I need a class named of Element, because stack has elements.
In linked list we used node.

```
class Element{
public:

   int data;
   Element* next;

};
```

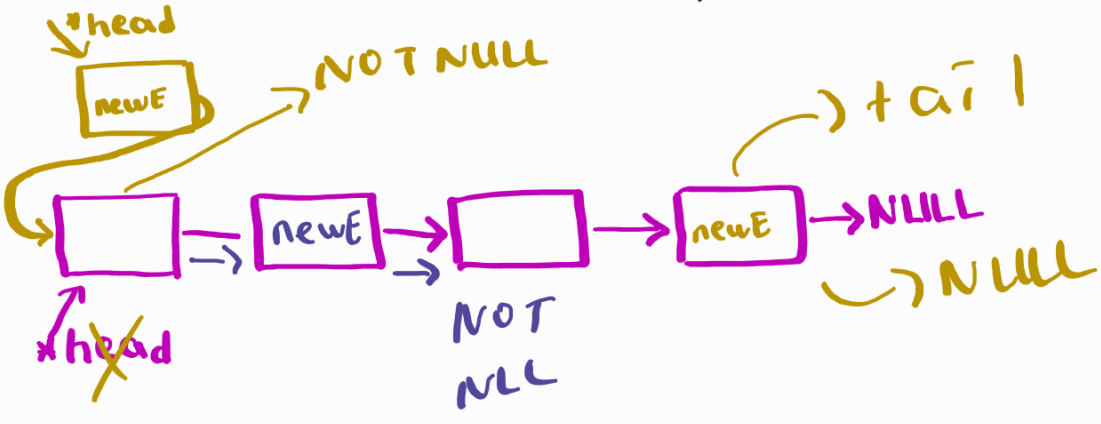
Also I need a stack class that has some methods.

```
class Stack {

 public:

   Stack();
 ~Stack();
```

```
  void push(int);
  void pop();
  void reverse();
  void print();
  int size();
  int top();
  void stackSort(Stack A, Stack B);
  bool empty(Element * );

  private:

   Element * tail;
  Element * head;

};
```

Method of push:

1.Create new element.
2. Define a data of element as value.
3. Next of element is head than head point newE.
4. Look stack is empty?
5. If stack is not empty , while newE->next is not NULL
newE =newE->next; so tail is newE.



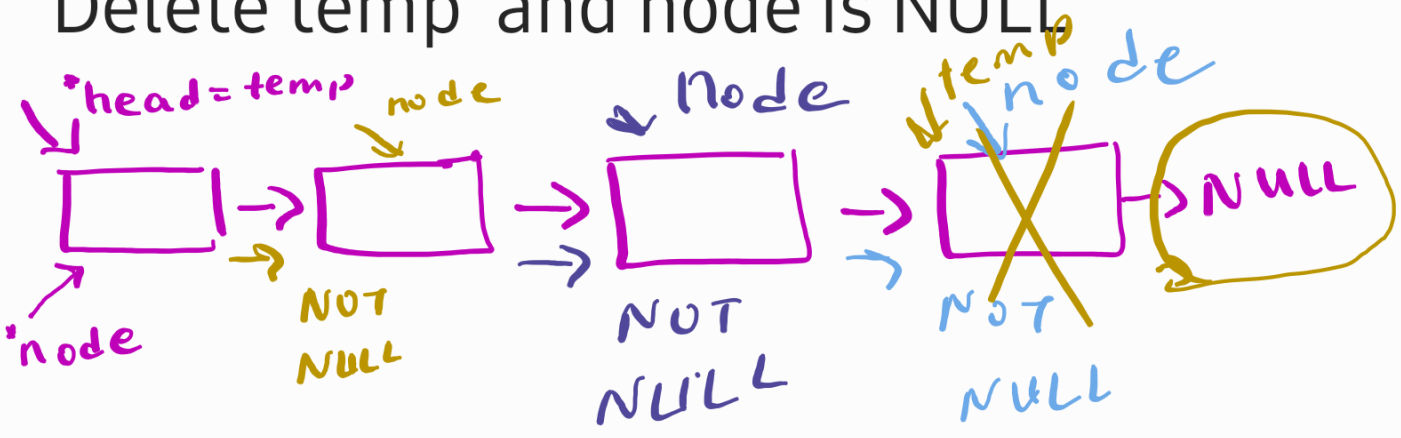
Method of pop:

1.If tail is NULL we can not delete.

2. Create *temp =head , *node =head

3. While next of node is not NULL node is next of node , temp is next of node.

Delete temp and node is NULL

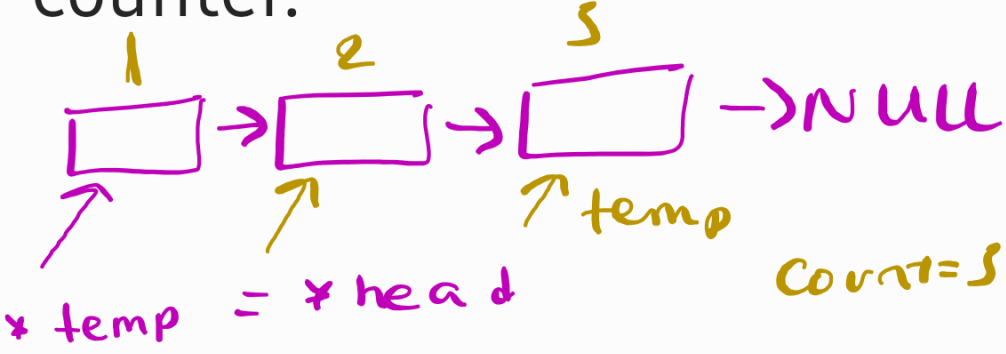


Method of size :

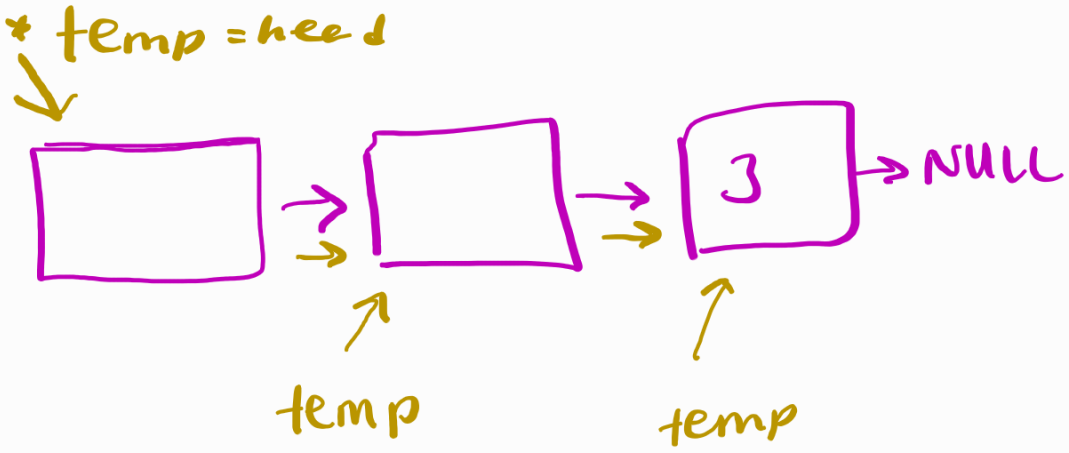
1. Define count which is type of int and it is 0

2. Create temp pointer that point head

3. While temp is not NULL increase plus 1 counter.



Method of size:




```
temp -> data = 3;
```