





WELCOME TO MY
PRESENTATION

Kavitha's presentation

DATA STRUCTURES

IN **C** LANGUAGE



OBJECTIVES..

- *What is data structure ? 🧠*
- *Why we have to learn ??*
- *Advantages and applications*
- *Types of data structures..*
- *Example*

DATA STRUCTURES

**IN Computer science, a data structure is a data organization, management, and storage format that enables efficient access and modification. More precisely, a data structure is a collection of data values.

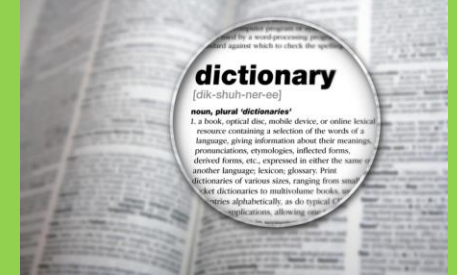
Simply,

To structure the information while storing!

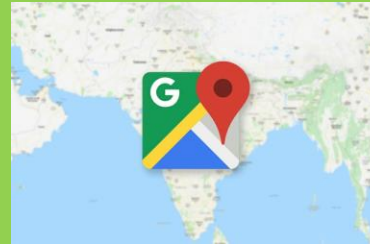
*IT IS NOT A LANGUAGE, IT'S A CONCEPT OF "SET OF ALGORITHMS" USED TO STRUCTURE THE INFORMATION.

ADVANTAGES AND REAL LIFE APPLICATIONS

- Data structures allow information storage on hard disks
- These are necessary for designing of efficient algorithms
- Allows safe storage of information on a computer
- Data structures provides means for management of large data set such as data bases or internet indexing services..

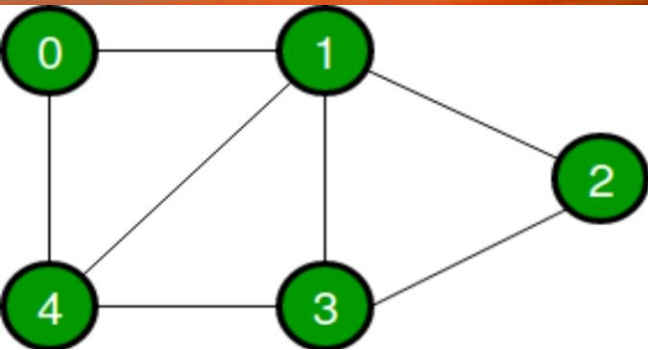


- IN REAL LIFE :



- To store a set of fixed key words which are referenced very frequently.
- To store the customer order information in a drive in burger place.
- To implement back functionality in internet browser.
- To store the genealogy information of biological species .

TYPES OF DATA STRUCTURES..



