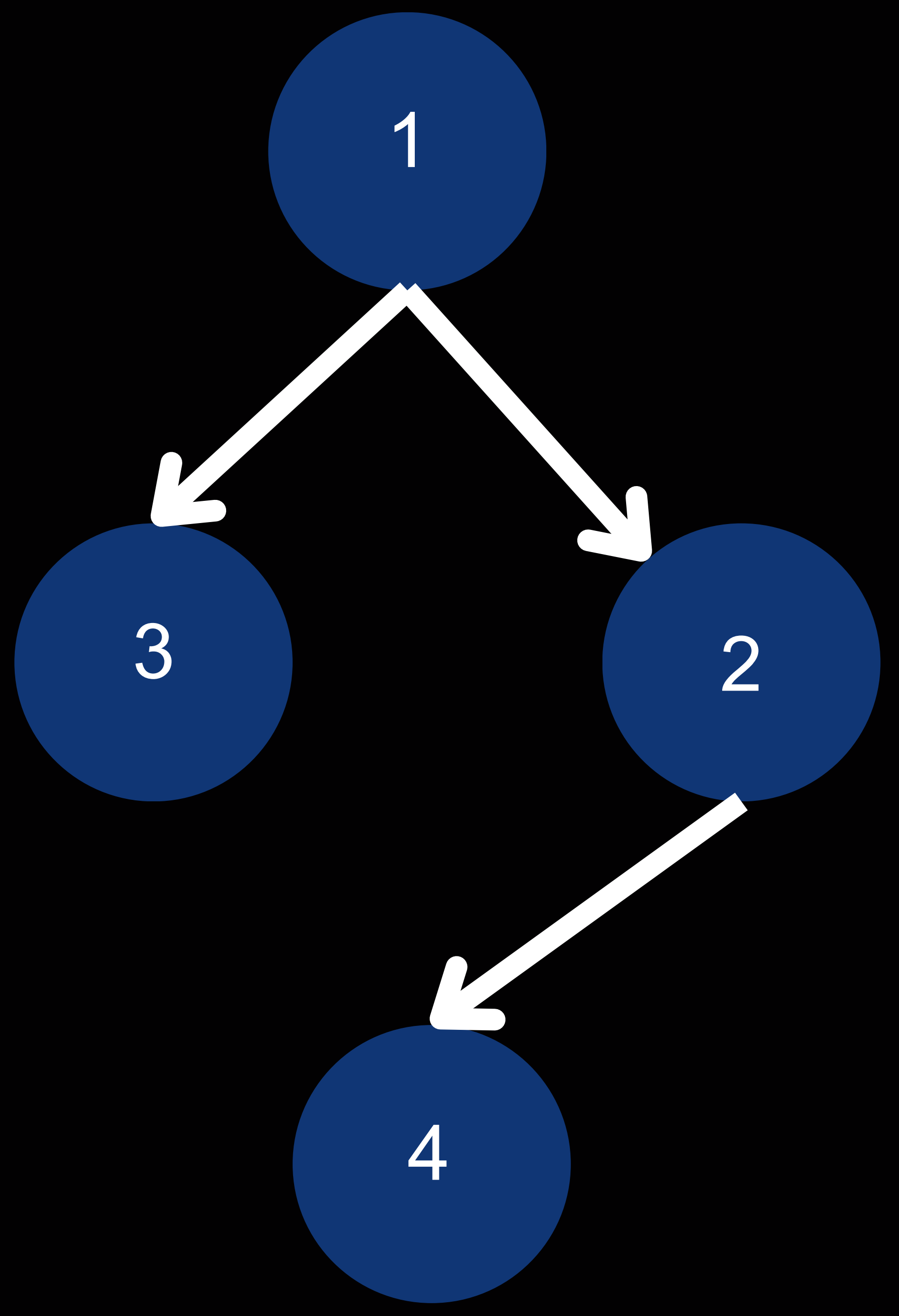
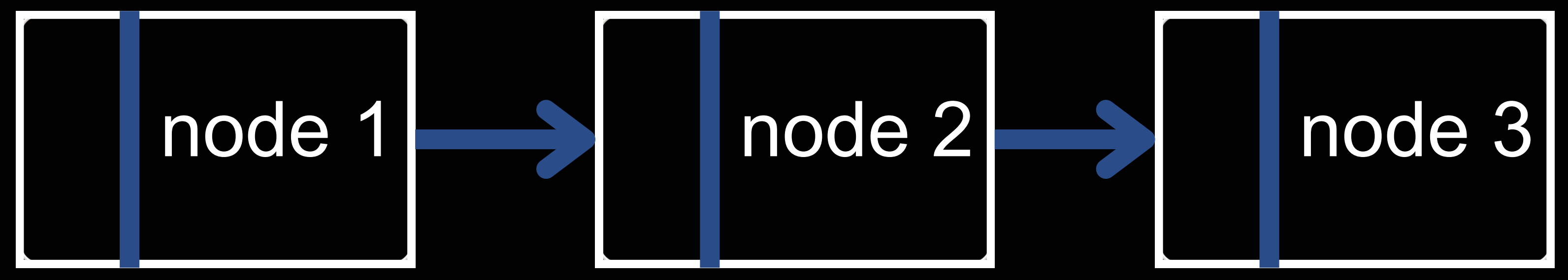
0x1A2B3C

