Stack

1] Stack Data Structure using list :

```
stack=[]
stack.append(10)
stack.append(20)
stack.append(30)
print(stack)
print(stack.pop())

top=stack[-1]
print(top)
size=len(stack)
print(size)
```

OUTPUT :

```
[10, 20, 30]

30

20

2
```

2] Stack Implementation using deque :

```
from _collections import deque
stack=deque()

stack.append(10)
stack.append(20)
stack.append(30)
print(stack)
print(stack.pop())

top=stack[-1]
print(top)
size=len(stack)
print(size)
```

OUTPUT :

```
deque([10, 20, 30])
30
20
2
```

3] Linked list implementation of stack in python :

```python
import math
#inf constant returns a floating-point positive infinity.
# For negative infinity, use -math. inf .
# The inf constant is equivalent to float('inf') .
class Node:
    def __init__(self,d):
        self.data=d
        self.next=None

class MyStack:
    def __init__(self):
        self.head=None
        self.sz=0

    def push(self,x):
        temp=Node(x)
        temp.next=self.head
        self.head=temp
        self.sz=self.sz+1

    def size(self):
        return self.sz

    def peek(self):
        if self.head==None:
            return math.inf
        return self.head.data

    def pop(self):
        if self.head==None:
            return math.inf
        res=self.head.data
        self.head=self.head.next
        self.sz=self.sz-1
        return res


s=MyStack()
s.push(10)
s.push(20)
s.push(30)
print(s.pop())
print(s.peek())
print(s.size())
```

OUTPUT :

```
30

20

2
```

4] Check for balanced parenthesis in Python :

```
def isMatching(a,b):
    if (a=="(" and b==")") or (a=="{" and b=="}") or \
            (a=="[" and b=="]"):
        return True
    else:
        return False


def isBalanced(exoer):
    stack=[]
    for x in exoer:
        if x in ("(","{","["):
            stack.append(x)
        else:
            if not stack:
                return False
            elif isMatching(stack[-1],x)==False:
                return False
            else:
                stack.pop()
    if stack:
        return False
    else:
        return True

a=input()

print(isBalanced(a))

a=input()

print(isBalanced(a))



#OUTPUT
"""
{{[]{{(())}}}}
True
[[[[[{{))}}}}}
False
"""
```