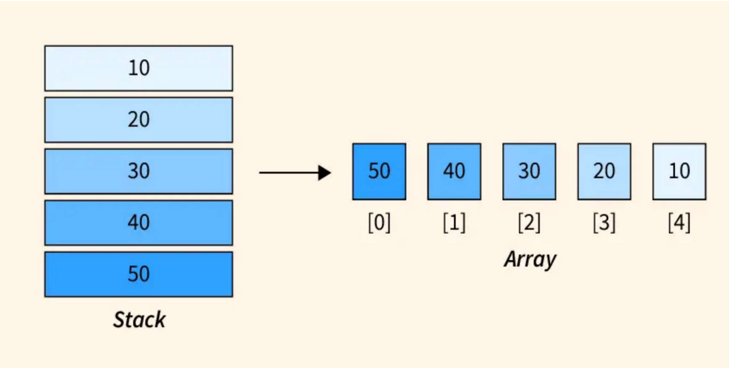
Stack

A stack is a linear data structure where all insertions and deletions are restricted to one end, the top. It is also known as a Last-In-First-Out (LIFO) list because the last element inserted into a stack is the first element removed.

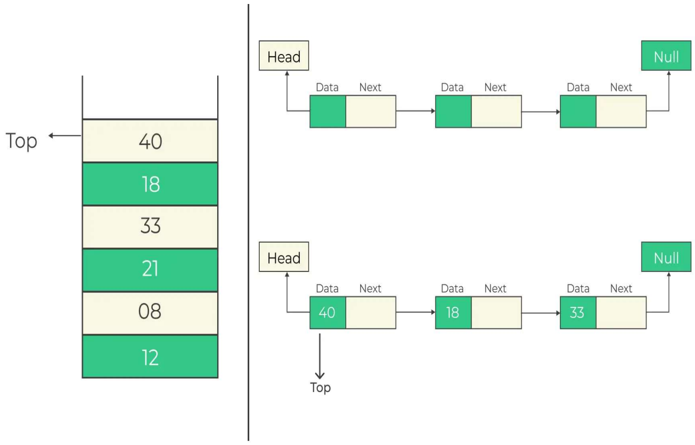
Stack using array:

Firstly, we are asking for the size of the stack required. The constructor in arrayStack makes a stack of size ‘n’. Then the stack's functions (or methods) are written for the array datatype. Now, we can create a stack in the main using array by importing the file ‘arrayStack.cpp’ .

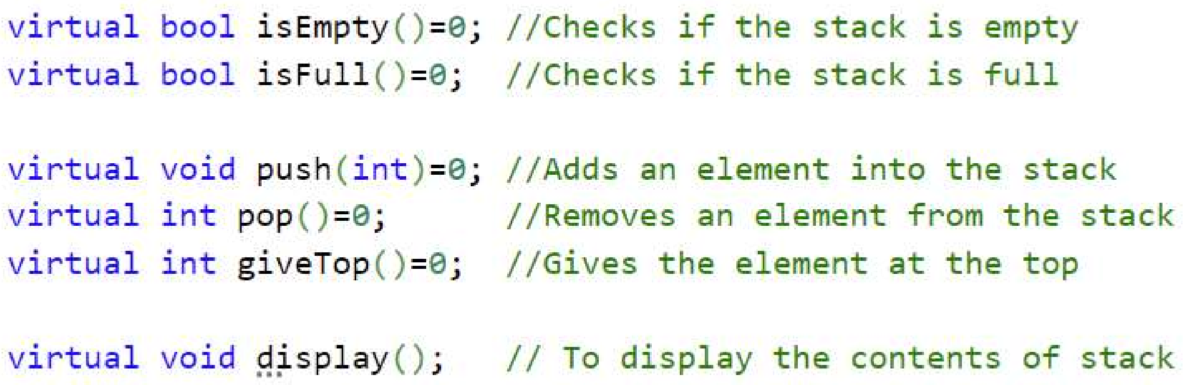


Stack using LinkedList:

Similarly, as an array, we use a linked list to represent a stack. There is a specific length of the stack as linked list is expandable. Also, we are using the linked list file made for lab 1 and the features of a linked list to perform the required operations to make a stack. Similarly, importing ‘linkedlistStack.cpp’ allows the user to create and operate on a stack made up of a linked list.



Features of code:

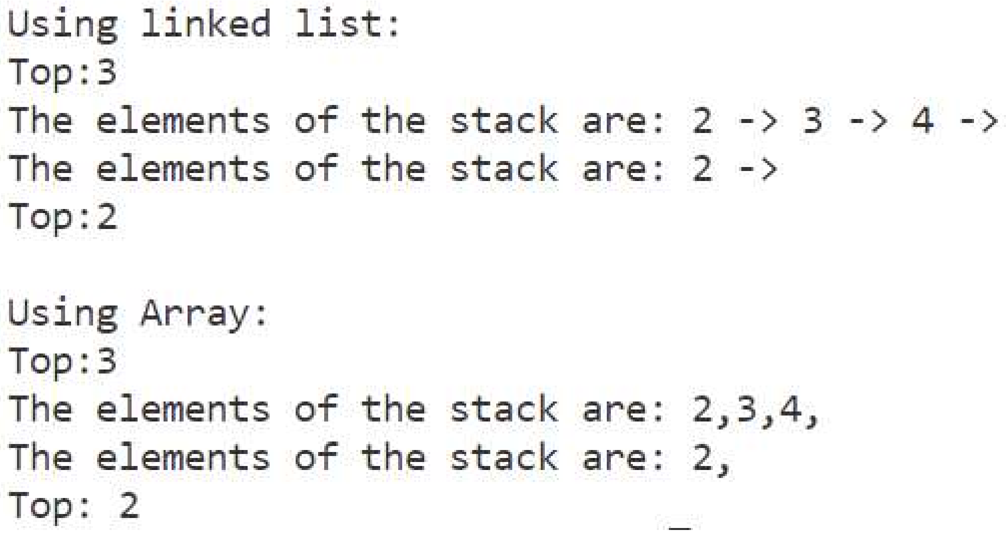


```
virtual bool isEmpty()=0; //Checks if the stack is empty
virtual bool isFull()=0;  //Checks if the stack is full

virtual void push(int)=0; //Adds an element into the stack
virtual int pop()=0;      //Removes an element from the stack
virtual int giveTop()=0;  //Gives the element at the top

virtual void display();   // To display the contents of stack
```

Snippet from Stack.h

Output showing the working of code

Contents of compiled main.cpp:



```
Using linked list:
Top:3
The elements of the stack are: 2 -> 3 -> 4 ->
The elements of the stack are: 2 ->
Top:2

Using Array:
Top:3
The elements of the stack are: 2,3,4,
The elements of the stack are: 2,
Top: 2
```