CSIT110 / CSIT810
Python

Lecture 10

Dr. Joseph Tonien

School of Computing and Information Technology
University of Wollongong

Objectives

Understanding of:

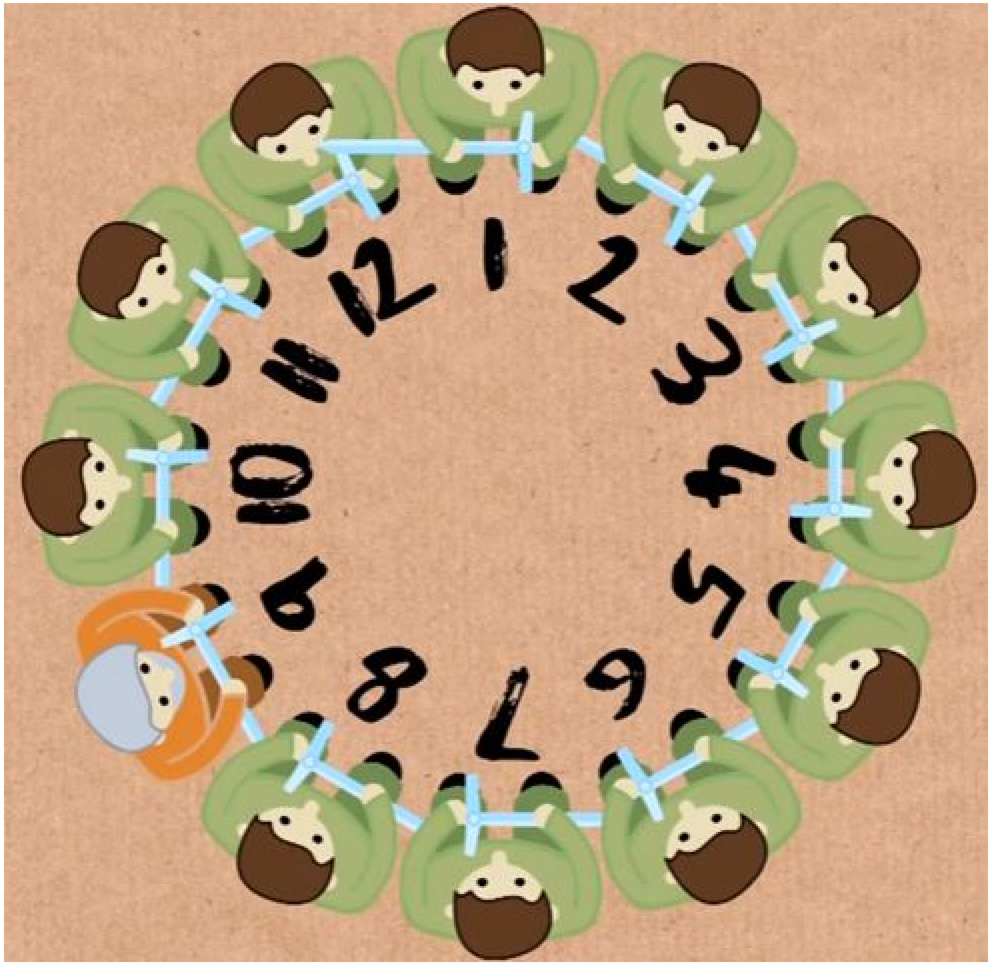
- List
- Multi-dimensional list

Problem solving using:

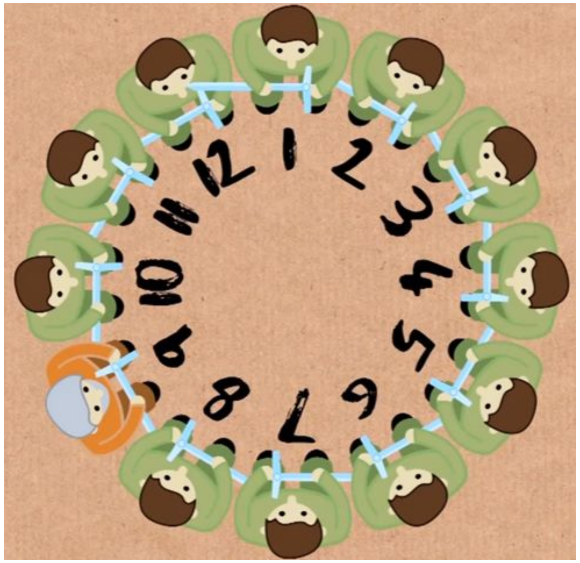
- List

The Josephus Problem

https://www.youtube.com/watch?v=uCsD3ZGzMgE



The Josephus Problem



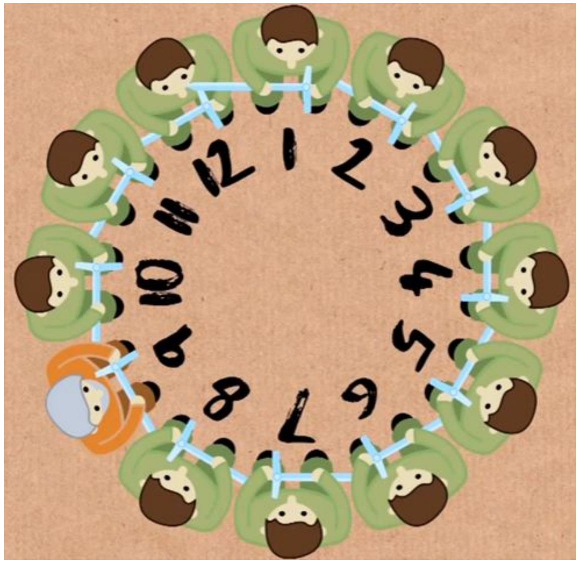
We want to use list to represent the current status

This is the initial status

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

and it is number 1's turn

The Josephus Problem



This is the initial status

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

12
1
3
4
5
6
7
8
9
10
11

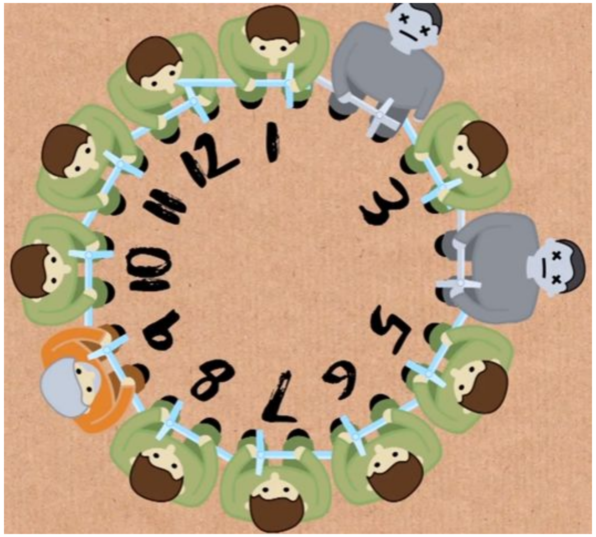
After 1 killed 2, the status is:

```
[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1]
```

and it is number 3's turn

The Josephus Problem

1
2
3
4
5
6
7
8
9
10
11
12



```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

After 1 killed 2

```
[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1]
```

After 3 killed 4

```
[5, 6, 7, 8, 9, 10, 11, 12, 1, 3]

[7, 8, 9, 10, 11, 12, 1, 3, 5]

[9, 10, 11, 12, 1, 3, 5, 7]

[11, 12, 1, 3, 5, 7, 9]

[1, 3, 5, 7, 9, 11]

[5, 7, 9, 11, 1]

[9, 11, 1, 5]

[1, 5, 9]

[9, 1]

[9]
```

← *winning seat*

The Josephus Problem

12
1
3
4
5
6
7
8
9
10
11

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

After 1 killed 2

