RED & WHITE®

Multimedia Education

Shaping "skills" for "scaling" higher...!!!

WELCOME, PROGRAMMERS

01.

What is Array?

WHAT IS ARRAY?

ARRAY

An array in C language is **a collection of elements of the same data type**.

Each element in the array is identified by an **index** or a key. The **index starts from 0** for the first element, 1 for the second, and so on.

Arrays provide a way to **store multiple values** of the same data type under a single name.

TYPES OF ARRAY

Here are types of Array:

One Dimensional Array

i.e, 1D Array

1

Multi Dimensional Array

i.e, 2D, 3D, ..., ND Array

2

Let’s see **1D Array** in detail with some examples…

1D ARRAY

A **one-dimensional array (1D array)** is a **collection of elements of the same data type** arranged in a linear sequence.

It provides a convenient way to store and manipulate multiple values under a single identifier.

The **elements** in a one-dimensional array are **accessed** using an **index**, which starts from **0** for the first element.

1D ARRAY

EXAMPLES

01

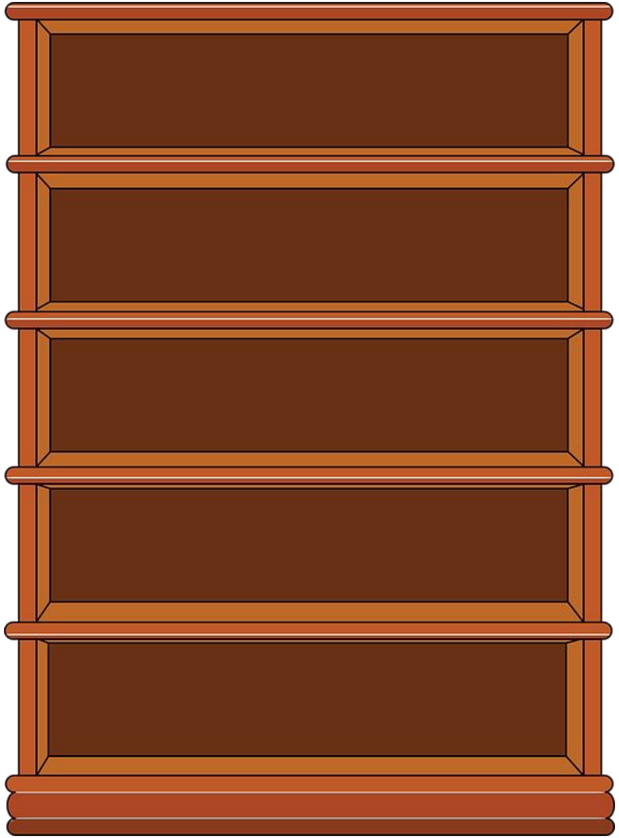
BENCH

02

TRAIN COACHES

03

BOOKSHELF



Let’s see **syntax** of **1D Array** in detail with some examples…

Syntax of 1D Array

```
datatype array_name[size];
```

ARRAY OPERATIONS

There are many operations can be perform on an array. But, here are the **most common operations** of Array:

Insertion

1

Iteration

2

Modification / Updation

3

Let’s see a **each operations** in detail…

01

Insertion Operation

Insertion Operation

	Elements					
int a[5] = {	6,	4,	8,	9,	3	};
Index / Position	0	1	2	3	4	

Predefined Array

Insertion Operation

	Elements				
int a[5]; // Empty Array	0	0	0	0	0
Index / Position	0	1	2	3	4

Empty Array

Insertion Operation

```
int a[5];

a[0] = 6;
```

	Elements				
int a[5];	6	0	0	0	0
Index / Position	0	1	2	3	4

Index-wise static insertion

Insertion Operation

```
int a[5];

a[0] = 6;
a[1] = 9;
```

	Elements				
int a[5];	6	9	0	0	0
Index / Position	0	1	2	3	4

Index-wise static insertion

Insertion Operation

```
int a[5];

a[0] = 6;
a[1] = 9;
a[2] = 5;
```

	Elements				
int a[5];	6	9	5	0	0
Index / Position	0	1	2	3	4

Index-wise static insertion

RED & WHITE®
Multimedia Education
Shaping "skills" for "scaling" higher...!!!

