Linked List

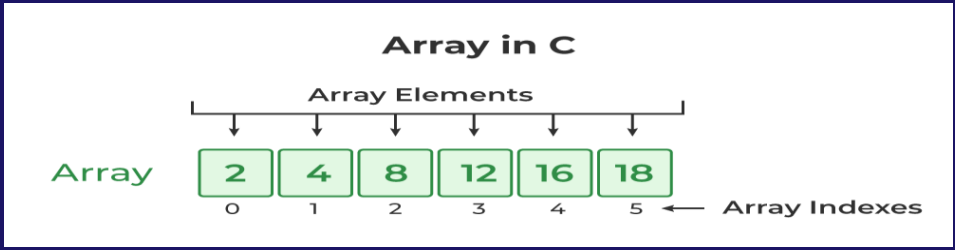
linked list is a linear data structure that includes a series of connected nodes

Prepared By-
Ifty Imam Bin Razzak
ID-2101014

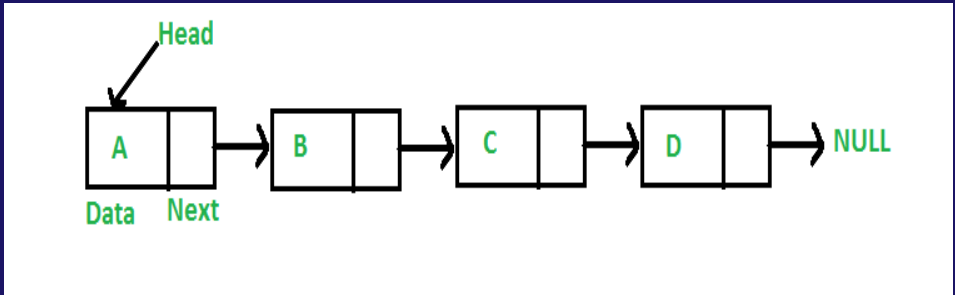
Ways to maintains a list in memory

Two Ways

Array



Linked List



Array

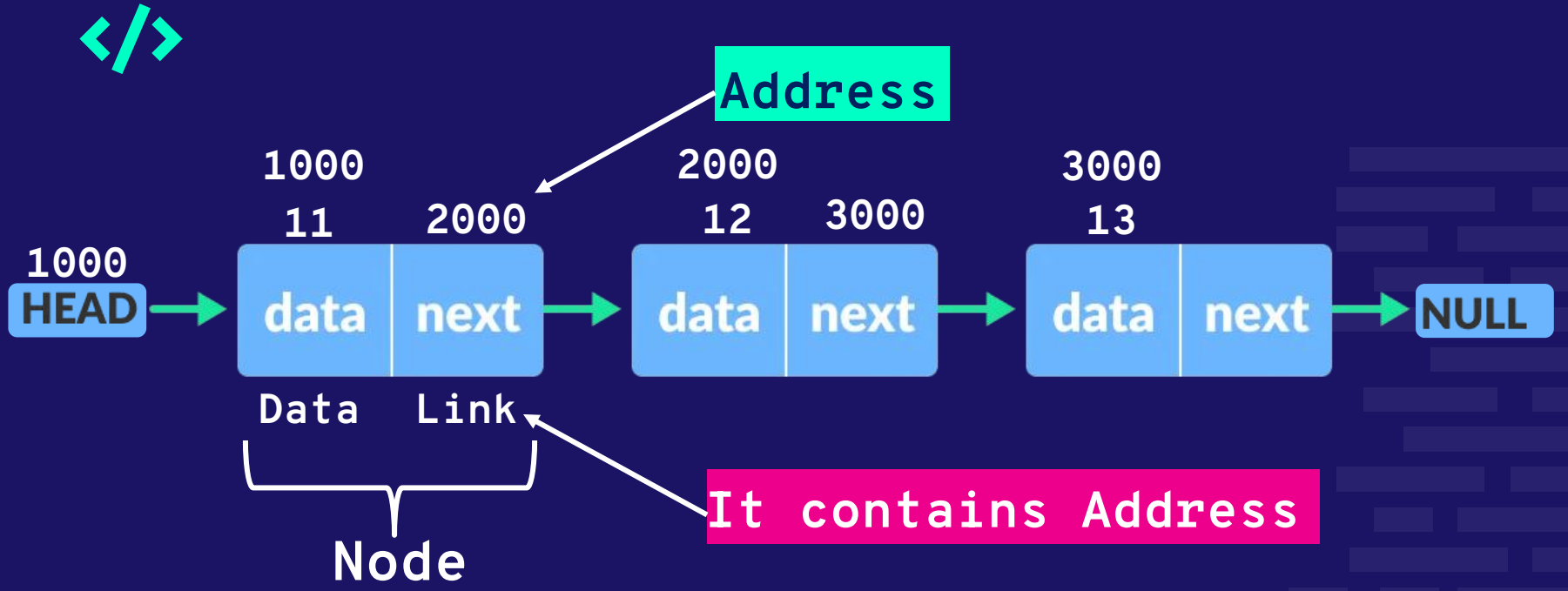
Array is the sequential representation of the list

Linked List

Linked List is the linked representation of the list

Linked List

A linked list is a linear data structure that includes a series of connected nodes. Here, each node stores the **data** and the **address** of the next node.


