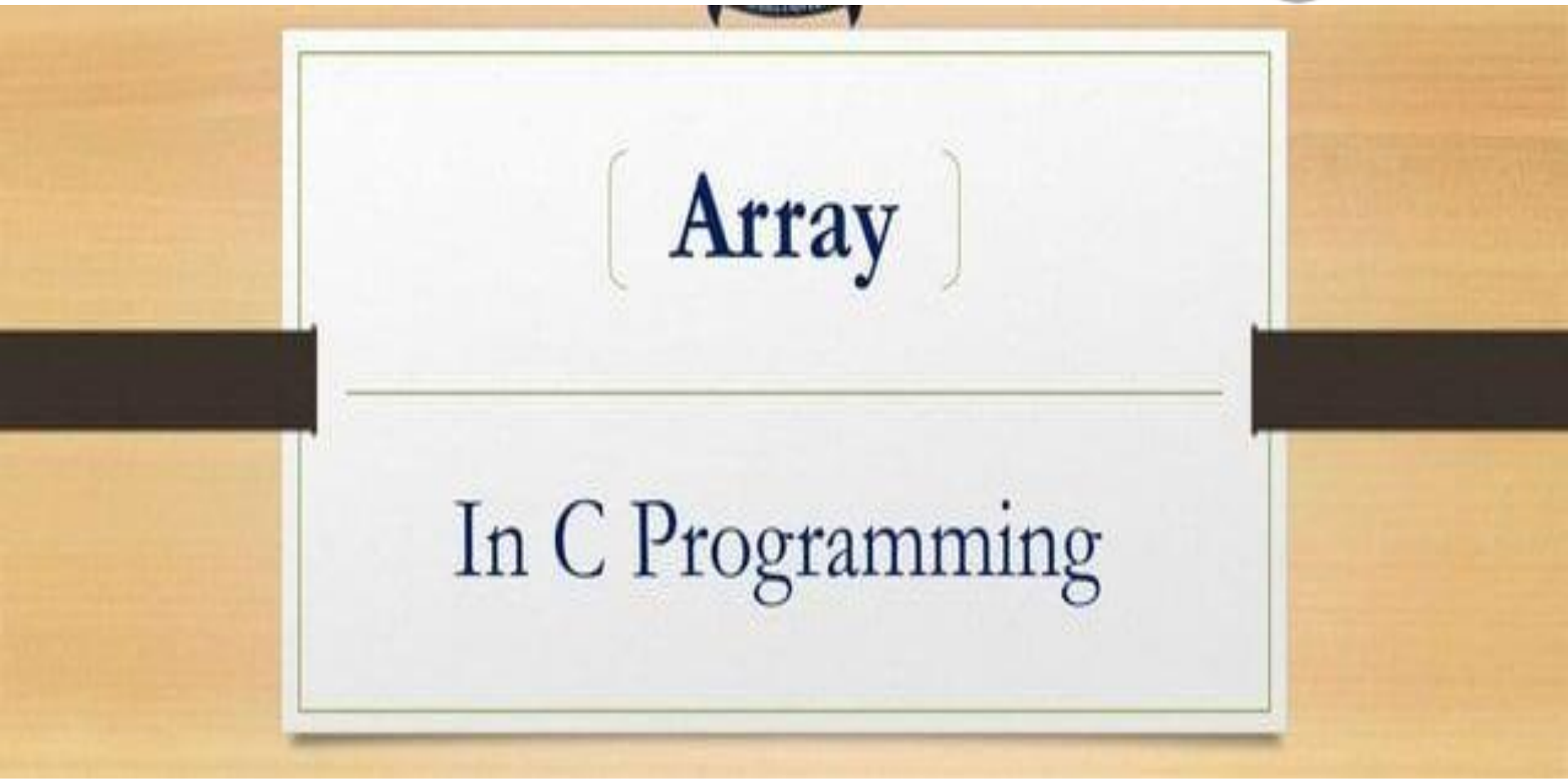

Array
In C Programming

Array

C Programming

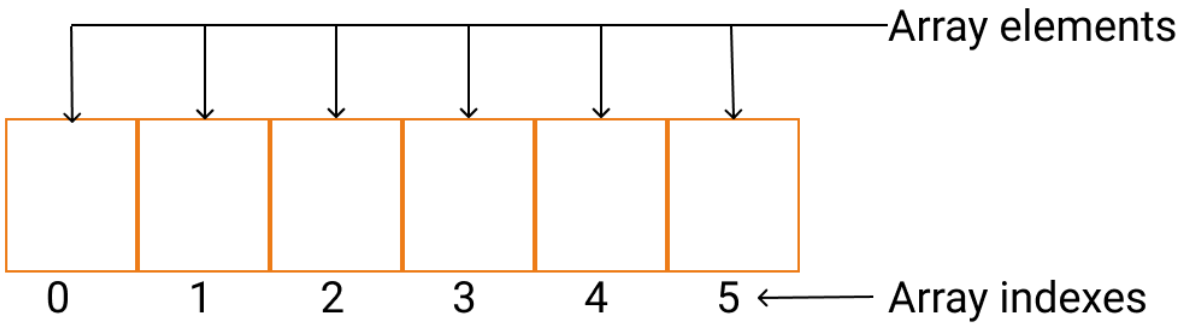
- Array is a data structure that represents a collection of the same types of data.
- Num reference → int num[5];

num [0]
num[1]
num[2]
num [3]
num[4]
num[5]

An Array of 5 Elements of type int.

Memory Representation of array

faceprep



Array Declaration

C Programming

- Syntax:

  `<type> <arrayName>[<array_size>]`

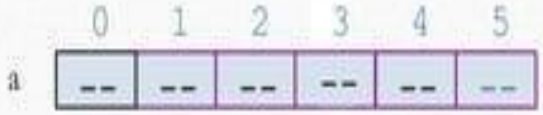
  `Ex. int a[6];`

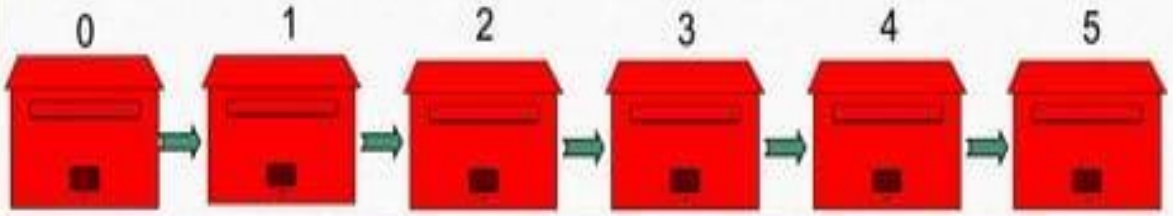
- The array elements are all values of the type `<type>`.
- The size of the array is indicated by `<array_size>`, the number of elements in the array.
- `<array_size>` must be an `int` constant or a constant expression. Note that an array can have multiple dimensions.

Array Declaration(continued)

```
// array of 6
int a[6];
```

- Array index
- Collection of 6 variable
- 1 dimensional Array
- Datatype
- Array name





- Declaring, creating, initializing in one step:

```
int my Array[5] = {1, 2, 3, 4, 5};
  int studentAge[4];
      studentAge[0] = 14;
      studentAge[1] = 13;
      studentAge[2] = 15;
      studentAge[3] = 16;
```

*Initializing Arrays

Initialization of 1D array:

- Should be initialized as other variable .

Examples:

-
```