```
[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1]
```

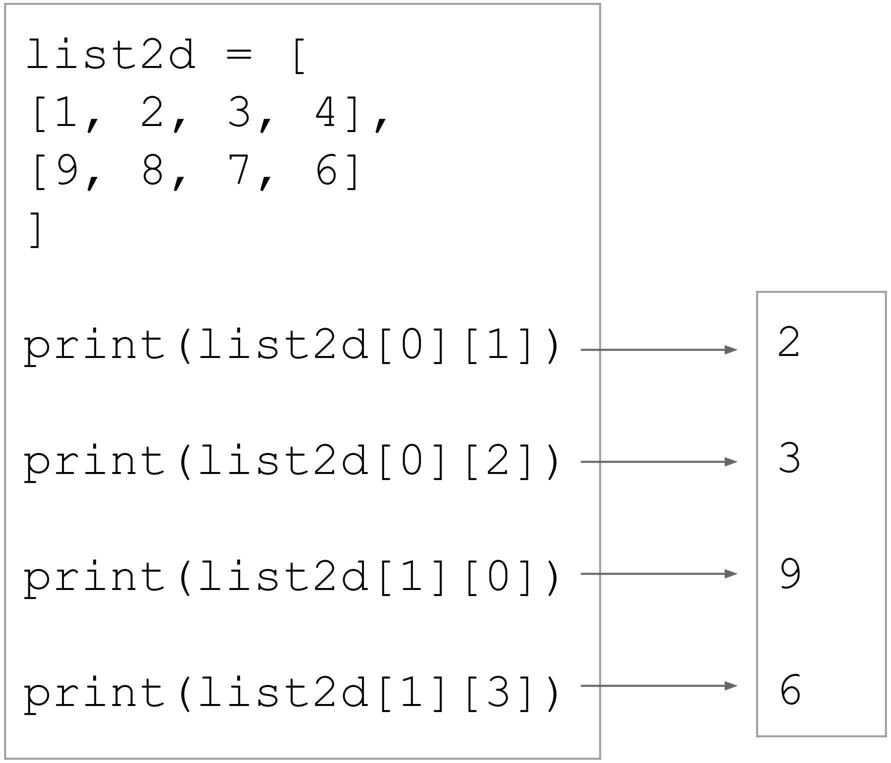
```
seat = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

# the person at the index 0 has the turn to kill the next person
turn = seat[0]
next_person = seat[1]
print("{0} kills {1}".format(turn, next_person))

# delete the first two persons
del seat[0]
del seat[0]

# append the person at the last turn to the end
seat.append(turn)
```

The Josephus Problem

```
seat = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
print(seat)

while True:
  # if there is only one person remaining then stop the loop
  if (len(seat) == 1):
    print("the winning seat is {0}".format(seat[0]))
    break

  # the person at the index 0 has the turn to kill the next person
  turn = seat[0]
  next_person = seat[1]
  print("{0} kills {1}".format(turn, next_person))