Insertion Operation

```
int a[5];

a[0] = 6;
a[1] = 9;
a[2] = 5;
a[3] = 2;
```

	Elements				
int a[5];	6	9	5	2	0
Index / Position	0	1	2	3	4

Index-wise static insertion

Insertion Operation

```
int a[5];

a[0] = 6;
a[1] = 9;
a[2] = 5;
a[3] = 2;
a[4] = 7;
```

	Elements				
int a[5];	6	9	5	2	7
Index / Position	0	1	2	3	4

Index-wise static insertion

Insertion Operation

	Elements				

`int a[5];`
`// Empty Array`
	0	0	0	0	0

Index / Position	0	1	2	3	4

Empty Array

Insertion Operation

```
int a[5];

scanf(“%d”, &a[0]); // 7
```

	Elements				
int a[5];	7	0	0	0	0
Index / Position	0	1	2	3	4

Index-wise dynamic insertion

Insertion Operation

```
int a[5];

scanf(“%d”, &a[0]); // 7
scanf(“%d”, &a[1]); // 3
```

	Elements				
int a[5];	7	3	0	0	0
Index / Position	0	1	2	3	4

Index-wise dynamic insertion

Insertion Operation

```
int a[5];

scanf(“%d”, &a[0]); // 7
scanf(“%d”, &a[1]); // 3
scanf(“%d”, &a[2]); // 4
```

	Elements				
int a[5];	7	3	4	0	0
Index / Position	0	1	2	3	4

Index-wise dynamic insertion

Insertion Operation

```
int a[5];

scanf("%d", &a[0]); // 7
scanf("%d", &a[1]); // 3
scanf("%d", &a[2]); // 4
scanf("%d", &a[3]); // 9
```

	Elements				
int a[5];	7	3	4	9	0
Index / Position	0	1	2	3	4

Index-wise dynamic insertion

Insertion Operation

```
int a[5];

scanf(“%d”, &a[0]); // 7
scanf(“%d”, &a[1]); // 3
scanf(“%d”, &a[2]); // 4
scanf(“%d”, &a[3]); // 9
scanf(“%d”, &a[4]); // 2
```

	Elements				
int a[5];	7	3	4	9	2
Index / Position	0	1	2	3	4

Index-wise dynamic insertion

RED & WHITE®
Multimedia Education
Shaping "skills" for "scaling" higher...!!!

02

Iteration Operation

Iteration Operation

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

printf(“%d”, a[0]); // 7
printf(“%d”, a[1]); // 3
printf(“%d”, a[2]); // 4
printf(“%d”, a[3]); // 9
printf(“%d”, a[4]); // 2
```

	Elements				
int a[5];	7	3	4	9	2
Index / Position	0	1	2	3	4

Index-wise static accessing of elements

Iteration Operation

```
int a[5] = {7, 3, 4, 9, 2};
int i;

for(i=0; i<=4; i++)
{
    printf(“%d ”, a[i]);
}
```

	Elements				
int a[5];	7	3	4	9	2
Index / Position	0	1	2	3	4

Index-wise dynamic accessing of elements

03

Modification/Updation Operation

Updation Operation

	Elements					
int a[5] = {	7,	3,	4,	9,	2	};
Index / Position	0	1	2	3	4	

Predefined Array

Updation Operation

```
a[3] = 6;
```

	Elements				
int a[5];	7	3	4	6	2
Index / Position	0	1	2	3	4

Index-wise static updation

Updation Operation

```
scanf(“%d”, &a[1]); // 5
```

	Elements				
int a[5];	7	5	4	6	2
Index / Position	0	1	2	3	4

Index-wise dynamic updation

C LANGUAGE

Let’s start now...

▶