int array [6]={1,3,5,7,9,8};
int array [5]={3,5};
int array [ ]={2,4,6,7};
```

Note: We can't initialized an array using a variable

- **For example:**

```
int x=5;
int array [x];
```

Above example is illegal.

Read and Print Elements of an Array

Problem Statement:

Write a C program to read `n` elements into an array and print them.

Example:

Input: `1 2 3 4 5`

Output: `Array elements are: 1 2 3 4 5`

Problem Statement:

Write a C program to find the sum and average of 5 marks entered by user using an array.

Example:

Input: `5` marks → `10 20 30 40 50`

Output: `Sum = 150, Average = 30`

Problem Statement:
Write a C program to create an array of size 5 and initialize at time of declaration.
Print the elements of an array in reverse order.
Example:
Input: `1 2 3 4 5`
Output: `5 4 3 2 1`

Write a C program to count how many numbers in an array are even and how many are odd.
Example:
Input: `1 2 3 4 5 6`
Output: `Even = 3, Odd = 3`

Two dimensional array

- A two dimensional array has two subscripts/indexes. The first subscript refers to the row, and the second, to the column.
- Its declaration has the following form,

```
data_type    array_name[1st dimension size][2nd
   dimension size];
```

- For examples,

```
int      xInteger[3][4];
float    matrixNum[20][25];
```

Declaration Of Two-dimensional Arrays :

