# DSC 20
# Discussion Section 2

ARDA CANKAT BATI

# Piazza Poll

- Let's look at the piazza poll for today's coverage

# Today's Plan

- A few general questions
- Going over the reading quizzes
- Talking about file operations
- Doing some coding questions (mostly on loops)

# General Questions

How is HW2 compared to HW1?

- A) Easier
- B) Somewhat easier
- C) Same level
- D) A bit harder
- E) Harder

# General Questions

How fast do you think the course is going so far?

- A) Very fast
- B) Fast
- C) Normal
- D) Slow
- E) Very slow

# Reading Quizzes

- Let's go over reading quizzes 4, 5, 6

# Reading Quizzes

- Reading Quiz 4 Answers

1. B
2. A
3. B
4. D
5. B
6. A
7. C
8. D
9. A
10. A

- Reading Quiz 5 Answers

1. D
2. D
3. A
4. C
5. D

- Reading Quiz 6 Answers

1. B
2. A, B
3. D
4. A,B,C,E
5. A

# File Operations

open vs with open?

open (**filename**, **mode**='r', **buffering**=-1, **encoding**=None, **errors**=None, **newline**=None, **closefd**=True, **opener**=None)

**with** open("hello.txt", r) **as** file: #Read only mode
**with** open("hello.txt", rb) **as** file: #Read only binary mode

**with** open("hello.txt", w) **as** file: #Write only mode
**with** open("hello.txt", wb) **as** file: #Write only binary mode

# File Operations

**with** open("hello.txt", a) **as** file: #append mode
**with** open("hello.txt", ab) **as** file: #append binary mode

- For the append mode, the pointer is places at the end of the file, instead of the beginning as in write and read modes.
- In Python 3 mode 'r' opens the files in the text encoding provided by the user (if not provided the default encoding). read() will give you strings.
- mode 'rb' opens the files iin binary format (just bytes). read() will give a bytes object.
- Usually 'rb' and 'wb' are used for non-text files, like images, videos etc.

# File Operations

Assume the code below is run in an empty directory

```
def test1(file_path):

    with open("hello.txt", mode = 'w') as file:
        pass

    file.close()
```

What will happen?

A) Nothing will happen

B) File `hello.txt` (empty file) will be created

C) Error

# File Operations

Assume the code below is run in a directory with a file named "hello.txt"

```
def test1(file_path):

    with open("hello.txt", mode = 'r') as file:
          pass

    contents = file.read()
```

What will happen?

A) Nothing will happen

B) Contents of the file will be read to variable contents

C) Error

# Coding Questions

Pascal Triangle, Single Line

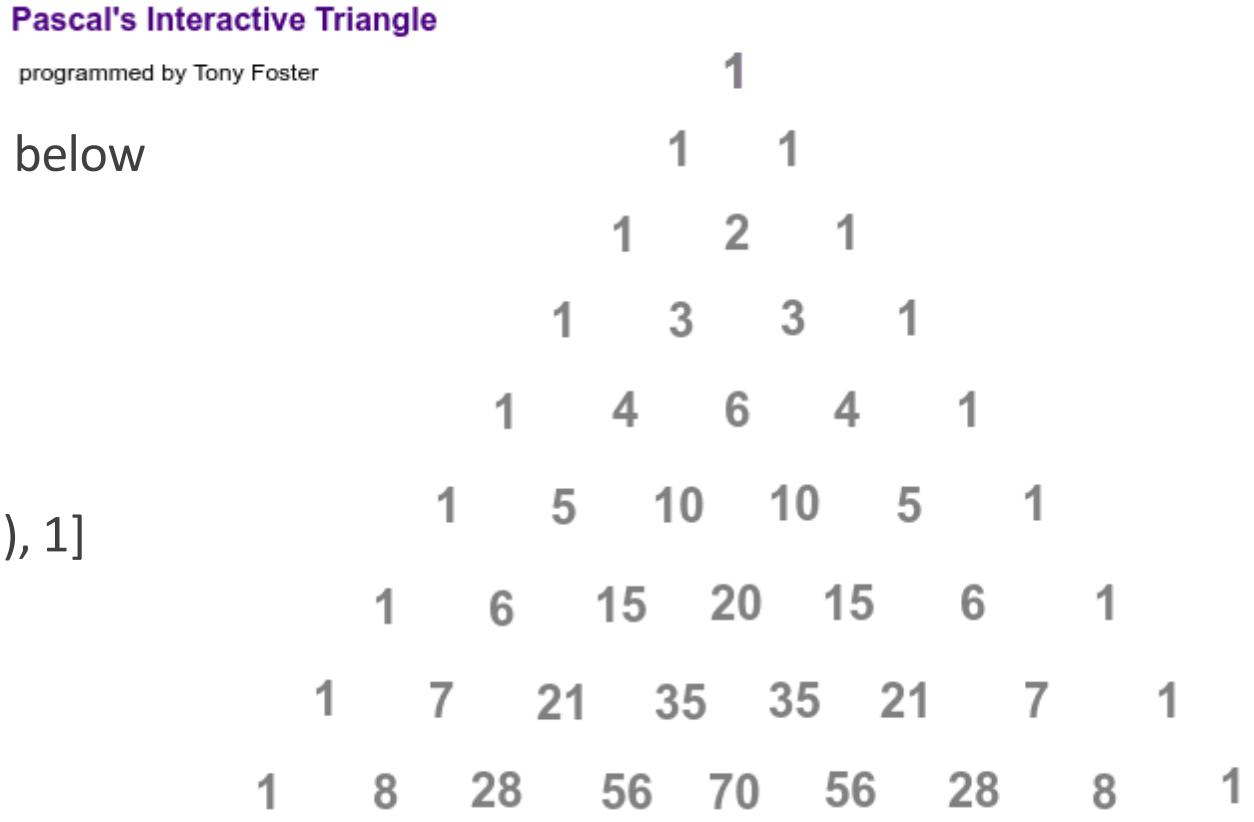
Given a line of pascal's triangle, return the line below it (both lines in list form)

