Q1. What is array? What are the various operations that can be performed on arrays?

Ans. The fundamental data types (int, char, float, double), can store only one value at a given time. Therefore, they can be used only to handle limited amount of data. In some situations we need to handle a large volume of data for reading, processing and printing. And to process large amounts of data, we need a powerful data type that would facilitate efficient storing, accessing and manipulation of data. One such data type is **Array.**

An array is a fixed-size sequenced collection of elements of the same data type. It is simply a grouping of similar type data that share a common name. For example: List of employees in an organisation, List of marks of all students etc.

Arrays can be classified in three types :

1. One – dimensional arrays
2. Two dimensional arrays
3. Multi dimensional arrays

Operationals that can be performed on arrays :

The various operations that can be performed on Arrays are –

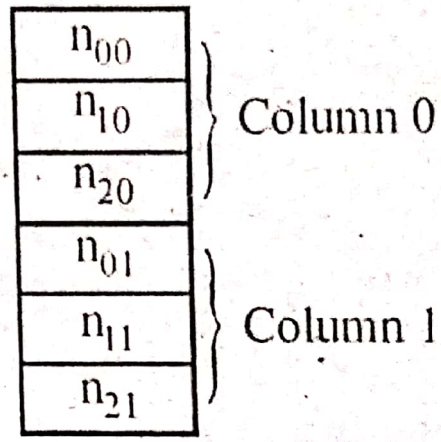
1. **Traversal :** Acceesing each element exactly once in order to process it. This operation is called visiting the element.
2. **Searching :** Finding the location of a given element.
3. **Insertion :** Adding a new element in the array.
4. **Deletion :** Removing an existing element from array.
5. **Sorting :** Arranging the elements in some logical order (ascending or descending)
6. **Merging :** Combining the element of two arrays into a single array.

Q8. Explain how two dimensional array is stored in memory.

Ans. Let a be a two dimensional m×n array. Though a is pictured as a rectangular pattern with m rows and n columns, it is represented in memory by a block of m*n sequential memory locations. However the sequence can be stored in two different ways.

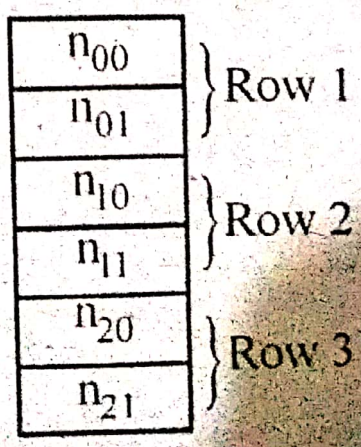
1. Column Major Order : In this the elements are stored column by column. First column, second column and so on.

For int n[3][2]; column major order would look like –



2. Row Major Order : In this the elements are stored row by row. First row, second row and so on.

For int n[3][2]; Row major order will look like –



Q1. What is Merging? How we can merge two arrays?

Ans. Merging is the process of combining the two arrays into a single array. Merging can be done in two ways :

1. Merging Unsorted Array : If the arrays are not sorted, we can combine them end to end i.e. we can first put the elements of first array into third array and then the elements of second array are placed after it in third array.

For Example :

Array A	12	40	2	5

Array B	15	5	10	7	8

Array C (after merging)	12	40	2	5	15	5	10	7	8

2. Merging Sorted Array: In many cases, we may have sorted arrays and our aim is to combine them in such way that the combined array is also in sorted order.

One straight forward approach is to join them end to end and then sort the combined array, but this approach is not efficient and economical.

Therefore the best approach is to compare the elements of the given array, and based on this comparision, decide which element should go first to the third array.

For Example :

Array A	1	2	12	40

Array B	5	7	10	15	60

Array C	1	2	5	7	10	12	15	40	60

First it will compare 1 & 5, 1 will go to third array. Then 2 & 5, 2 will go to C and so on.

// WAP TO READ AND PRINT THE ELEMENTS OF AN ARRAY

(1)

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[4],i;
clrscr();
printf(" Enter the elements of array");
for(i=0;i<4;i++)   //scan four elements from the keyboard by using loop
{
scanf("%d",&a[i]);
}
printf(" The elements of array are");
for(i=0;i<4;i++)   //display four elements by using loop
{
printf("\n %d",a[i]);
}
getch();
}
```

display

//WAP TO READ & PRINT THE ELEMENTS OF 2-D ARRAY

(2)

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[3][3],i,j;
clrscr();
printf("Enter the elements of 2-D array");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
scanf("%d",&a[i][j]);
}
}
printf("The elements of array are:\n");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
printf("%d\t",a[i][j]);
}
printf("\n");
}
getch();
}
```