The general form of two dimensional array declaration is :

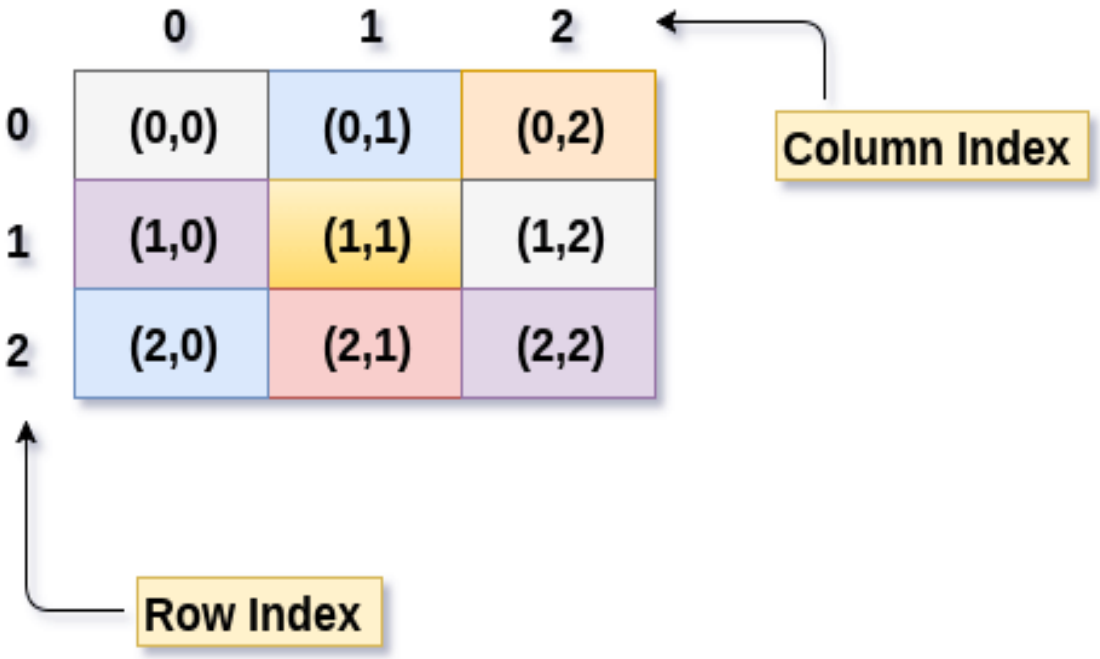
Data_type *array-name[row_size][column_size];*

Here the Data_type specifies the type of elements contained in the array, such as int, float, or char.

array_name is a valid C identifier.

The size should be either a numeric constant or a symbolic constant.

Memory representation of arrays(2-D)



2) Two dimensional array:

- Simplest form of multi dimensional array .
- Requires two pair of square brackets.
- **Syntax:**

  To declare 2D array of size x , y.

  ```
  data_type array_name[x][y];
  ```

- First bracket is know as number of rows and second numbers of columns.

Initialization of 2D array:

- 2D array is initialized by specifying bracket value of each value .
- Following is an array with 3 rows and 4 columns:

```
int a[3][4]={    {0,1,2,3},
                 {4,5,6,7},
                 {8,9,4,7}     };
```

