

16

ARRAY

- **ARRAY:** Array is a collection of values of same data type. In other words, Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.
- To declare an array, define the variable type with **square brackets**:
Ex: `int arr[];`
For declaring an array you have to use following syntax:
`datatype array_name[size];`
For example `int A[5];`
This statement declares an array A which allocates 5 continuous memory locations for storing integer data type. These locations are A[0], A[1], A[2],A[3],A[4]
- **SYNTAX TO INITIALIZE ARRAY ELEMENTS:**
`datatype array_name[size]={ list of elements separated by comma };`
The list of elements should be separated by commas.
For example: `int A[5]={1,2,3,4,5,};`
Here A is an integer array of size 5. A[0] will have the value 1, A[1] will have the value 2, A[2] will have the value 3, A[3] will have the value 4, A[4] will have the value 5.
- While initializing the array elements we can leave size as blank.
For example, `int A[]={1,2,3,4,5}.`
If the size of array is more than the initialized elements then for non-initialized array elements it will take garbage value
- **RUN-TIME INITIALIZATION:**
`int A[5];`
`cout<< "Enter elements " ;`
`for(i=0;i<=4;i++)`
`{ cin>>A[i];`
Here, A is an integer array of size 5. When you run this program, it will ask the value for A[0], A[1]... A[4]. Those values will be stored in the array A.
- **ACCESSING ARRAY ELEMENTS:**
`int i;`
`int A[2]={10,20,30};`
`for (i=0; i <= 2; i++) { cout << A [i] ; }`
This program will display A[0], A[1] and A[2] values as 10, 20 and 30 respectively.
- **PROCESSING OF AN ARRAY:**
The following operations can be performed on Arrays.
 - Traversal
 - Searching
 - Sorting
 - Insertion
 - Deletion
- **TRAVERSAL:** It means to access each location or each element of an array.
For Example: `include <iostream>`
`using namespace std;`
`int main()`
`{ //Initializing array`
`int arr[5]={ 10, 0, 20, 0, 30};`

```
//traversing array
for (int i = 0; i < 5; i++)
{
    cout<<arr[i]<<"\n";
}
}
```

Output:

10
0
20
0
30

- **SEARCHING:** This method finds out whether the data entered by the user is present in an array or not. There are two types of searching method.
 - Linear or Sequential search
 - Binary search

- **LINEAR/SEQUENTIAL SEARCH:** This method is slower, inefficient and works on unsorted list.

Example 2 : // Linear search

```
# include < iostream.h >
void main ( )
{ int A[5], i, data, flag = 0;
  cout << "Enter five values";
  for (i = 0; i < 5; i++) cin >> A [i];
  cout << 'Enter data to be searched'
  cin >> data;
  for (i=0; i < 5; i ++ )
  { if (A[i] == data) flag = 1; } if (flag
  == 1) cout << "Data present";
  else
  cout << "Data not present"; }
```

Output:

Enter five values
34
67
28
54
15

Enter data to be searched

67

Data Present

- **BINARY SEARCH:** Binary Search is a method to find the required element in a sorted array by repeatedly halving the array and searching in the half. ... This is done continuously until either the required data value is obtained or the remaining array is empty.

For Example:

```
#include<iostream>
using namespace std;
int main()
{
    int i, arr[10], num, first, last, middle;
    cout<<"Enter 10 Elements (in
    ascending order): ";
    for(i=0; i<10; i++)
        cin>>arr[i];
    cout<<"\nEnter Element to be
    Search: ";
    cin>>num;
    first = 0;
    last = 9;
    middle = (first+last)/2;
    while(first <= last)
    {
        if(arr[middle]<num)
            first = middle+1;
        else if(arr[middle]==num)
        {
            cout<<"\nThe number,
            "<<num<<" found at Position
            "<<middle+1;
            break;
        }
        else
            last = middle-1;
        middle = (first+last)/2;
    }
    if(first>last)
```

```

    cout<<"\n The number,
"<<num<<" is not found in given
Array";
    cout<<endl;
    return 0;
}

```

Output:

Enter 10 elements (in ascending order):

1
2
3
4
5
6
7
8
9
10

Enter element to be searched: 8

The number ,8 found at position 8

- **SORTING:** It is a process to arrange the list either in ascending or descending order.
- **Bubble Sort-** Bubble sort, sometimes referred to as sinking sort. It is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted.

