorting :-

Sorting means arranging an array in an orderly fashion (ascending or descending).

0	1	2	3	4	5
1	9	11	13	8	5

Unsorted array

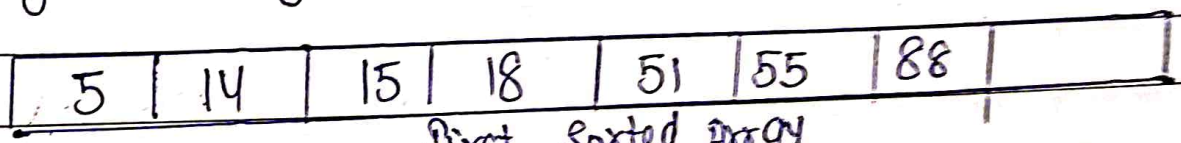
0	1	2	3	4	5
1	5	8	9	11	13

Sorted Array

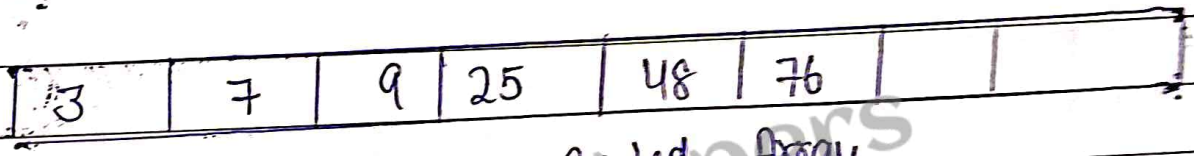
0	1	2	3	4	5
1	5	8	9	11	13

6. Merging:-

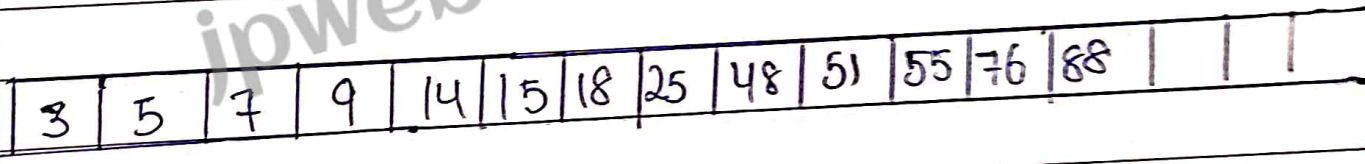
Merging Process is used to merge two or more different linear array in-to single array.



First Sorted Array



Second Sorted Array



(Merged Array - in which two sorted arrays are copied into third one).