- Also equivalent to

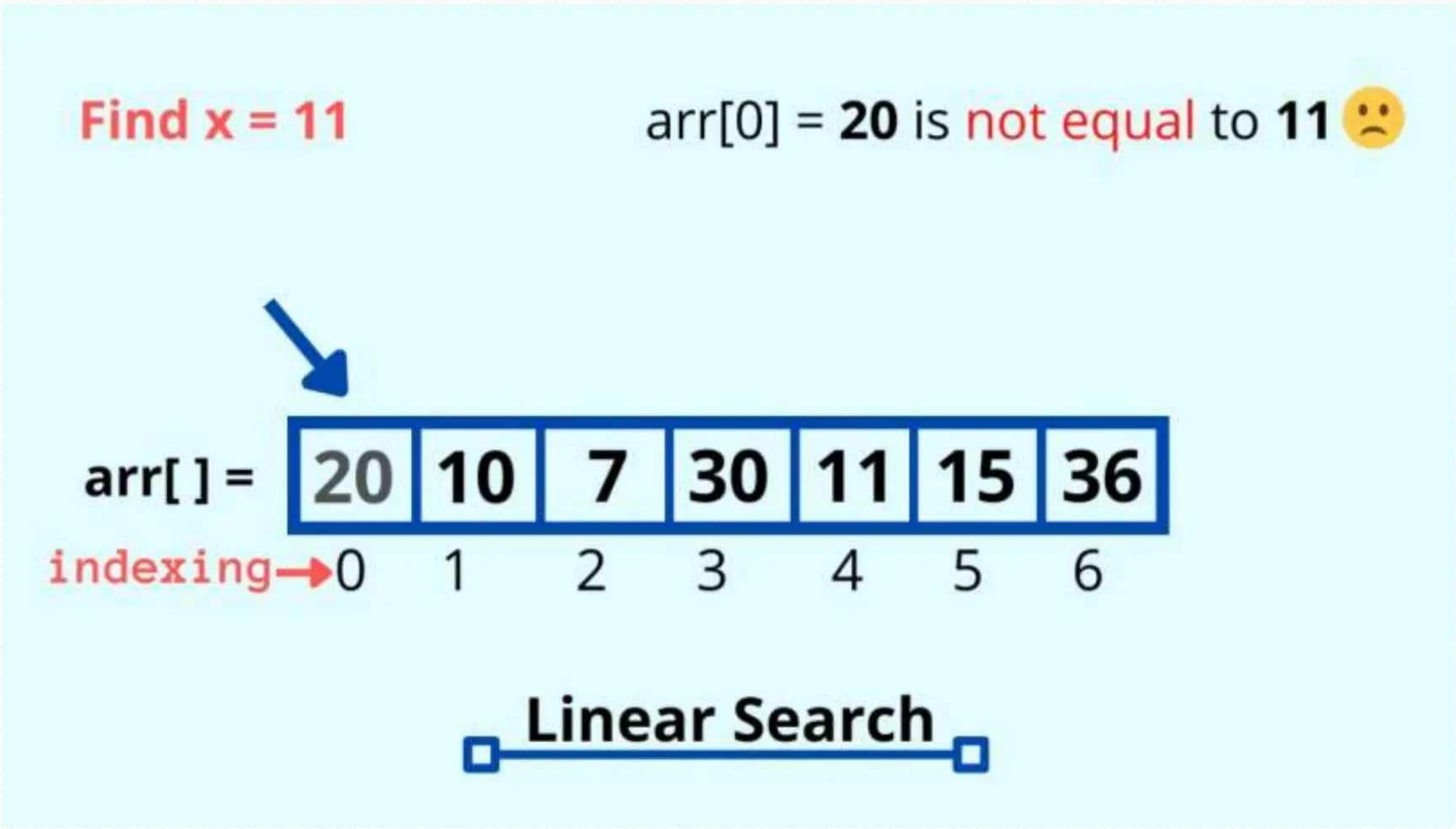
```
int a[3][4]={0,1,2,3,4,5,6,7,8,9,4,7};
```

LINEAR SEARCH

A linear search is also known as a sequential search that simply scans each element at a time.

The linear search starts searching from the first element and compares each element with a searched element till the element is not found.

EXAMPLES


Find x = 11
arr[0] = 20 is not equal to 11
arr[ ] =
20
10
7
30
11
15
36
indexing
0
1
2
3
4
5
6
Linear Search

normal

Sum 1,2,3,4

```
#include<stdio.h>
int main()
{
int a,b,c,d;
a=1 ;
b=2 ;
c=3 ;
d=4 ;
int sum=a+b+c+d;
printf("%d",sum);
return 0;
}
```

Output

10

Array

Array

Sum 1,2,3,4

```
#include<stdio.h>
int main()
{
int num[4];
num[0]=1 ;
num[1]=2 ;
num[2]=3 ;
num[3]=4 ;
int sum=num[0]+num[1]+num[2]+num[3];
printf("%d",sum);
return 0;
}
```

Output

10

Sum from user input

```
#include<stdio.h>
int main()
{
int num[4];

scanf("%d%d%d%d",&num[0] ,&num[1] ,&num[2],&num[3]);

int sum=num[0]+num[1]+num[2]+num[3];

printf("%d",sum);
return 0;
}
```

Output

Sum from user input

```
#include<stdio.h>
int main()
{
int num[4],sum=0,i;

for(i=0; i<4; i++)
{  scanf("%d",&num[ i ]); }

for(i=0; i<4; i++)
{  sum= sum + num [ i ] ; }

printf("%d",sum);
return 0;
}
```

Output


Questions?