Data Structure

**Linear data
Structure**

Array

Stack

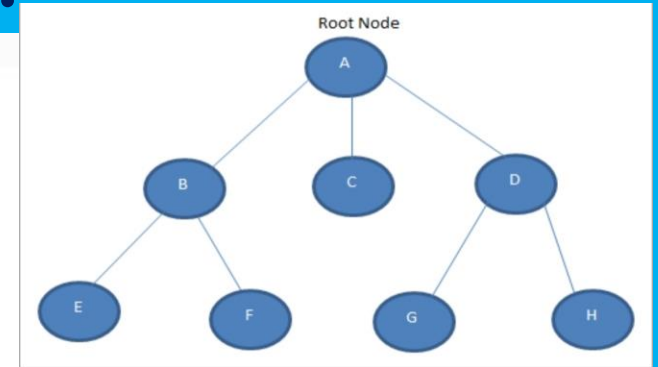
Queue

**Linked-
list**

**Non-linear data
Structure**

Graphs

Trees

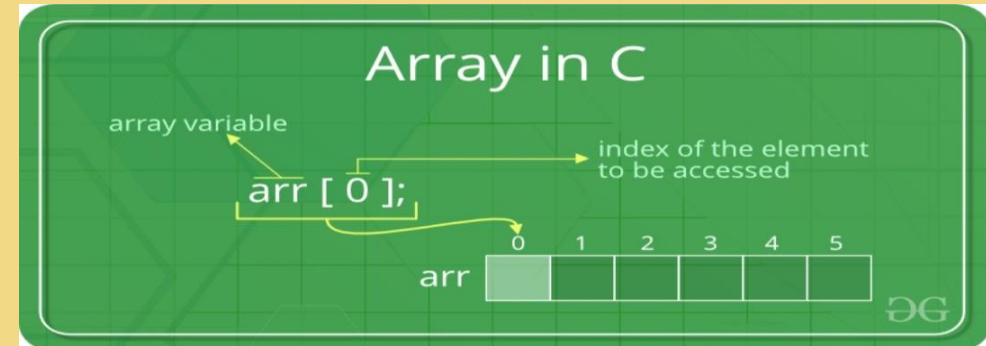


👍 **ARRAY** : An array is a collection of similar datatype, accessed using a common name .

A one dimensional array is like a list

A two dimensional array is like a table.

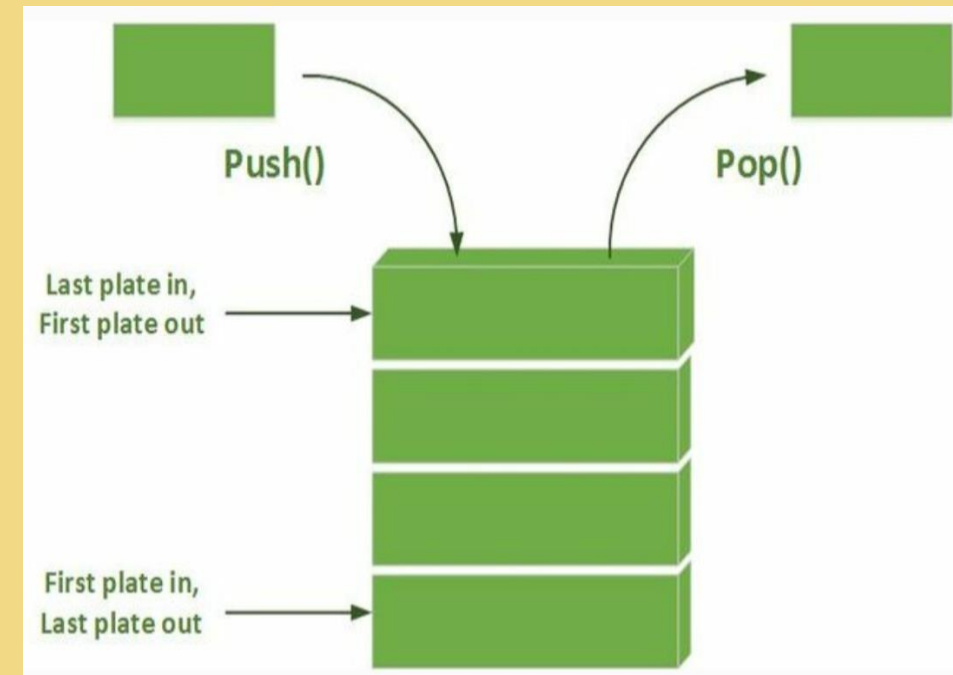
In c, no limits on the number of Dimensions in an array



👍 **STACK** : A stack is a container of objects that are inserted and removed according to the **last in first out [LIFO]** Principle .

“A stack is a limited access data structure elements can be added and removed from the stack only at the top. “

Push adds an item to the top of the stack
pop removes the item from the top.

