In [1]:

```
def ct(L):
    rows = len(L)
    for row in range(rows):
        M = L[row]
        M.reverse()
    for row in range(rows):
        col = row
        print(L[row][col])
L = [[3, 2], [5, 6]]
ct(L)
print(L)
```

```
2
5
[[2, 3], [6, 5]]
```

In [2]:

```
import copy
def ct(L):
    M = L
    L = copy.copy(L)
    N = copy.deepcopy(L)
    L[0][0] = 4
    M[0] = [6]
    M[1][0] = 1
    N[1][0] = 3
    print(L)
    print(M)
    print(N)
L = [[2], [5]]
ct(L)
print(L)
```

```
[[4], [1]]
[[6], [1]]
[[2], [3]]
[[6], [1]]
```

In [3]:

```
import copy
# Hint: this prints 2 lines
def ct(L):
    L = copy.copy(L)
    M = copy.deepcopy(L)
    rowList = L[1]
    M[0] = rowList * rowList[0]
    rowList = L.pop()
    M[0][0] += rowList[0]
    rowList[0] += 1
    M[1] = L[0]
    return M
L = [[5], [2], [7]]
print(ct(L))
print(L)
```

```
[[9, 2], [5], [7]]
[[5], [2], [8]]
```

In [ ]:

```
def ct(L):
    rows = len(L)
    for row in range(rows):
        M = L[row]
        M.reverse
    for row in range(rows):
```

```
        col = row
        print(L[row][col])
L = [[5, 3], [5, 6]]
ct(L)
print(L)

# prints:
# 3
# 5
# [[3, 5], [6, 5]]
```

In [5]:

```
def ct(L):
    result = []
    rows, cols = len(L), len(L[0])
    for row in range(0, rows, 2):
        for col in range(cols - 1, -1, -1):
            result.append(L[row][col])
    return result
L = [[8, 9, 5], [1, 4, 7], [2, 6, 3]]
print(ct(L))
```

```
[5, 9, 8, 3, 6, 2]
```

In [6]:

```
import copy
def ct(L):
    A = copy.deepcopy(L)
    B = [A[0], L[1]]
    B[1][0] = 1
    A[0].pop()
    A = A[1] + A[0]
    B[1] = sorted(B[0])
    print(A)
    print(B)
L = [[1, 3, 2], [5, 4]]
ct(L)
print(L)
```

```
[5, 4, 1, 3]
[[1, 3], [1, 3]]
[[1, 3, 2], [1, 4]]
```

In [8]:

```
import copy
# Hint: this prints 3 lines
def ct(A):
    L, M, N = A, copy.copy(A), copy.deepcopy(A)
    L[0][0] += 4
    # Hint: L += M is mutating but L = L + M is non-mutating
    M[1] += [3]
    M[1] = M[1] + [4]
    N[1] = L[0]
    N[1][0] = N[1][0] + 7
    print(M)
    return N
L =     [[4], []]
print(ct(L))
print(L)
```

```
[[15], [3, 4]]
[[4], [15]]
[[15], [3]]
```

In [9]:

```
def ct(M):
    rows, cols = len(M), len(M[0])
    result = []
```

```
    for row in range(rows):
        for col in range(1, cols):
            M[row][0] += M[row][col]
            result = [M[row][0]] + result
    return result
L = [[6, 1, 4], [5, 3, 2]]
print(ct(L))
print(L)
```

```
[10, 8, 11, 7]
[[11, 1, 4], [10, 3, 2]]
```

In [10]:

```
def ct1(L):
    M = copy.copy(L)
    N = copy.deepcopy(L)
    L[0] = L[1]
    M[1] = M[2]
    L[1][1] = 3
    N[0] = L[2]
    return (M, N)
L = [[5],[6,7],[8]]
print(ct1(L))
print(L) # don't miss this!
```

```
([[5], [8], [8]], [[8], [6, 7], [8]])
[[6, 3], [6, 3], [8]]
```

In [12]:

```
def ct2(L):
    rows, cols = len(L), len(L[0])
    M = [ ]
    for i in range(min(rows, cols)):
        M.append(L[i].pop(i))
    L.append(M)
L = [[1,2],[3,4],[5,6]]
ct2(L)
print(L)
```

```
[[2], [3], [5, 6], [1, 4]]
```

In [13]:

```
import copy
def f(a):
    return 10*a[0][0]+a[1][0]

def ct1(a):
    b = copy.copy(a)
    c = copy.deepcopy(a)
    d = a
    e = a[0:len(a)]
    c[0][0] = 1
    d[0] = [2]
    e[1] = [3]
    b[0][0] = 4
    print(f(b),f(c),f(d),f(e)) #f is defined above

a= [[5],[6]]
ct1(a)
print(f(a))# don't miss this
```

```
46 16 26 43
26
```

In [14]:

```
#prints 2 lists containing lists
import copy
def ct1(L):
```

```
    a = L
    b = copy.copy(L)
    c = copy.deepcopy(L)
    b[0] = a[1] * a[1][0]
    a[0][0] += a.pop()[0]
    b[1] = c[0]
    return b
# Be careful to get the brackets and commas right!
L = [[1],[2],[3]]
print(ct1(L))
print(L)
```

```
[[2, 2], [1], [3]]
[[4], [2]]
```

In [15]:

```
# Prints 1 list which may contain anything
def ct2():
    a = []
    for i in range(1,4):
        b = [i]
        for j in range(i):
            if j % 2 == 0:
                b.append(j)
            else:
                a.append(j)
        a.append(b)
    return a
print(ct2())
```

```
[[1, 0], 1, [2, 0], 1, [3, 0, 2]]
```