0x4D5E6F

0x7F8A9B

DATA STRUCTURE





node 1

node 2

node 3

essammohamedst1@gmail.com

ARRAYS

First lets talk about array :

what is array 🤔 ?

- An array is a data structure that stores a collection of items, typically of the same data type
- Array stored in a contiguous block of memory
- Each item in an array can be accessed by an index or a position number

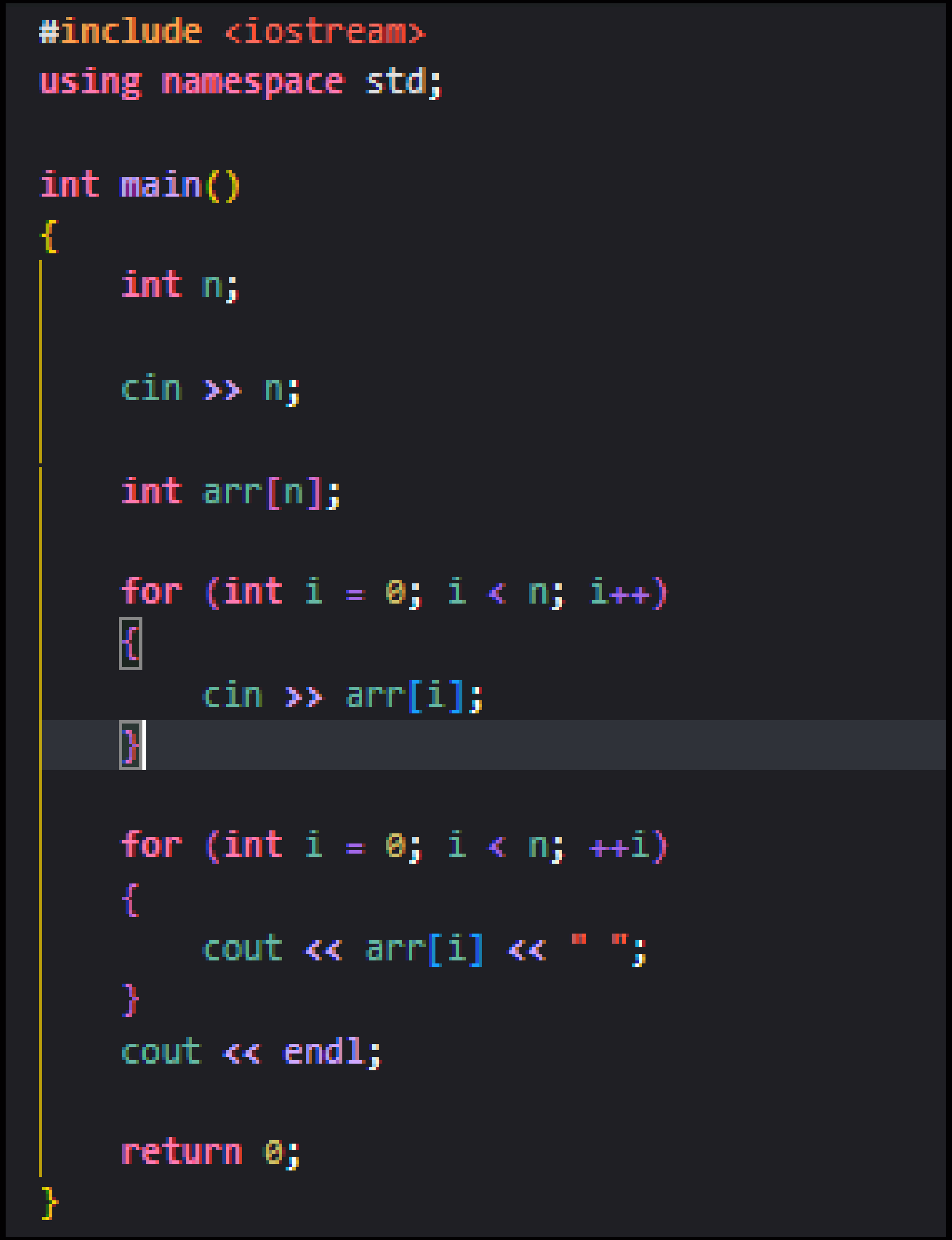
Memory layout of an array with elements [10, 20, 30, 40]:

Memory

Address	Data
0x1000	10
0x1004	20
0x1008	30
0x100C	40

$$\text{Address of element at index } i = \text{Base address} + (i \times \text{Size of each element})$$

Array Code :



```cpp
#include <iostream>
using namespace std;

int main()
{
    int n;

    cin >> n;

    int arr[n];

    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }

    for (int i = 0; i < n; ++i)
    {
        cout << arr[i] << " ";
    }
    cout << endl;

    return 0;
}
```

As we see, an array is a linear data structure

We can access all elements of the array by looping on it and using the index to access them by sequence.

LINKED LISTS

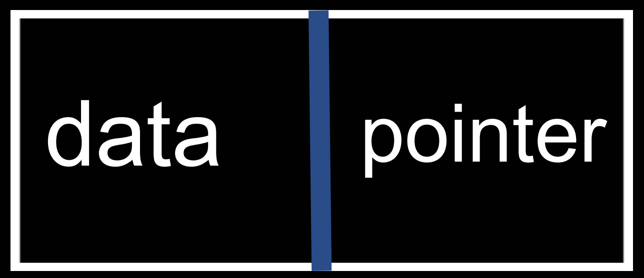
linked list is a dynamic data structure made up of nodes

what is node ?

node have two parts

- Data part: is data you want to store in the linked list like string, integers, floats, or any data.
- link part (pointer): pointer stores the address of the next node of a linked list.

