ARRAYS

1.1 DECLARING AND INITIALIZING ONE-DIMENSIONAL ARRAY AND ARRAY OPERATIONS

- ✓ An array is a fixed-size sequenced collection of elements of the same datatype that shares a common name.
- ✓ An array is a collection of variables of same datatype known by a same name

Example:

Marks of a class of students.
List of employees in an organization.

- ➢ **Types of Arrays**

- ✓ One dimensional array
- ✓ Two dimensional array
- ✓ Multi dimensional array

DECLARING ONE DIMENSIONAL ARRAYS

- ✓ **Syntax:** Data type array_name [size] ;

- ✓ **Data type:** it defines the type of element that will be contained in the array, like int,float,char,double etc.
- ✓ **Array name**: it is the name of variable which represent array.
- ✓ **Size** : it indicates the maximum no.of elements that can be stored inside the array.

For example:

```
int a[5];
float height[10];
char name[10];
```

- ✓ In one dimensional array, individual element of any array is identified using name of array & index.
- ✓ In one dimensional array, the index always starts with 0.
- ✓ First element of array is a[0],second is a[1],and so on….

- ✓ Any reference to the arrays outside the declared limit would not necessarily cause an error.rather, it might result in unpredictable program results

INITIALIZING ONE-DIMENSIONAL ARRAY

- Array can be initialized in two ways
  - ✓ Compile time initialization
  - ✓ Run time initialization
- **Compile Time Initialization**
  - We can initialize the elements of an array at the time of declaration.

- **Syntax:** Data type array_name [size]={ list of values };

✓ The values in the list are separated by commas,

✓ If we initialized more value than declared size, the compiler will produce an error.

- **Example 1:** int a[3]={23,12,32};
  - here,a[0]=23, a[1]=12, a[2]=32;
- **Example 2:** int num[5]={54,23,3};
  - Here, num[0]=54, num[1]=23, num[2]=3, num[3]=0, num[4]=0,

✓ If we don't initialize array elements at the time of declaration then by default all elements are initialized to 0

- **Example 3:** int a[]={10,20,40,70,60};
  - The above statement will create an array of 5 integer elements.

✓ If we don't specify the size of array at time of declaring & initialization, compiler will automatically counts the value specified in initialization and allocate memory for specified no. of elements.

INITIALIZING ONE-DIMENSIONAL ARRAY

- **Run Time Initialization**

✓ An array can be explicitly initialized at run time.

- **Example:**

```
int a[3];
scanf("%d  %d %d", &a[0], &a[1], &a[2]);
```

- **Example:**

```
int a[3];
printf("enter array elements :\n");
for (i=0;i<3;i++)
        {
                scanf("%d",&a[i]);
        }
```

A PROGRAM TO STORE & DISPLAY THE VALUES IN AN ARRAY

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int a[5],i;
        clrscr( );
        printf("Enter the elements :");
        for(i=0;i<5;i++)
        {
                scanf("%d",&a[i]);
        }
        printf("Elements of array are :\n");
        for(i=0;i<5;i++)
        {
                printf("Element no %d = %d \n",i+1,a[i]);
        }
        getch();
        }
```

ADVANTAGE OF AN ARRAY

- An array is a fixed-size sequenced collection of elements of the same data type that shares a common name.
- We can use one name for similar elements.
- Two-Dimensional array are used to represents matrices.
- Array is used to implement other data structure like link list, stack,tree etc.

CHARACTERISTICS OF ARRAY

- Array store elements that have same data type.
- Array store elements in subsequent memory location.
- Array size should be mention in the declaration.
- Array name represent the address of starting elements.

LIST OF OPERATIONS ON ONE DIMENSIONAL ARRAY

- **Sorting**: We can sort the elements of array in ascending and descending order.
- **Merging:** We can merge or joint elements of Two One Dimensional array into third one Dimensional array.
- **Searching** :We can search the any elements from the given array.
- **Insertion** :We can insert the elements into array at beginning or ending or at specific position.
- **Deletion** :We can delete the elements in array at beginning or ending or at specific position.

INSERT OPERATION

- We can insert the elements from the array at beginning of array or at ending of array or at specific position of array.

Example:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[5]={10,20,30,40,50};
    int val,i,pos,temp,n=5;
    clrscr();
    printf("Elements of array A before insertion : \n");
    for(i=0; i<n ;i++)
    {
        printf("%d\n",a[i]);
    }
```

```
printf("Enter the element to be inserted into array A : ");
scanf("%d",&val);
printf("Enter the index of the element :");
scanf("%d",&pos);
temp=n;
While(temp-1 >= pos)
{
        a[temp+1]=a[temp];
        temp--;
}
a[temp]=val;
n=n+1;
printf("Elements of array A after insertion : \n");
for(i=0;i<n;i++)
{
        printf("%d\n",a[i]);
}
getch();
}
```

DELETE OPERATION

- We can delete the elements from the array at beginning of array or at ending of array or at specific position of array.