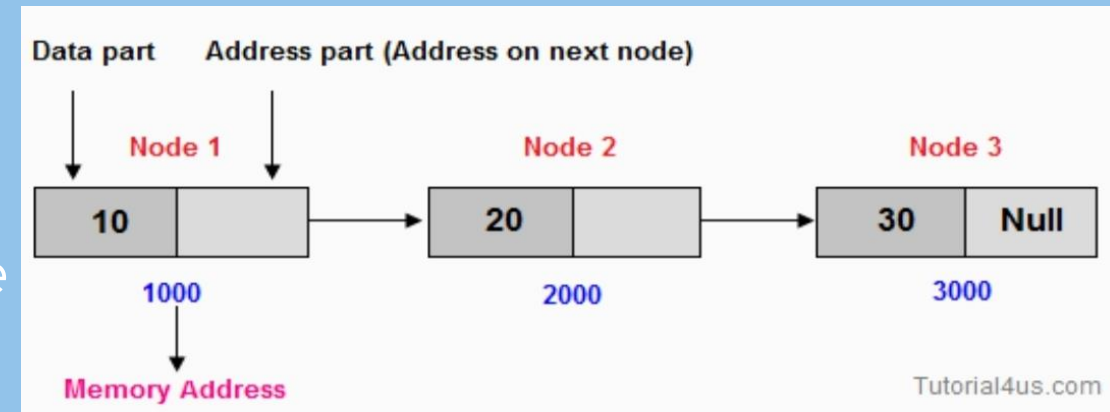


- **Linked list** : A linked list is a sequence of data structures, which are connected together via links . Linked list is a sequence of links which contains items.

- * Each link contains a connection to another link.

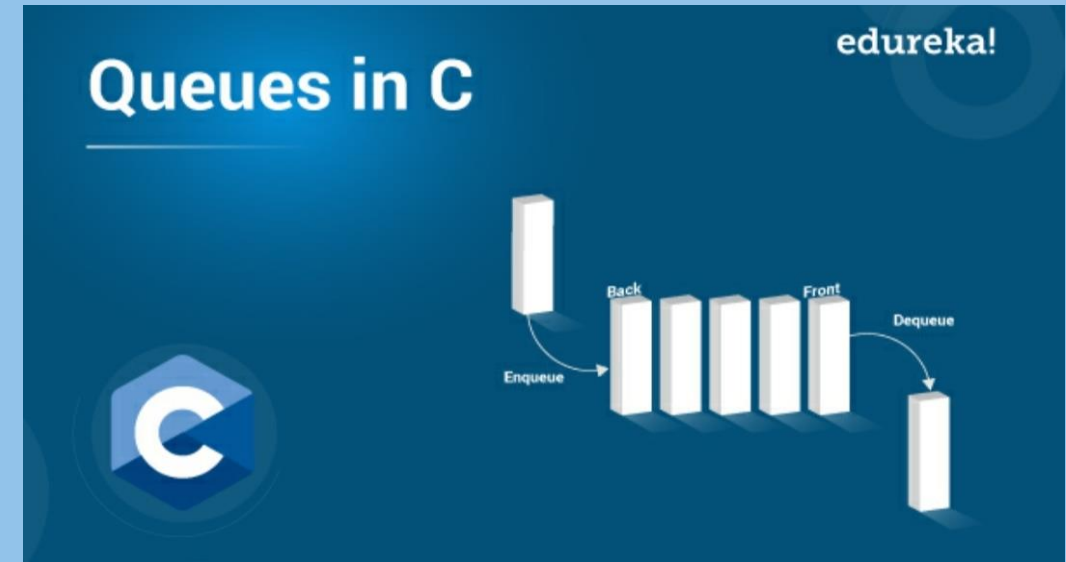
- * Its a dynamic memory allocation collection type

- * There is no fixed size for linked list .



QUEUES : A queue is a linear data structure That stores a collection of elements.

The queue operates on *FIRST IN FIRST OUT* [*FIFO*] Algorithm.



