# Arrays

# 1. Arrays

An array is a group of consecutive, adjacent memory locations (i.e. elements) that all have the same data type. Arrays may have from one to several dimensions. We will study the one-dimensional (1D) and two-dimensional (2D) arrays.

## 1.1 1D Array

### 1.1.1 Definition:

*data type arrayName[ Size ];*

The *Size* must be an integer constant greater than zero.

For example:

```
int    a[10];
char   name[20];
float  temperature[6];
```

### 11.2 Accessing array elements:

*arrayName[index]*

- All arrays have 0 as the index of their first element and *Size-1* as the index of their last element.
- The *arrayName* represents the address of the first element in the array.

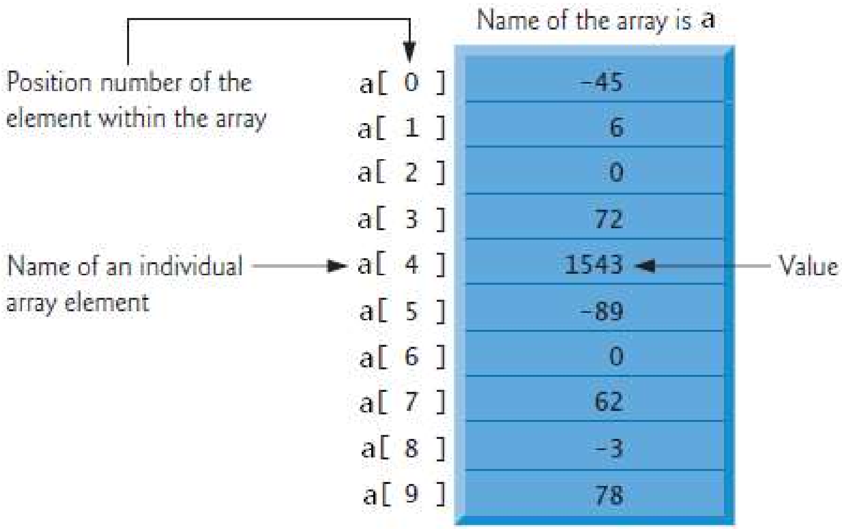
For example:

```
int   a[10];
```

The first element is `a[0]`

The last element is `a[9]`

The array is a[0] , a[1] , a[2] , . . . , a[9]



For example:

- `a[3] = 60;       // assign 60 to the fourth element`
- `cin >> mark[3];  // read the value of the 4th mark`

-
  ```
  for(int i=0; i<10; i++)
      cin >> a[i];  // input values to the array
  ```

-
  ```
  for(int j=0; j<10; j++)
      cout << a[j]; // print values of the array
  ```

**Example:** Write a C++ program that loads an integer array with the numbers 0 through 9 and prints the array values.

```
#include <iostream.h>
void main()
{
  int a[10];
  for(int i=0 ; i<10 ; i++)
     a[i] = i;
  cout << "Array is " << endl;
  for(i=0; i<10 ; i++)
     cout << a[i] << " ";
}
```

## 1.1.2 Array initialization:

C++ allows the initialization of arrays at the time of their declaration. For example:

```
   int a[5] = { 8 , 5 , 13 , 2 , 9};
   int a[ ] = { 8 , 5 , 13 , 2 , 9};
```

**Example:** Write a C++ program that calculates the sum and average of an initialized integer array.

```
#include <iostream.h>
void main()
{
  int b[5] = { 9 , 3 , 11 , 7 , 1 };
  int sum  = 0;
  for(int i=0 ; i<5 ; i++)
       sum += b[i];
  cout<<"Sum is " << sum << endl
      <<"Average is " <<sum/5.0;
}
```

**Example:** Write a C++ program that inputs ten integer values into an array and finds the maximum number in the array.

```
#include <iostream.h>
void main()
{
  const int size = 10;
  int c[size] , max;
  cout<<"Enter ten integer values: ";
  for(int i=0 ; i<10 ; i++)
      cin >> c[i];
  max = c[0];
  for(i=1 ; i < 10 ; i++)
  if(c[i] > max)
     max = c[i];
  cout<<"The maximum number is " << max;
}
```

**Example:** Write a C++ program that computes the number of even integer numbers in an array entered by the user.

```
#include <iostream.h>
void main()
{
  const int size = 10;
  int a[size] , count = 0;
  cout<<"Enter ten integer numbers: ";
  for(int i=0 ; i<10 ; i++)
    {
     cin >> a[i];
     if(a[i] % 2 == 0)
       count++;
    }
  cout<<"The number of even numbers is " << count;
}
```

**Note**

Only constants can be used to declare the size of arrays. Not using a constant for this purpose will generate a compilation error.

