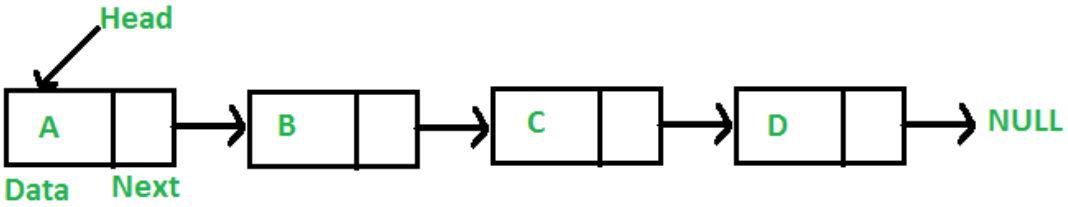
Name: Hitendra Sisodia
Sap id :500091910

About LinkList

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers.



In simple words, a linked list consists of nodes where each node contains a data field and a reference(link) to the next node in the list.

We cannot access any element directly in the link list as they are non-continuous.

Types of link list:

- Singly link list – Item navigation is forward only.
- Circular link list – Items can be navigated forward and backward.
- Doubly link list – Last item contains link of the first element as next and the first element has a link to the last element as previous.

Name: Hitendra Sisodia
Sap id :500091910

Lab intro

Ques: Create an integer array 'A' of size 10. Insert elements, Read data from A[9] to A[0]. Print the middle element. Create a linked list of the same elements and traverse the list (print items).

Ans:

Source Code

```
/* Name: Hitendra Sisodia
Sap Id: 500091910 */
#include<iostream>
using namespace std;
struct node                             //defination of struct
{
    int data;
    node *next;
};
int main()
{
    node *head,*newnode,*temp;
    head=0;

    int a[10];
    for(int i=0;i<10;i++){
        cout<<"Enter the "<<i+1<<" element at "<<i<<" index:";          //loop for input
        int x;
        cin>>x;
        a[i]=x;
        newnode=(node *)malloc(sizeof(node));
        newnode -> data=x;
        newnode -> next=NULL;
        if(head==0){
            head=temp=newnode;
        }
        else{
            temp -> next =newnode;
            temp=newnode;
        }
    }
    for(int i=9;i>=0;i--){
        cout<<a[i]<<" ";                                //loop for display resultant array
    }
    cout<<endl;
    cout<<"Middle Element:"<<a[10/2]<<endl;
    temp=head;
    while(temp!=0){
        cout<<temp->data<<" ";              //loop for display resultant linklist
        temp=temp->next;
    }
    return 0;
}
```

Output

```
Enter the 1 element at 0 index:1
Enter the 2 element at 1 index:2
Enter the 3 element at 2 index:3
Enter the 4 element at 3 index:4
Enter the 5 element at 4 index:5
Enter the 6 element at 5 index:6
Enter the 7 element at 6 index:7
Enter the 8 element at 7 index:8
Enter the 9 element at 8 index:9
Enter the 10 element at 9 index:10
10 9 8 7 6 5 4 3 2 1
Middle Element:6
1 2 3 4 5 6 7 8 9 10
```