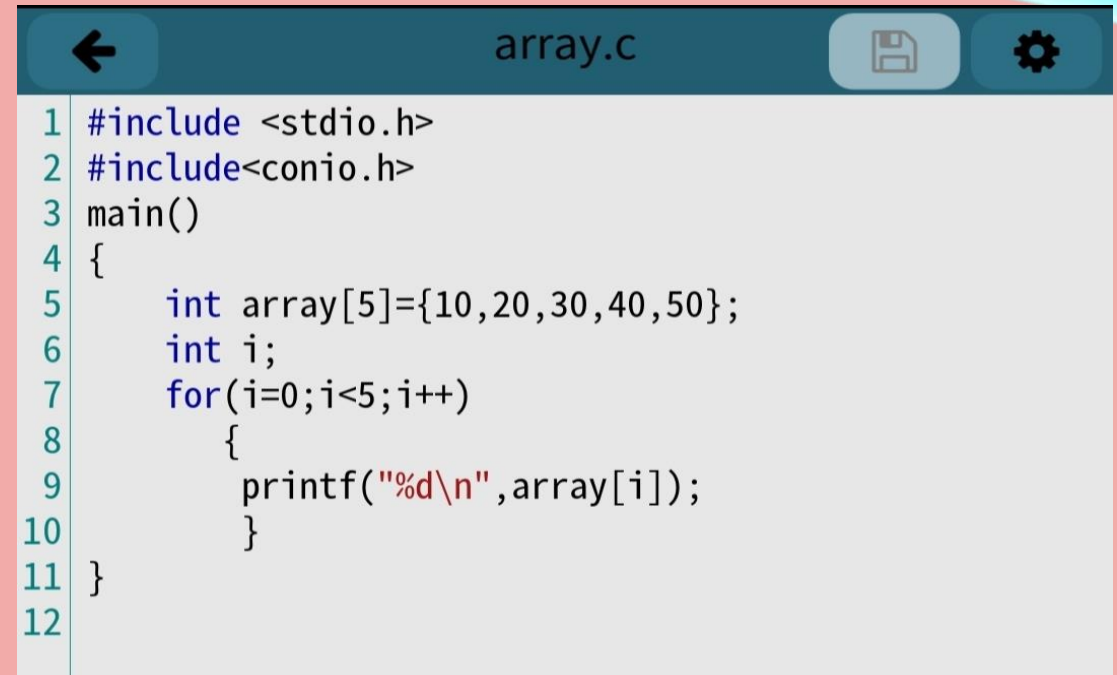
A red banner with a white border and a white arrow pointing right, containing the word 'EXAMPLE' in white capital letters.

EXAMPLE

- **ARRAY DATA STRUCTURE :** *An Array is a collection of elements of the same data type placed in contiguous memory locations that can be individually referenced by using an index to a unique identifier .*
- **Applications :**
 - *To store elements of same data .*
 - *Used for maintaining multiple variable names using single name .*
 - *Used for sorting elements .*
 - *Used in CPU scheduling [queue] .*
 - *Can be used in recursive function .*

PROGRAM :

```
1. #include<stdio.h>
2. #include<conio.h>
3. Main()
4. {
5.     Int array[5]={10, 20,30,40,50} ;
6.     Int i ;
7.     For ( i=0 ; i<5 ;i++ )
8.     {
9.         Printf("%d\n", array[ i ]) ;
10.    }
11. }
```



```
array.c
1 #include <stdio.h>
2 #include<conio.h>
3 main()
4 {
5     int array[5]={10,20,30,40,50};
6     int i;
7     for(i=0;i<5;i++)
8     {
9         printf("%d\n",array[i]);
10    }
11 }
12
```



```
array
10
20
30
40
50
```


EXAMPLE 2 :

```
1.  /*sum of all array elements */
2.  #include<stdio.h>
3.  #include<conio.h>
4.  Main()
5.  {
6.      Int array[ 6 ]= { 10 , 9, 8 , 7 , 6 , 5 } ;
7.      Int i , sum=0 ;
8.      for( i=0 ; i<6 ; i++ )
9.          {
10.             Sum = sum + array[ i ] ;
11.          }
12.      Printf ("sum%d\n" , sum) ;
13. }
```

```
← sum of array elements.c [Save] [Settings]
1 #include <stdio.h>
2 #include<conio.h>
3 main()
4 {
5     int array[6]={10,9,8,7,6,5};
6     int i,sum = 0 ;
7     for(i=0 ; i<5 ; i++)
8     {
9         sum = sum + array[i];
10    }
11    printf("sum is %d\n",sum);
12 }
13
```

```
← sum of array elements [Settings]
sum is 40
```