  # delete the first two persons
  del seat[0]
  del seat[0]

  # append the person at the last turn to the end
  seat.append(turn)
  print(seat)
```

Multi-dimensional list

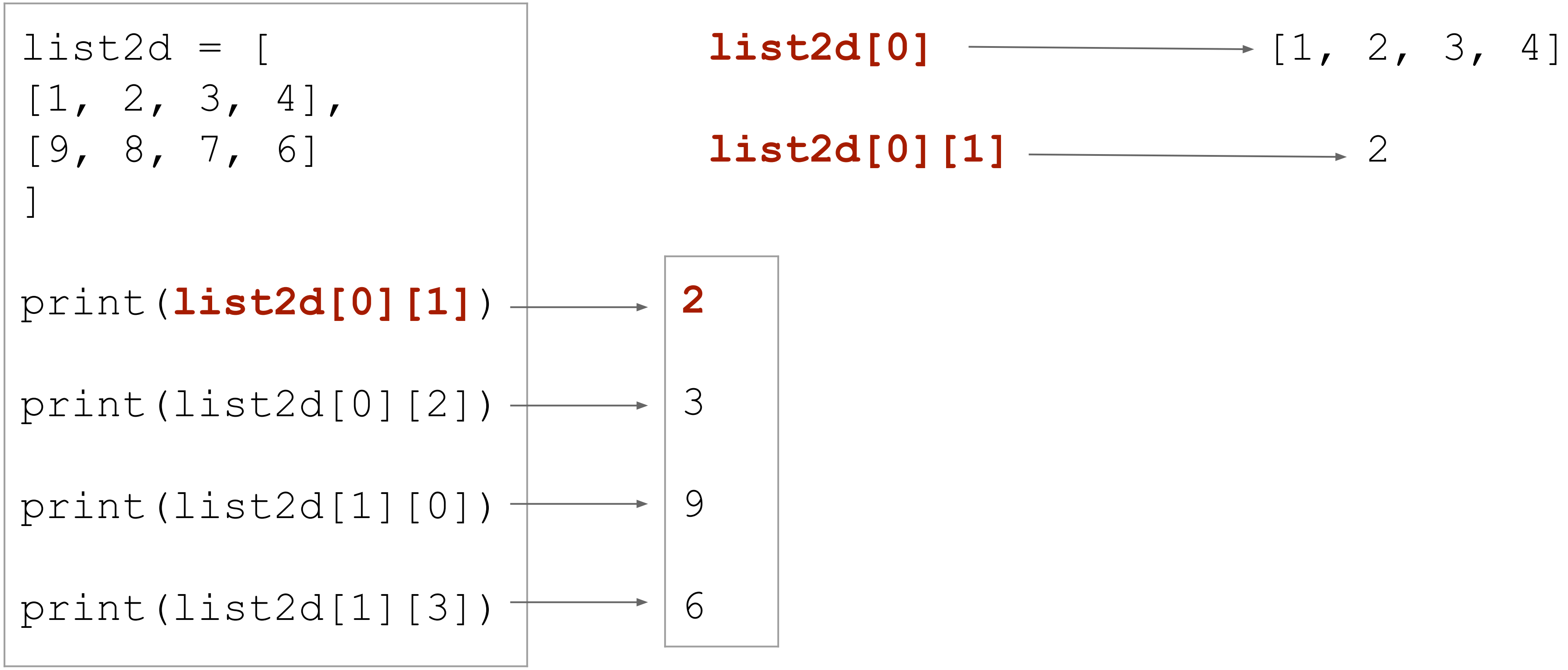
Two-dimensional list



```
list2d = [
[1, 2, 3, 4],
[9, 8, 7, 6]
]

print(list2d[0][1])

print(list2d[0][2])

print(list2d[1][0])