**Example:** Write a C++ program that inputs an integer array a[10] and arranges it in an ascending order.

```
#include <iostream.h>
void main()
{
  const int size = 10;
  int a[size];
  cout<<"Enter ten integer array values: ";
  for (int i=0; i<size; i++)
     cin>>a[i];
  for (i=0; i<size-1; i++)
   for(int j=i+1; j<size; j++)
    if(a[i] > a[j])
     {
       int temp = a[i];
       a[i] = a[j];
       a[j] = temp;
     }
  cout << "Array in ascending order: " <<endl;
  for(i=0; i<size ;i++)
    cout<<a[i]<<" ";
}
```

## 1.2 2D Array (Matrix)

Two-dimensional arrays consist of values arranged in rows and columns.

### 1.2.1 Definition:

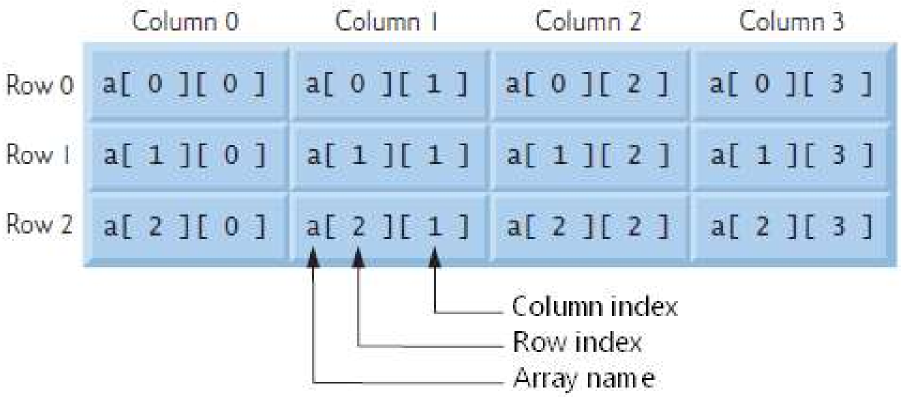
*data type arrayName[ RowSize ][ColumnSize];*

For example:

```
int   a[3][4];
float b[10][20];
```

### 1.2.2 Accessing 2D array elements:

*arrayName[RowIndex] [ColumnIndex]*



|  | Column 0 | Column 1 | Column 2 | Column 3 |
|---|---|---|---|---|
| Row 0 | a[ 0 ][ 0 ] | a[ 0 ][ 1 ] | a[ 0 ][ 2 ] | a[ 0 ][ 3 ] |
| Row 1 | a[ 1 ][ 0 ] | a[ 1 ][ 1 ] | a[ 1 ][ 2 ] | a[ 1 ][ 3 ] |
| Row 2 | a[ 2 ][ 0 ] | a[ 2 ][ 1 ] | a[ 2 ][ 2 ] | a[ 2 ][ 3 ] |

For example:

-
  ```
  a[3][4] = 60;
  ```

-
  ```
  cin >> mark[3][1];
  ```

-
  ```
  for(int i=0; i<10; i++)
   for(int j=0; j<10; j++)
     cin >> a[i][j]; // input values to the 2D array
  ```

-
  ```
  for(int m=0; m<10; m++)
  {
   for(int n=0; n<10; n++)
    cout<< a[m][n]<<”\t”; //print values of 2D array
   cout<<endl;
  }
  ```

### 1.2.3 2D Array initialization:

```
 int  b[2][2] = { {1 , 2} , {3 , 4} };
 int  a[3][4] = { {1 , 2 , 3 , 4} , {5 , 6 , 7 , 8} ,
                 {3 , 4 , 1 , 2} };
```

**Example:** Write a C++ program that adds two initialized 3×4 matrices A and B and then stores the result in a matrix C.

```
#include <iostream.h>
void main()
{
  int A[3][4] = { {1, 4, 3, 2},
                  {5, 6, 7, 8},
                  {9, 10, 11, 12} };
  int B[3][4] = { {3, 4, 3, 1},
                  {8, 7, 5, 6},
                  {12, 9, 11, 8} };
  int C[3][4];
  for (int i=0; i<3; i++)
   {
    for (int j=0; j<4; j++)
     {
      C[i][j] = A[i][j] + B[i][j];
      cout << C[i][j] << "\t";
     }
  cout << endl;
 }
}
```

**Example:** Write a C++ program that finds the average of each row of a 3× 4 matrix input by the user.

```
#include <iostream.h>
void main()
{
  int a[3][4];
  int sum;
  cout<<"Enter 3x4 integer matrix: ";
  for (int i=0; i<3; i++)
   for (int j=0; j<4; j++)
     cin>>a[i][j];
  cout<<"Average of each row: "<<endl;


  for (i=0; i<3; i++)
   {
    sum = 0;
    for (j=0; j<4; j++)
       sum += a[i][j];
    cout<<sum/4.0<<endl;
   }
}
```