Given list1 = [1, a1, a2, …, an, 1]

Return list2 = [1, (1 + a1), (a1 + a2), …, (an + 1), 1]

Example: Input: [ 1 3 6 3 1]

Output: [1 4 9 9 4 1]



# Coding Questions

```
def p_triangle_sum1(lst1):
    '''
    Returns one line of pascal's triangle, given the line above
    '''
    lst2 = [1]

    for i in range(len(lst1)-1):
        lst2.append(lst1[i] + lst1[i+1])

    lst2.append(1)
    return lst2
```

```
half_size = 2
skip_index = -2

def p_triangle_sum2(lst1):
    '''
    Returns one line of pascal's triangle, given the line above
    '''
    global half_size
    global skip_index

    lst1_even = len(lst1) % 2 == 1    # The 2 here is a magic number, should be fixed!
    lst2 = [1]

    for i in range(len(lst1)//half_size):
        lst2.append(lst1[i] + lst1[i+1])

    if(lst1_even):
        skip_index = -1

    lst2 += lst2[skip_index::-1]

    return lst2
```

# Coding Questions

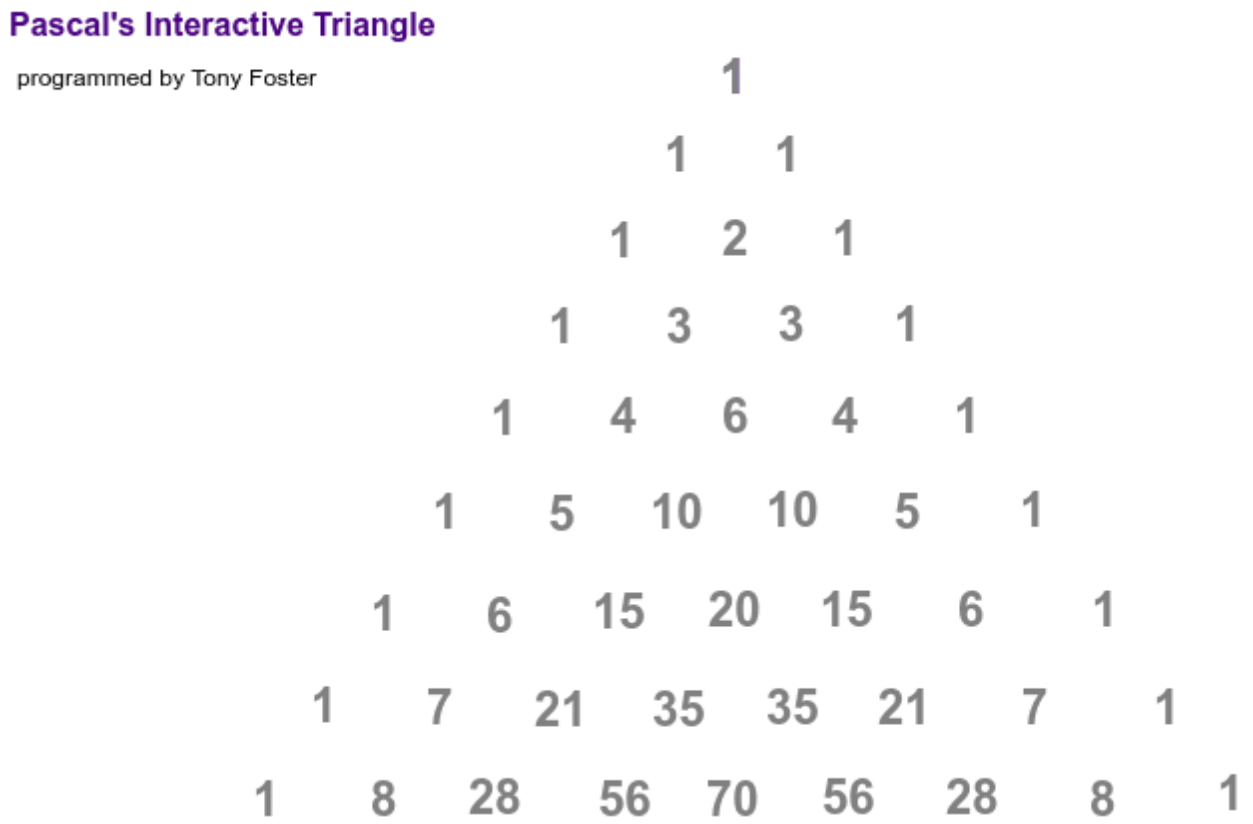
Form the first n levels of

pascal's triangle in list form:

[ [1], [1,1], [1,2,1], [1,3,3,1], …]

Assume you can use the function given in

The previous section



# Coding Questions

```
def build_pascal_triangle(n):
    n = n - 1
    pascal_list = [[1]]
    for i in range(n):
        cur_line = p_triangle_sum2(pascal_list[-1])
        pascal_list.append(cur_line)

    return pascal_list
```

# Coding Questions

Given two sets A & B (in the form of lists) find:

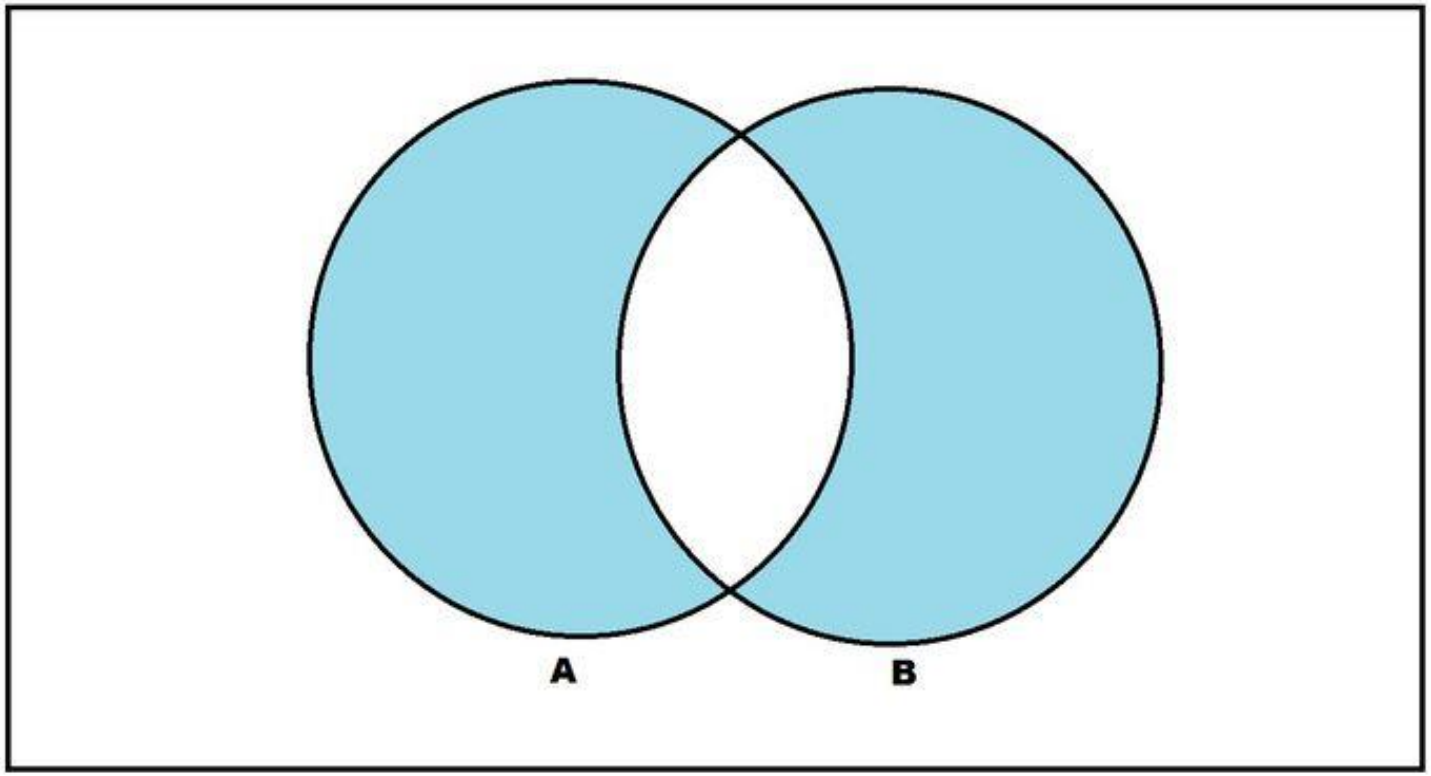
(A – B) U (B – A)

For example:

A = [1,2,3]

B = [2,3,4]

Result = [1,4]



# Coding Questions

Test Scores / Student Ids

You are given two lists:

List1 includes test scores of students (assume each score is unique)

List 2 includes student ids of the corresponding students

Order list 1 in ascending order, and order list2 accordingly such that Test score / student pairs remain correct.