

```

class _Node:
    __slots__ = '_element', '_next'

    def __init__(self, element, next):
        self._element = element
        self._next = next

class StacksLinked:
    def __init__(self):
        self._top = None
        self._size = 0

    def __len__(self):
        return self._size

    def isempty(self):
        return self._size == 0

    def push(self, e):
        newest = _Node(e, None)
        if self.isempty():
            self._top = newest
        else:
            newest._next = self._top
            self._top = newest
        self._size += 1

    def pop(self):
        if self.isempty():
            print('Stack is empty')
            return
        e = self._top._element
        self._top = self._top._next
        self._size -= 1
        return e

    def top(self):
        if self.isempty():
            print('Stack is empty')
            return
        return self._top._element

    def display(self):
        p = self._top
        while p:
            print(p._element, end='-->')
            p = p._next
        print()

```

```
S = StacksLinked()
S.push(5)
S.push(3)
print(len(S))
S.display()
print(S.pop())
print(S.isempty())
```

```
print(S.pop())
print(S.isempty())
S.push(7)
S.push(9)
print(S.top())
S.push(4)
print(len(S))
print(S.pop())
S.push(6)
S.push(8)
print(S.pop())
```