Example:

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a[5]={10,20,30,40,50};
        int i,pos,n=5;
        clrscr();
        printf("Before Deletion elements of Arrays are as below:\n");
        for(i=0;i<5;i++)
        {
                printf("%d\n",a[i]);
        }
```

CONT…

```
printf("Enter the position of element to delete: ");
scanf("%d",&pos);
while(pos<=n-1)
{
        a[pos]=a[pos+1];
        pos++;
}
n=n-1;
printf("Array Elements after deletion:\n");
for(i=0;i<n;i++)
{       printf("%d\n",a[i]);
}
getch();
}
```

SEARCH OPERATION

- This operation is used to search particular element from one dimensional array

```
#include<stdio.h>
#include<conio.h>
void main()
{       int a[5]={10,20,30,40,50};
        int val,i,f=0;
        clrscr();
        printf("\nenter element to search in array :");
        scanf("%d",&val);
        for(i=0; i<5; i++)
        {       if(a[i] == val)
                        {   printf("\nElement is found & element is at :a[%d] =%d",a[i],val);
                            f=1;
                        }
        }
        if(f==0)
                {       printf("\nElement is not found");
                }
        getch();
        }
```

MERGE OPERATION

- This operation is used to merge two one dimensional array into third one dimensional array.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a[5],b[5],c[10],i,j,k=0;
        clrscr();
        printf("Enter elements of array A :\n");
        for(i=0;i<5;i++)
        {
                scanf("%d",&a[i]);
        }
        printf("Enter elements of array B :\n");
        for(j=0;j<5;j++)
        {
                scanf("%d",&b[j]);
        }
```

CONT…

```
for(i=0;i<5;i++)
{
        c[k]=a[i];
        k++;
}
for(j=0;j<5;j++)
{
        c[k]=b[j];
        k++;
}
printf("Elements of array C after Merge Operation :\n");
for(k=0;k<10;k++)
{
        printf("%d\n",c[k]);
}
getch();
}
```

SORT OPERATION

- This operation is used to sort elements of one dimensional array in to specific order, it is either in ascending order or descending order.

```
#include<stdio.h>
#include<conio.h>
void main( )
{       int a[5]= {10,40,50,30,20};
        int n=5,i,j,temp;
        clrscr( );
        for(i=0;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;
                        }
                }
        }
```

CONT...

```
printf("Array Elements in Ascending Order is: \n");
for(i=0;i<5;i++)
{
    printf("\n%d",a[i]);
    printf("\n");
}
getch();
}
```

1.2 INTRODUCTION OF STRING AS ARRAY OF CHARACTERS DECLARATION AND INITIALIZATION OF STRING

- A string is a sequence of characters that is treated as single data item.
- It is written between double quotation marks.

- **Declaring of String Variable**
  - A string variable is declared as an array of characters.
- **Syntax:** char str_name [size];
- here, size is the number of characters.
- **Example:**
  - char city[15];

- The common operation performed on character strings :
  - Reading and writing strings
  - Combining strings together
  - Copying one string to another
  - Comparing strings for equality
  - Extracting a portion of string

CONT…

- **Initialization of string variable**
- **Example 1:**
  - char city[9]="NEW DELHI";
  - char city[9]={'N','E','W',' ','D','E','L','H','I','\0'};
- **Example 2:**
  - char cty_nm[10]= "BANGALORE";
- which declares the name as a character array variable that can hold a maximum of 10 characters.
  - char cty_nm[10]={'B','A','N','G','A','L','O','R','E','\0'};
- When the compiler sees a character string, then it terminate it with an additional null character. So, the element cty_nm[10] holds the null character '\0'.
- When declaring character arrays, we must allow one extra element space for the null character('\0').
- **Example 3:**
  - char str_nm [] = {'M,'O','N','D','A','Y','\0'}

  Here array str_nm as a 7 element array.
- If char array is initialized without specifying the no. of elements.in such cases, size of array will be determined automatically, based on no. of element initialized.

1.3 TWO-DIMENSIONALARRAY

- Two dimensional array is a collection of elements of same data types shares common name and having two dimension.
- First dimension indicates the row index and second dimension indicates column
- Two dimensional arrays is used to create a matrix.

- **Syntax of Two dimensional arrays:**
- datatype array_name [row-size] [column-size];
- Example: int a[3][3];
- Here row size =3 and column size=3.

	Column 0	Column 1	Column 2
Row 0	a[0][0]	a[0][1]	a[0][2]
Row 1	a[1][0]	a[1][1]	a[1][2]
Row 2	a[2][0]	a[2][1]	a[2][2]

- In Two dimensional array row index and column index always starts with 0.

