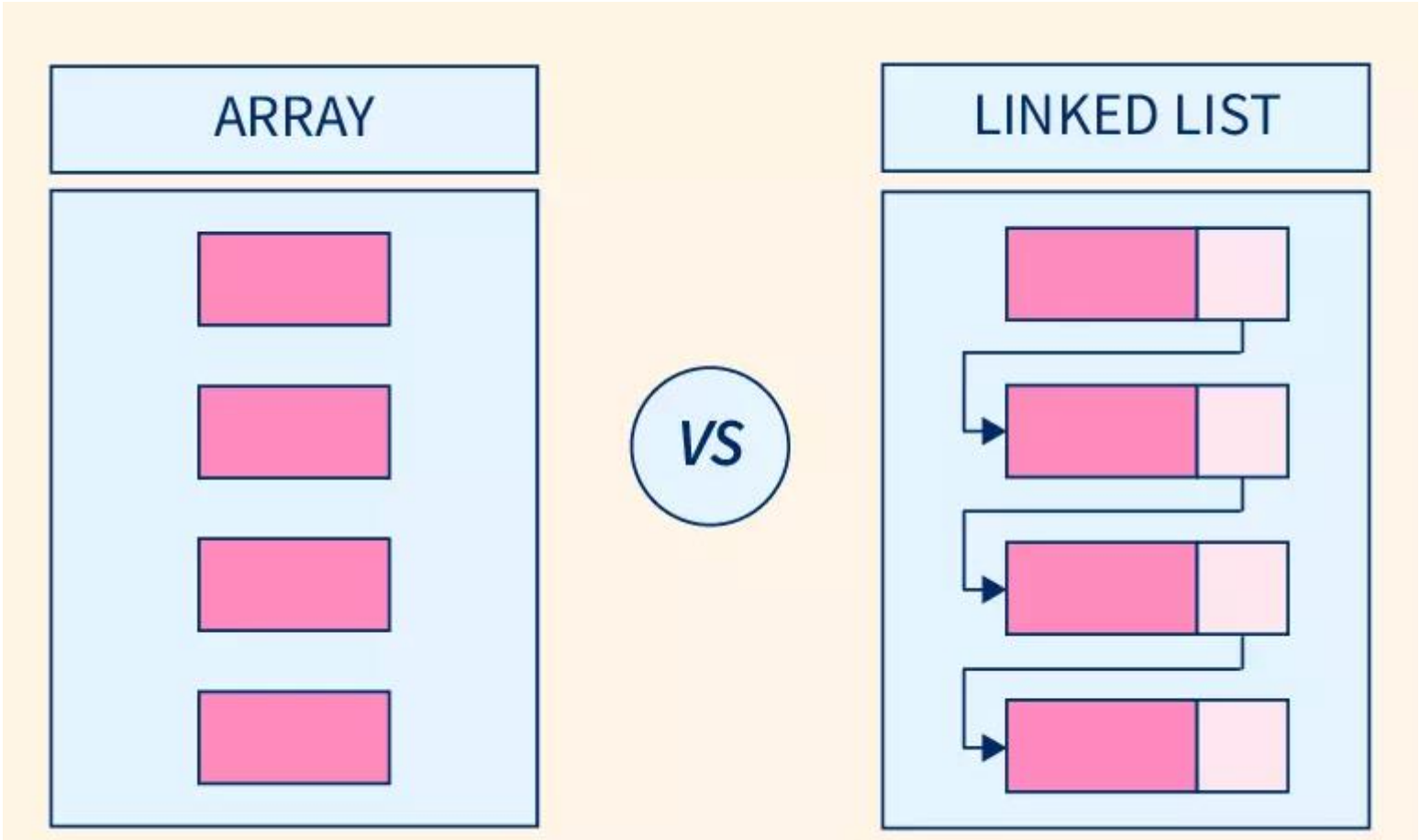
Linked list data structure in c++



Array:*Arrays store elements in contiguous memory locations, resulting in easily calculable addresses for the elements stored and this allows faster access to an element at a specific index.*

