

version of stack where last item on list is top of stack

```
class Stack:
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return self.items == []

    def push(self, item):
        self.items.append(item)

    def pop(self):
        return self.items.pop()

    def peek(self):
        return self.items[len(self.items) - 1]

    def size(self):
        return len(self.items)

def reverse_string(stack, input_str):
    for i in range(len(input_str)):
        stack.push(input_str[i])

    rev_str = ""
    while not stack.isEmpty():
        rev_str += stack.pop()
    return rev_str

def base_converter(dec_number, base):
    digits = "0123456789ABCDEF"

    rem_stack = Stack()

    while dec_number > 0:
        remainder = dec_number % base
        rem_stack.push(remainder)
        dec_number = dec_number // base

    bin_string = ""
    while not rem_stack.isEmpty():
        a = digits[rem_stack.pop()]
        bin_string = bin_string + a
        # print(a)

    return bin_string

def parChecker(symbolString):
    s = Stack() # creates a empty stack
    balanced = True # flag
    index = 0

    while index < len(symbolString) and balanced:
        symbol = symbolString[index]
        if symbol == '(':
            s.push(symbol)
        else:
            if s.isEmpty():
                balanced = False
            else:
                s.pop()
        index = index + 1
```

```
if balanced and s.isEmpty():  
    return True  
else:  
    return False
```