Initialization of Two dimensional arrays:

✓ we can initialize the elements of two dimensional array at the time of declaration

- General form of initialization of array is :

  datatype array_name [row size] [column size] ={list of values};

- If we initialized more value than declared size, the compiler will produce an error.

Example :

```
int a[2][2]={{4,5,6};
```

- here, a[0][0]=4, a[0][1]=5, a[1][0]=6, a[1][1]=0,
- If we don't initialize array elements at the time of declaration then by default all elements are initialized to 0
- We can also initialize a two dimensional array in the form of matrix as given below.

Example :

```
int A[3][3]={   {11,12,13},
                {14,15,16},
                {17,18,19}      };
```

Example :

```
int a[2][3] = {  {1,1},
                 {2}};
```

here first two elements of first row is 1 and first element of second row is 2.

CONT..

✓ We can also initialize array elements at run time

Example:

```
int a[3][3],i,j;
clrscr();
printf("Enter element into 3 X 3 matrix A :");
for(i=0;i<3;i++)
{
        for(j=0;j<3;j++)
        {
                scanf("%d",&a[i][j]);
        }
}
```

MATRIX ADDITION OPERATION

	0	1	2	3
0	1	0	5	4
1	3	-1	7	2
2	8	2	4	0
3	6	5	-2	3

+

	0	1	2	3
0	2	-1	4	3
1	3	1	0	5
2	1	3	5	0
3	2	1	4	2

=

	0	1	2	3
0	3	1	9	7
1	6	0	7	7
2	9	5	9	0
3	8	6	2	5

PROGRAM TO ADD TWO MATRICES OF SIZE 3 X 3

```
#include<stdio.h>
#include<conio.h>
void main()
{    int a[3][3],b[3][3],c[3][3],i,j;
     clrscr();
     printf("Enter element into 3 X 3 matrix A :");
     for(i=0;i<3;i++)
     {          for(j=0;j<3;j++)
                {          scanf("%d",&a[i][j]);
                }
     }
     printf("Enter element into 3 X 3 matrix B :");
     for(i=0;i<3;i++)
     {          for(j=0;j<3;j++)
                {          scanf("%d",&b[i][j]);
                }
     }
```

CONT…

```
        for(i=0;i<3;i++)
        {
                for(j=0;j<3;j++)
                {
                        c[i][j]=a[i][j]+b[i][j];
                }
        }
        printf(“Elements of Matrix C is given below :”);
        for(i=0;i<3;i++)
        {       for(j=0;j<3;j++)
                {
                        printf(" %3d",c[i][j]);
                }
        printf("\n");
        }
        getch();
        }
```

1.4 MULTI-DIMENSIONALARRAYS

- Three or more dimensions array is collection of elements of same data types shares common name
- It is an array of arrays,an array that has multiple levels.

- The general form of multidimensional array is
  Data_type array_name[size1][size2]....[sizeN];

Example: int a[5][10][20];

Here,Array int a[5][10][20] can store total (5*10*20)=1000elements.

Example : float table[5][4][5][3];

1.5 SSCANF() AND SPRINTF() FUNCTIONS

- **sscanf() function :** it is used to reads data from character array instead of standard input.
- **Syntax**:
- sscanf(Name of character Array, “format specification", variables);

- This will extract the data from the character array according to the conversion specifier and store into the respective variables.

- sscanf() will read subsequent characters until a whitespace is found (whitespace characters are blank, newline and tab).

CONT…

PROGRRAM OF SSCANF() FUNCTION

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        char name[50]={“ABC DEF GHI”};
        char f_name[10],m_name[20],l_name[10];
        sscanf(name,”%s %s %s”,f_name,m_name,l_name);
        printf(“First Name = %s“,f_name);
        printf(“Middle Name = %s“,m_name);
        printf(“Last Name = %s“,l_name);
        getch();
}
```

- **sprintf() function:** it is used to writes data to character array instead of standard input.
- Sprint() function writes the formatted text to a character array.
- **Syntax:**

  sprintf(Name of Character Array,"Conversion Specifier", variables);

PROGRAM FOR SPRINTF() FUNCTION

```
#include<stdio.h>
#include<conio.h>
void main( )
{       char name[50];
        char fname[10]= {“ABC”};
        char mname[20]={“DEF”};
        char lname[10]={“GHI”};
        sprintf(name,”%s %s %s”,fname,mname,lname);
        printf(“Full Name = %s”,name);
        getch();
}
```

1.6 DRAWBACKS OF LINEAR ARRAY

- The size of array mast is constant or known at compile time.
  - If no. of element stored in array is less than size then memory may be wasted

- Once we declare the size of array it can not be changed at run time

- The operation of Array is more complex and time consuming
  like insertion, deletion, sorting.
  - To insert element in array we have shift element to create space for new element
  - To delete element from array we have to shift element to take vacant space.