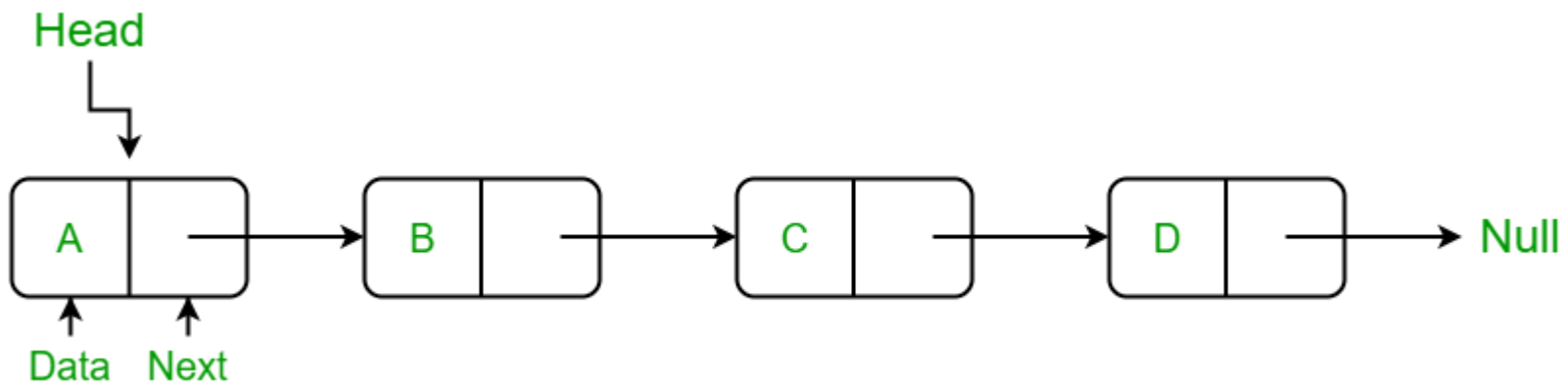
40	55	63	17	22	68	89	97	89
0	1	2	3	4	5	6	7	8