**Example:** Write a C++ program that exchanges row3 with row1 in a 4× 4 integer matrix input by the user.

```
#include <iostream.h>
void main()
{
  int a[4][4];
  cout<<"Enter 4x4 integer matrix: ";
  for (int i=0; i<4; i++)
   for (int j=0; j<4; j++)
     cin>>a[i][j];
  for (i=0; i<4; i++)
   {
    int temp = a[0][i];
    a[0][i] = a[2][i];
    a[2][i] = temp;
   }
  cout<<"Matrix after exchanging row3 with row1:"
      <<endl;
  for (i=0; i<4; i++)
   {
    for (j=0; j<4; j++)
      cout<<a[i][j]<<" ";
    cout<<endl;
   }
}
```

**Example:** Write a C++ program that inputs a 4×4 integer matrix and finds the maximum value in the primary diagonal and the minimum value in the secondary diagonal.

```
#include <iostream.h>
void main()
{
  int b[4][4] , max , min;
  cout<<"Enter 4x4 integer matrix: ";
  for (int i=0; i<4; i++)
   for (int j=0; j<4; j++)
     cin>>b[i][j];
  max = b[0][0];
  min = b[0][3];
  for (i=1; i<4; i++)
    {
     if(b[i][i] > max)
        max = b[i][i];
     if(b[i][3-i] < min)
        min = b[i][3-i];
    }
cout << "Max value is " << max <<endl
     << "Min value is " << min;
}
```

**Example:** Write a C++ program that multiplies 3×4 matrix by 4×3 matrix both are entered by the user. Then the program should store the result in a third matrix.

```
#include <iostream.h>
void main()
{
  const int row_a=3 , col_a=4 ,
            row_b=4 , col_b=3;
  int a[row_a][col_a];
  int b[row_b][col_b];
  int c[row_a][col_b];

  cout<<"Enter "<<row_a<<"x"<<col_a
      <<" integer matrix: " << endl;
  for (int i=0; i<row_a; i++)
   for (int j=0; j<col_a; j++)
     cin>>a[i][j];

  cout<<"Enter "<<row_b<<"x"<<col_b
      <<" integer matrix: " << endl;
  for (i=0; i<row_b; i++)
   for (j=0; j<col_b; j++)
     cin>>b[i][j];

  for(i=0; i<row_a; i++)
   for(j=0; j<col_b ; j++)
    {
      c[i][j] = 0;
      for(int k=0; k<col_a ; k++)
        c[i][j] += a[i][k] * b[k][j];
      }

  cout<<"Resulted Matrix is " << endl;
  for (i=0; i<row_a; i++)
   {
    for (j=0; j<col_b; j++)
      cout<<c[i][j]<<" ";
    cout<<endl;
   }
}
```

## 1.3 String (1D array of characters)

String is a character array that is terminated with null. Null is zero and can be expressed as NULL or '\0'. The compiler adds the null to the end of string automatically.

For example:

```
  char name[11];   // holds 10 characters plus null
  char str[12]= {’H’,’e’,’l’,’l’,’o’,’ ’,’t’,’h’,
                   ’e’,’r’,’e’};
```

***or***

```
  char str[12] = ”Hello there”;   //string constant plus
                                // null
```

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| str | H | e | l | l | o | | t | h | e | r | e | 0 |

We can input and output the string with or without using loop.

```
   cout << “Enter your name: ”;
   for(int i=0; i<11; i++)
    cin >> name[i];
```

**or**

```
   cout << “Enter your name: ”;
   cin >> name;


   cout << “Your name is ” << name << endl;
```

**Example:** Write a C++ program that reads a string and then computes the number of capital letters in the string.

```
#include <iostream.h>
void main()
{
  char str[30];
  int count = 0;
  cout<<"Enter your string: ";
  cin >> str;
  for(int i=0 ; str[i] ; i++)
   if(str[i] >= 'A' && str[i] <= 'Z')
      count++;
  cout<<"No. of capital letters is " << count;
}
```

**Example:** Write a C++ program that computes the length of a string entered by the user.

```
#include <iostream.h>
void main()
{
  char str[100];
  int length = 0;
  cout<<"Enter your string: ";
  cin >> str;
  for(int i=0; str[i] ; i++)
    length++;
  cout<<"Length of string is " << length;
}
```