print(list2d[1][3])
```

```
2

3

9

6
```

Two-dimensional list



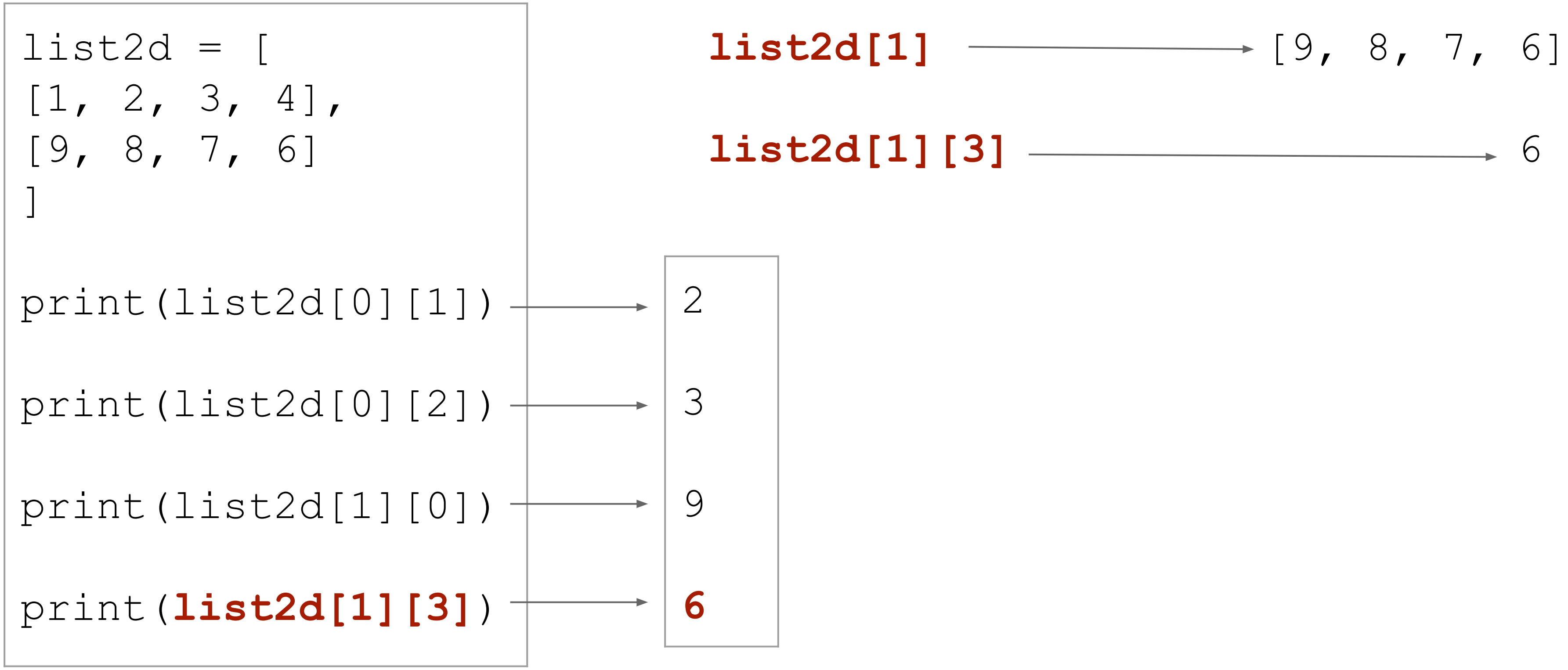
```
list2d = [
[1, 2, 3, 4],
[9, 8, 7, 6]
]

print(list2d[0][1])

print(list2d[0][2])

print(list2d[1][0])

print(list2d[1][3])
```

```
2

3

9

6
```

list2d[0] → [1, 2, 3, 4]

list2d[0][1] → 2

Two-dimensional list



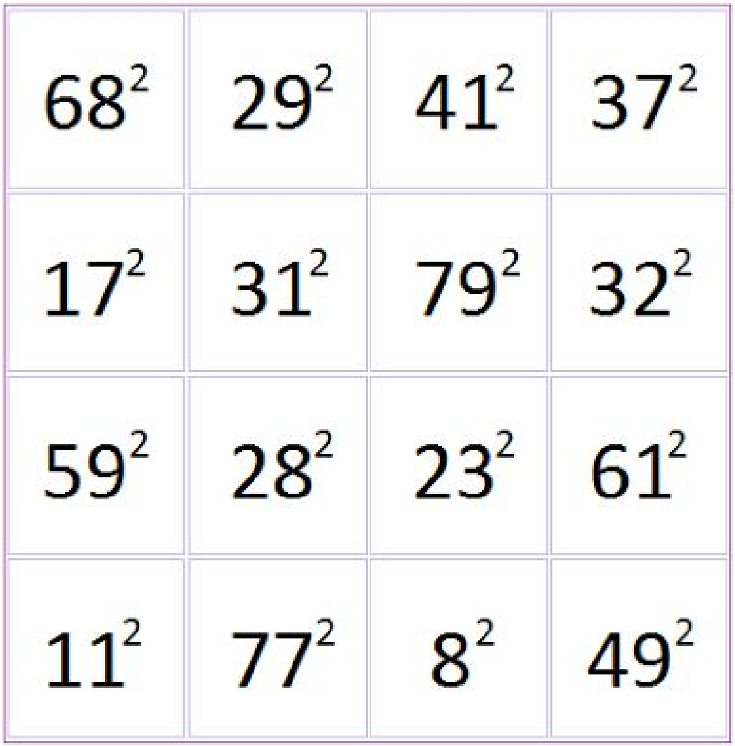
```
list2d = [
[1, 2, 3, 4],
[9, 8, 7, 6]
]