<- **Array Indices**

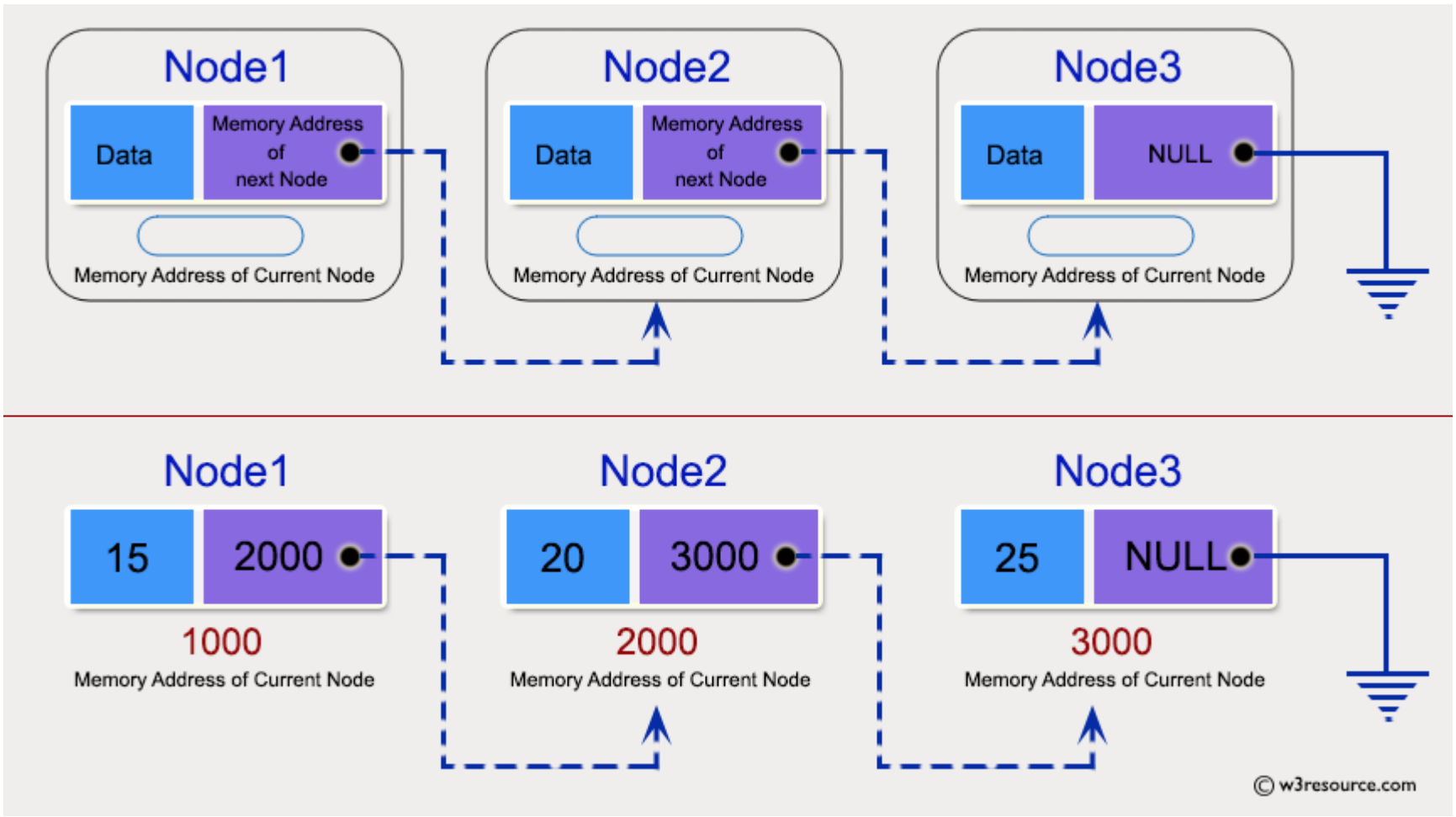
Array Length = 9
First Index = 0
Last Index = 8

But in Array, we can not ADD element or REMOVE from defined ARRAY size during compilaton time (possible but method is complex), and this is one of the big disadvantage of using Array.

Linked List: **Linked lists** are **less rigid** in their storage structure and elements are usually not stored in contiguous locations, hence they need to be stored with additional tags giving a reference to the next element.



Every Node store Value + Location of next node in linked list, here all node linked with each other using pointer / location so known as "linked" List. Easy to insert and remove element from list just need to change link between node, this is very costly in case of ARRAY.



First element from list identify using Head pointer and last element of NODE contains NULL pointer indicates end of list.

Note: Linked list is not built in type data structure, so we will have to create this if we want to use inside our code.
Linked list is user define datatype.Need to make related class and method if want to use it insid code.

ARRAY	LINKED LISTS
1. Arrays are stored in contiguous location.	1. Linked lists are not stored in contiguous location.
2. Fixed in size.	2. Dynamic in size.
3. Memory is allocated at compile time.	3. Memory is allocated at run time.
4. Uses less memory than linked lists.	4. Uses more memory because it stores both data and the address of next node.
5. Elements can be accessed easily.	5. Element accessing requires the traversal of whole linked list.
6. Insertion and deletion operation takes time.	6. Insertion and deletion operation is faster.