Address
1000
11
2000
2000
12
3000
3000
13
1000
HEAD
data
next
data
next
data
next
NULL
Data
Link
Node
It contains Address

Types of Linked Lists:

Simple Linked List

In this type of linked list, one can move or traverse the linked list in only one direction. where the next pointer of each node points to other nodes but the next pointer of the last node points to NULL. It is also called “Singly Linked List”.

Doubly Linked List

Will be explained by @Sifat Rayhan

Circular Linked List

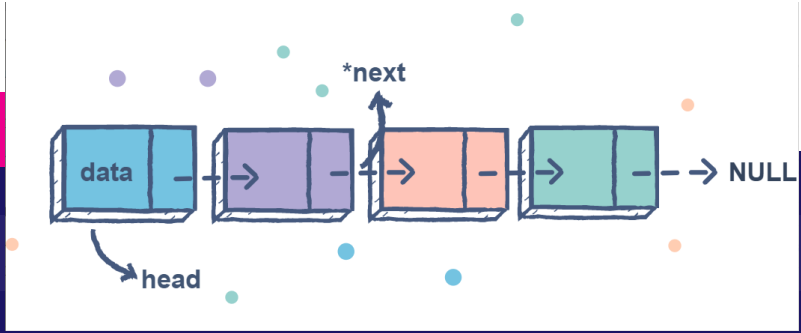
Will be explained by @Sifat Rayhan

Doubly Circular Linked List

Header Linked List

singly linked list

A singly linked list is a type of linked list that is unidirectional, that is, it can be traversed in only one direction from head to the last node (tail).



Before entering the deep of the Linked List we need some basic knowledge about Structure & Pointer

Structure

In C programming, a struct (or structure) is a collection of variables (can be of different types) under a single name.

```
struct Person {
  char name[50];
  int citNo;
  float salary;
};
```

Pointer

A pointer is a derived data type in C that can store the address of other variables or a memory. We can access and manipulate the data stored in that memory location using pointers.

Syntax of C Pointers-
datatype * *pointer_name*;

Pointer: Dynamic memory allocation

Library functions are-

1.malloc()

The “malloc” or “memory allocation” method in C is used to dynamically allocate a single large block of memory with the specified size.

Syntax of malloc-ptr = (cast-type*) malloc(byte-size)
ptr=(int*) malloc(100*sizeof(int));

2.calloc()

3.free()

4.realloc()

In C programming there are 4 library functions provided under <stdlib.h> header file to facilitate dynamic memory allocation

```c
#include <stdio.h>
#include<stdlib.h>
struct Node{
    int data;
    struct Node *link;
};
int main()
{
    struct Node *a=NULL;
    struct Node *b=NULL;
    struct Node *c=NULL;
    a=(struct Node*)malloc(sizeof(struct Node));
    b=(struct Node*)malloc(sizeof(struct Node));
    c=(struct Node*)malloc(sizeof(struct Node));
    a->data=10;
    b->data=20;
    c->data=30;
    a->link=b;
    b->link=c;
    c->link=NULL;
    while(a!=0)
    {
        printf("%d ->",a->data);
        a=a->link;
    }
    return 0;
}
```

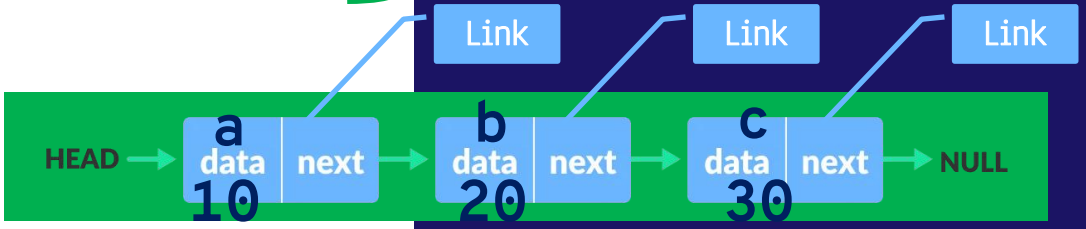
For Pointer's library function **malloc()**

Structure

An Arrow operator in C/C++ allows to access elements in Structures and Unions.

->

Library function of pointer




Thank you