print(list2d[0][1])

print(list2d[0][2])

print(list2d[1][0])

print(list2d[1][3])
```

```
2

3

9

6
```

list2d[1] → [9, 8, 7, 6]

list2d[1][3] → 6

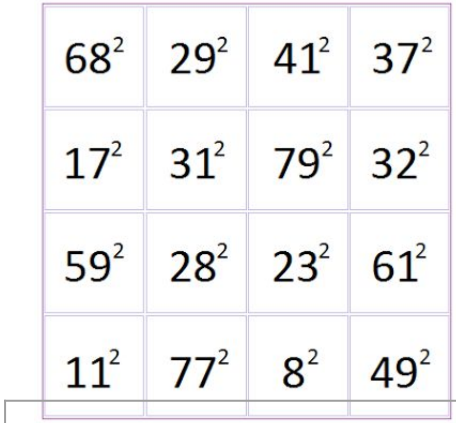
Euler’s magic square



68^2	29^2	41^2	37^2
17^2	31^2	79^2	32^2
59^2	28^2	23^2	61^2
11^2	77^2	8^2	49^2

Sum of numbers on each row, each column, and each diagonal is the same!

Euler’s magic square

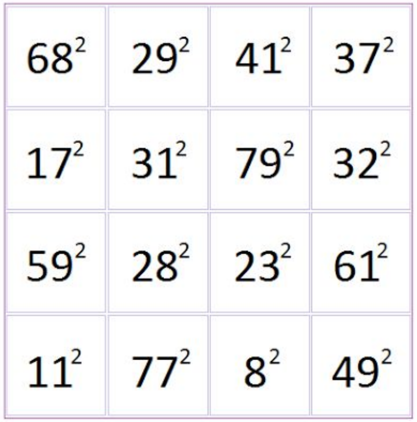


68^2	29^2	41^2	37^2
17^2	31^2	79^2	32^2
59^2	28^2	23^2	61^2
11^2	77^2	8^2	49^2

```
euler = [
[68**2, 29**2, 41**2, 37**2],
[17**2, 31**2, 79**2, 32**2],
[59**2, 28**2, 23**2, 61**2],
[11**2, 77**2,  8**2, 49**2]
]
```

```
# row sums
r1 = euler[0][0] + euler[0][1] + euler[0][2] + euler[0][3]
r2 = euler[1][0] + euler[1][1] + euler[1][2] + euler[1][3]
r3 = euler[2][0] + euler[2][1] + euler[2][2] + euler[2][3]
r4 = euler[3][0] + euler[3][1] + euler[3][2] + euler[3][3]
# column sums
c1 = euler[0][0] + euler[1][0] + euler[2][0] + euler[3][0]
c2 = euler[0][1] + euler[1][1] + euler[2][1] + euler[3][1]
c3 = euler[0][2] + euler[1][2] + euler[2][2] + euler[3][2]
c4 = euler[0][3] + euler[1][3] + euler[2][3] + euler[3][3]
# diagonal sums
d1 = euler[0][0] + euler[1][1] + euler[2][2] + euler[3][3]
d2 = euler[0][3] + euler[1][2] + euler[2][1] + euler[3][0]
print("r1={0}, r2={1}, r3={2}, r4={3}, c1={4}, c2={5}, c3={6},
c4={7}, d1={8}, d2={9}".format(r1, r2, r3, r4, c1, c2, c3, c4,
d1, d2))
```

Euler’s magic square



68^2	29^2	41^2	37^2
17^2	31^2	79^2	32^2
59^2	28^2	23^2	61^2
11^2	77^2	8^2	49^2

```
euler = [
[68**2, 29**2, 41**2, 37**2],
[17**2, 31**2, 79**2, 32**2],
[59**2, 28**2, 23**2, 61**2],
[11**2, 77**2,  8**2, 49**2]
]
```

```
r1=8515, r2=8515, r3=8515, r4=8515,
c1=8515, c2=8515, c3=8515, c4=8515,
d1=8515, d2=8515
```