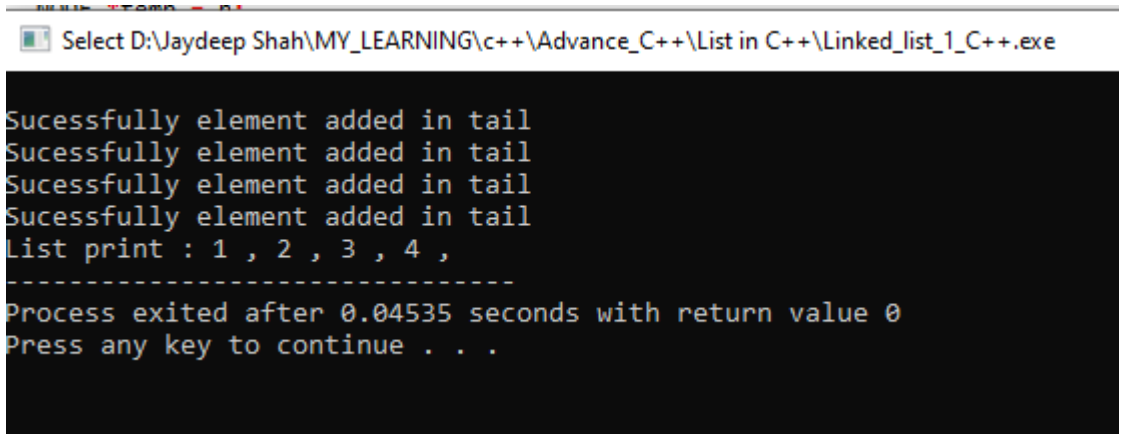
CODE:

```
//Linked list - Every node contains value + Location of the next node
//Easy to ADD / Remove element not rigid fix size structre like Array.
//Linked list is user define datatype,need to necessary class and method for operations.
//Jaydeep Shah


#include<iostream>
using namespace std;

//Create class "NODE" for storing value + ADD of next node
class NODE
{
	public:
		int     data;        //value of data
		NODE*  next;   //ADDRESS of next node

	//make constructor so during creating node object we can assign value and address of next node.

	NODE(int value)
	{
		data = value;
		next = NULL;       //initially null pointer act as last element of list
```

```
        }

};


//-------------------------------------------------------
//Create method/function for updating linked list
//Two Parameter as a argument required: 1)Value 2)Head Node
void insert_at_tail(NODE* &head,int value)
{
        //1]
        //Create temp node for storing argument value
        //This node need to place on Heap,otherwise memory will be erase after complition of function call
        NODE* temp = new NODE(value);  //We have temp node with user value

        //2]
        //If no any list created in past and passed element is first one
        if(head == NULL)
        {
                head = temp;
                return;
        }

        //3]
        //Make local variable for furthe process
        NODE *test = head;

        //4] Finding last element inside list
        while(test->next != NULL)
        {
                test = test->next;  //Iterator
        }

        //5]Link list
        test->next = temp;

        cout << endl << "Sucessfully element added in tail";
}
//-------------------------------------------------------
```

```
//-------------------------------------------------------
//For printing linked list
void print_link(NODE *p)
{
  NODE *temp = p;
  cout<< endl << "List print : ";
  while(temp->next != NULL)
  {
        //PRINT current element
        cout << temp->data << " , ";
        //Update for next node
        temp = temp->next;

  }
}
//-------------------------------------------------------




int main()
{
      //Create null pointer act as HEAD
      NODE *user = NULL;

      //Insert value in list
      insert_at_tail(user,1);
      insert_at_tail(user,2);
      insert_at_tail(user,3);
      insert_at_tail(user,4);
      insert_at_tail(user,5);

      //print list
      print_link(user);

      return 0;
}
```

OUTPUT:


Select D:\Jaydeep Shah\MY_LEARNING\c++\Advance_C++\List in C++\Linked_list_1_C++.exe
Sucessfully element added in tail
Sucessfully element added in tail
Sucessfully element added in tail
Sucessfully element added in tail
List print : 1 , 2 , 3 , 4 ,

Process exited after 0.04535 seconds with return value 0
Press any key to continue . . .