node

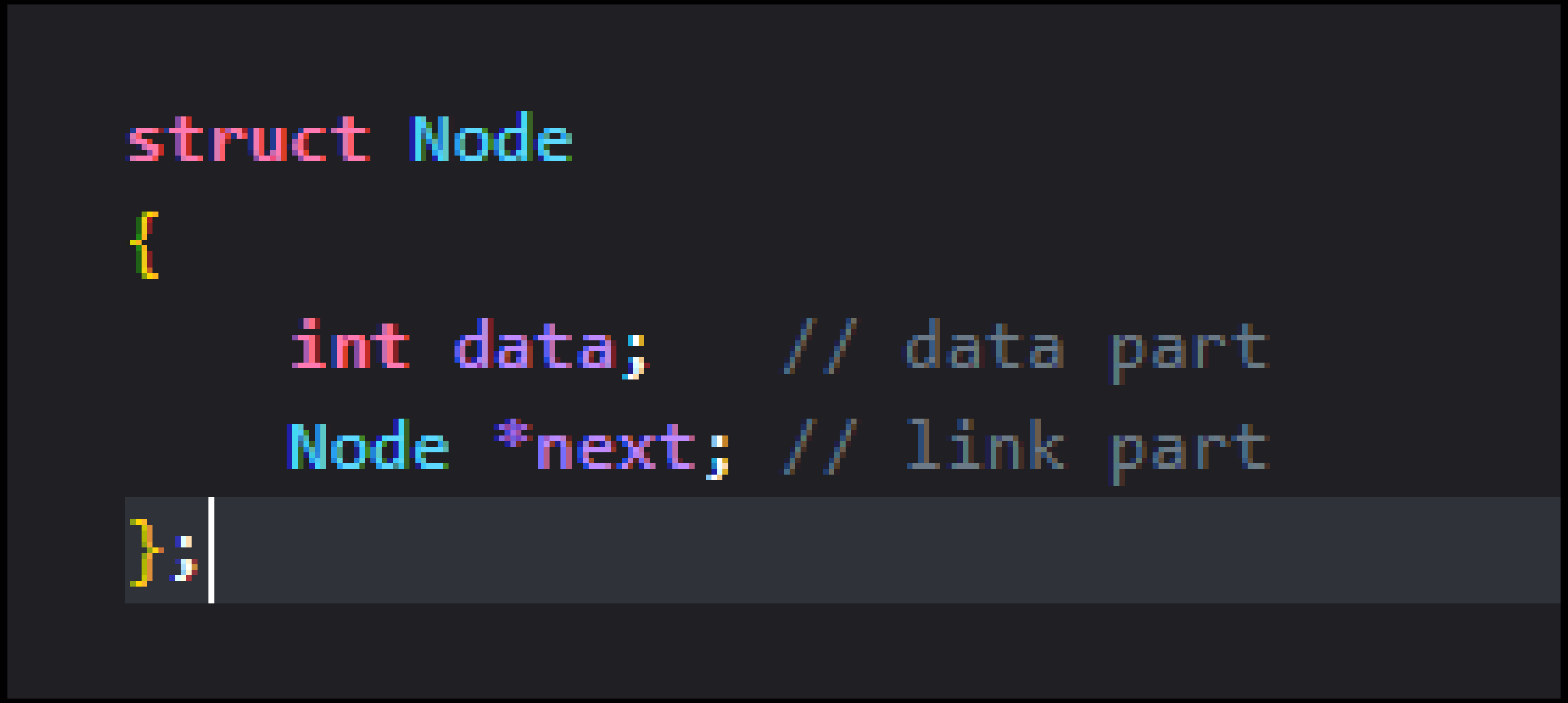


data

pointer

Linked list is some of these nodes.

Single node code:



```
struct Node
{
    int data;   // data part
    Node *next; // link part
};
```

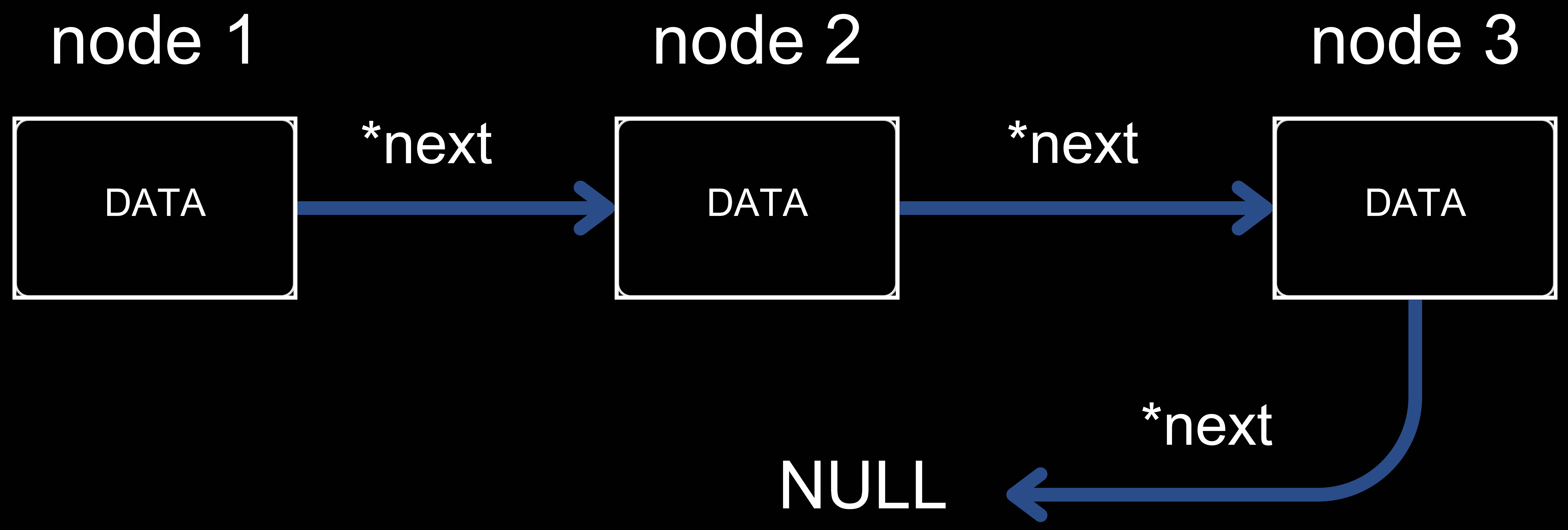
We have three main pointers. We make them point to the linked list.

As we know, a linked list is a linear data structure, but unlike an array, a linked list is not stored in contiguous memory addresses.