**Example:** Write a C++ program that converts any capital letter to small in a string entered by the user.

```
#include <iostream.h>
void main()
{
  char str[20];
  cout<<"Enter your string: ";
  cin >> str;
  for (int i=0 ; str[i] ; i++)
    if (str[i] >= 'A' &&  str[i] <= 'Z')
      str[i] += 32;
  cout <<"\nConverted String: " << endl << str;
}
```

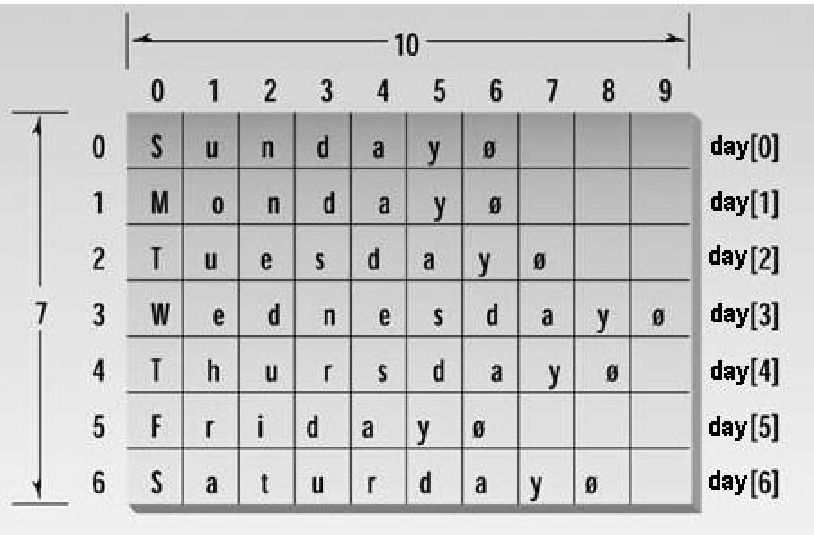
## 1.4 Array of strings (2D array of characters)

To create an array of strings, we use a two-dimensional character array.

The number of rows determines the number of strings and the number of columns specifies the maximum length of each string.

For example:

```
char str_array[30][80];
char day[7][10] = { "Sunday" , "Monday",
                    "Tuesday" , "Wednesday",
                    "Thursday", "Friday",
                    "Saturday" };
```



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | S | u | n | d | a | y | ø | | | | day[0] |
| 1 | M | o | n | d | a | y | ø | | | | day[1] |
| 2 | T | u | e | s | d | a | y | ø | | | day[2] |
| 3 | W | e | d | n | e | s | d | a | y | ø | day[3] |
| 4 | T | h | u | r | s | d | a | y | ø | | day[4] |
| 5 | F | r | i | d | a | y | ø | | | | day[5] |
| 6 | S | a | t | u | r | d | a | y | ø | | day[6] |

To access an individual string, we simply specify only the row index.

For example:

```
cout << day[1];
```

**Example:** Use array of string to write a C++ program that prints the week days.

```
#include <iostream.h>
void main()
{
  char day[7][10] = { "Sunday" , "Monday",
                      "Tuesday" , "Wednesday",
                      "Thursday", "Friday",
                      "Saturday" };
  for(int i=0 ; i<7  ; i++)
    cout<<day[i]<<endl;
}
```

**Example:** Write a C++ program that initializes 5 names into array of strings. Then the program reads a name and determines whether it is found in the array or not.

```
#include <iostream.h>
#include <string.h>
void main()
{
  char name[5][10] = { "ahmed" , "mohammad" ,
                       "hamdy" , "samy"   ,
                       "nabil"    };
  char myname[10];
  cout<<"Enter your name: "<<endl;
  cin>>myname;
  for(int i=0 ; i<5  ; i++)
    if(strcmp(myname,name[i]) == 0)
      {
         cout<<"Your name is found.";
         break;
      }
      else
        cout<<(i==4 ? "Your name is not found.":"");
}
```

## Homework:

1. Write a C++ program that inputs an integer array of 10 elements and prints only the prime numbers in the array.
2. Write a C++ program that reads an integer array a[10] and finds the max value with its position and the min value with its position.
3. Write a C++ program that inputs an integer array b[10] and then reverse it and print the reversed array.
4. Write a C++ program that exchanges the primary and secondary diagonals of 4×4 matrix.
5. Write a C++ program that converts a two dimensional array into one dimensional array. Then print the 1D array.

6. Write a C++ program that creates the following matrix:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

7. Write a C++ program that creates the following matrix:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

8. Write a C++ program that finds the transpose of the following matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix} \longrightarrow A^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

9. Write a C++ program that computes the sum of the secondary diagonal elements in a square integer matrix.

10. Write a C++ program that inputs a 4×4 matrix and then exchanges the upper triangle above the main diagonal with the respect lower triangle.