Ex:- int A[5];

1	9	5	0	2
0	1	2	3	4

Process: A[0] is compared with A[1]. If A[0] is greater than A[1], the values are swapped. Then A[1] is compared with A[2], A[2] is compared with A[3], and A[3] is compared with A[4]. In all cases if the ith location has value greater than i + 1th location, the values are swapped. The entire process is repeated N-1 times where N is the number of data in an array.

```

#include < iostream.h >
const int N = 5;
void main ( )
{
int A [N], i, j, T; cout << "Enter values";
for (i = 0; i < N; i++)
cin >> A [ i ]; // sorting
for ( i = 0; i < N - 1; i++)
for (j = 0; j < N - i; j++)
if (A[j] > A [j + 1])
{
T = A [ j ];
A [ j ] = A [ j + 1 ];
A [ j + 1 ] = T;
}
cout << "Sorted array is \n";
for (i = 0; i < N; i++)
cout << A [ i ]<< "\n";
}

```

Output:

Enter values 17

8
39
54
32

Sorted array is: 8

17
32
39
54

Selection Sort: Consider an array having N elements to be sorted in ascending order. Initially, first element is compared with others so that it holds the smallest value. In the next pass, second element is compared with others so that it holds the second smallest value. This procedure is repeated for the entire array.

For Example: # include < iostream.h >
const int N = 5;

```
void main ( )
{
int A [N], i, j, T;
cout << "Enter values";
for ( i = 0; i < N; i ++ )
cin >> A [ i ];
// sorting
for ( i = 0; i < N - 1; i ++ )
for ( j = 1; j < N; j ++ )
if (A [ i ] > A [j])
{ T = A [i];
A [i] = A [j];
A [j] = T;
}
// printing the sorted data
cout << "Sorted array is :";

for ( i = 0; i < N; i ++ )
cout << A [ i ];
}
```

Output: Enter values 17

8

39

54

32

Sorted array is: 8

17

32

39

54

- **INSERTION:** It means addition of a data item in the middle or at the end of the array.

Example : # include < iostream.h >
void main ()

```
{
int x [20]; int i, loc, n, data;
cout << "Enter the no. of elements";
cin >> n;
for (i = 0; i < n; i ++ )
{
cout << "Enter the array element";
cin >> x [ i ];
}
cout << "Enter the location after which
data is to be inserted";
cin >> loc;
for (i = n - 1; i >= loc; i - - )
x [ i + 1 ] = x [ i ];
cout << "enter the new data to be
added";
cin >> x [loc];
n ++ ;
cout << "Array elements after
insertion";
for (i = 0; i < n; i ++ )
cout << x [ i ]; }
```

Let data 12 to be inserted at location 2

Before insertion:

6	8	9	7	10
X[0]		X[1]	X[2]	X[3]
X[4]				

After insertion:

6	8	12	9	7	10
X[0]	X[1]	X[2]	X[3]	X[4]	X[5]

- **DELETION:** It means removal of a data. Example : # include <iostream.h

```
>
void main ( )
{
    int x [20];
    int i, j, n, loc, data;
    cout << "Enter the no. of elements";
    cin >> n;
    for (i = 0; i < n; i++)
    {
        cout << "Enter value";
        cin >> x [ i ];
    }
    cout << "Enter the location to be
    deleted";
    cin >> loc;
    if (loc != 0)
    {
        data = x [loc ];
        for {j = loc; j < n - 1; j++ )
        x [ j ] = x [ j + 1];
    }
    n = n - 1;
    cout << "Elements after deletion";
    for ( i = 0; i < n; i++)
        cout << x [ i ];
    }
```

- **ARRAY OF STRINGS:** Array of string is nothing but an array of characters.
- Syntax for declaring an array of string is:
char array_name [size]={array of character in single quotes separated by commas}

char

name[5]={'G','A','U','R','A','V'}

- String is a series of character stored in continuous memory location. The last character of each string is a null character '\0'.

Char

name[5]={'G','A','U','R','A','V',
'\0'};

CHECK YOURSELF

1. Which of the following correctly declares an array?
 - A. int array[10];
 - B. int array;
 - C. array{10};
 - D. array array[10];
2. The sequence of objects that have the same data type, is called _____.
3. What is the index number of the last element of an array with 9 elements?
 - A. 9
 - B. 8
 - C. 0
 - D. Programmer-defined
4. Which of the following accesses the seventh element stored in array?
 - A. array[6];
 - B. array[7];
 - C. array(7);
 - D. array;

5. Which of the following gives the memory address of the first element in array?

- A. array[0];
- B. array[1];
- C. array(2);
- D. array;

ANSWERS

Answers to Check Yourself:

- 1. A
- 2. Array
- 3. B
- 4. A
- 5. D

STRETCH YOURSELF

1. Define array? Give an example to enter elements to an array.

2. What is the output of this program?

```
#include <iostream.h >
using namespace std;
int main()
{
char str[5] = "ABC";
cout << str[3];
cout << str;
return 0;
}
```

3. What is sorting ? Give an example of bubble sort.