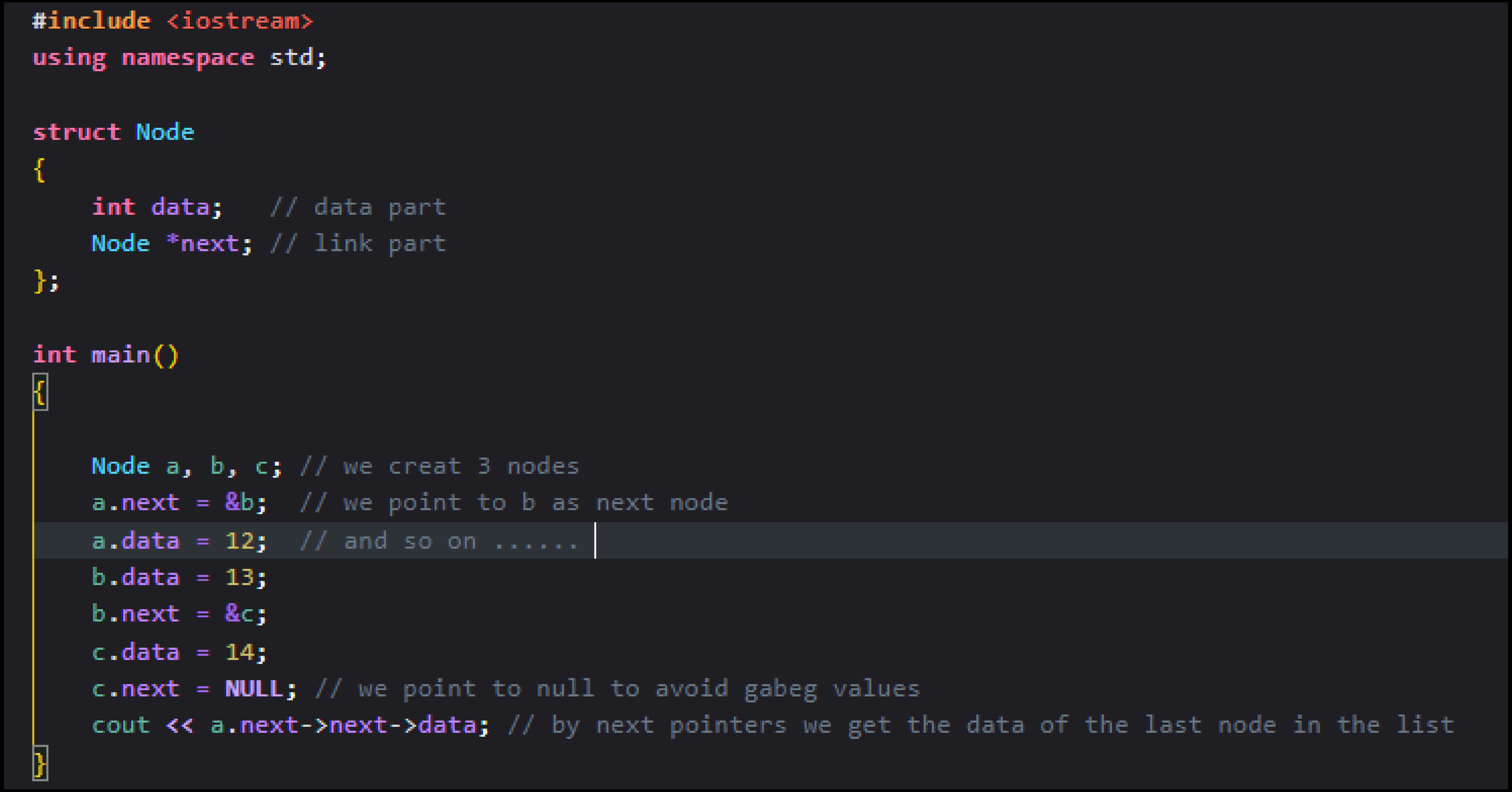
Pointers hold our linked list, and by them, we can add, delete, and search on the linked list.

- HEAD: Point to the first node of linked list
- TAIL: Point to the last node of the linked list
- CURRENT: The office boy

Linked list figure :



We have our nodes and pointers let’s make linked lsit



```
#include <iostream>
using namespace std;

struct Node
{
    int data;   // data part
    Node *next; // link part
};

int main()
{

    Node a, b, c; // we creat 3 nodes
    a.next = &b;  // we point to b as next node
    a.data = 12;  // and so on ......
    b.data = 13;
    b.next = &c;
    c.data = 14;
    c.next = NULL; // we point to null to avoid gabeg values
    cout << a.next->next->data; // by next pointers we get the data of the last node in the list
}
```

Thanks

All code and examples can be found on GitHub https://github.com/ESSAMMOHAMED1