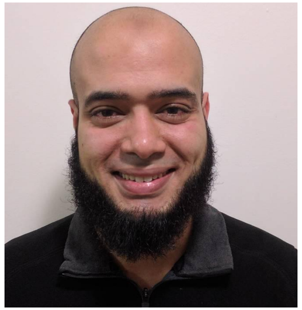
Python Programming

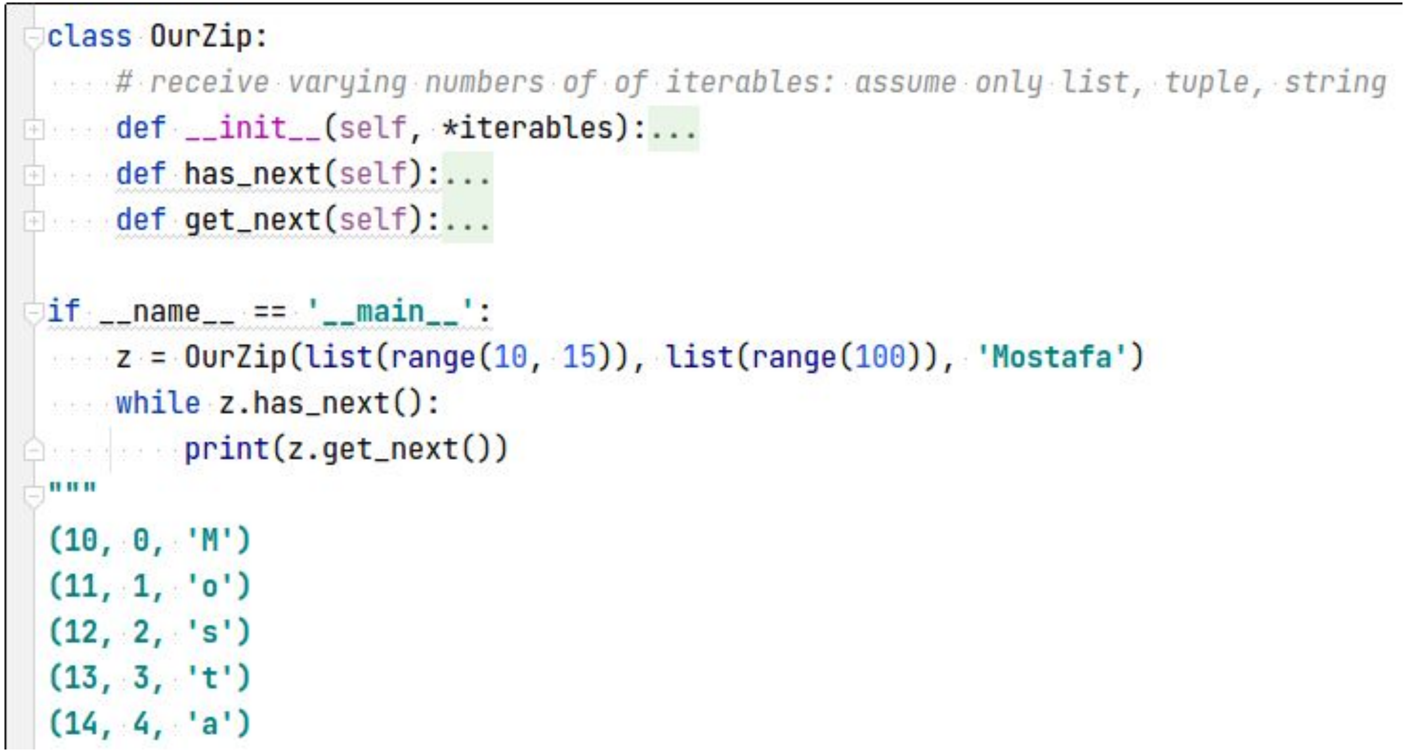
Nested List Homework 2

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Problem #1: Implement our zip: v1

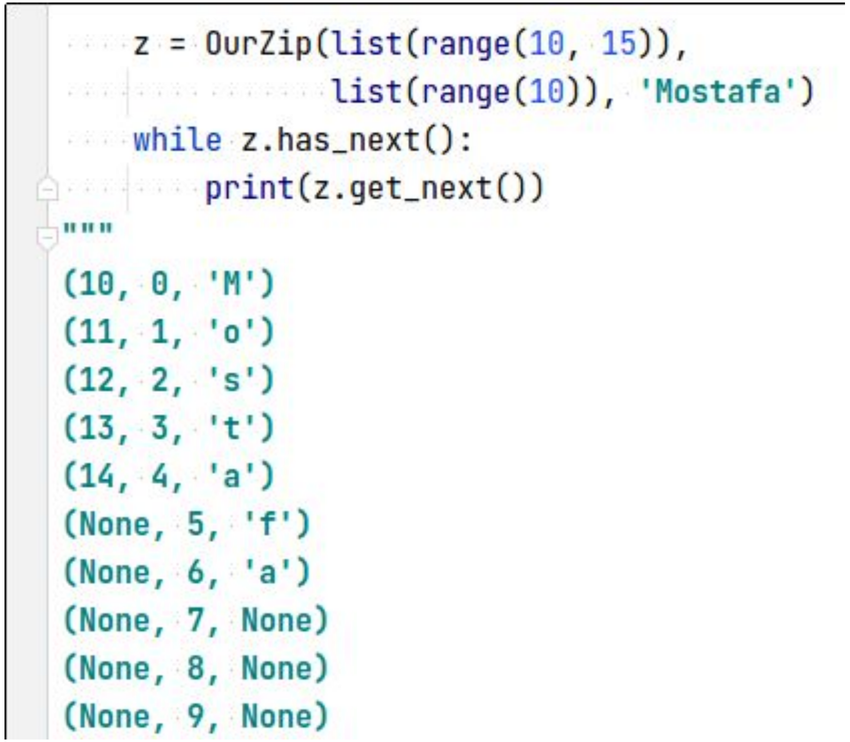


```python
class OurZip:
    # receive varying numbers of of iterables: assume only list, tuple, string
    def __init__(self, *iterables):...
    def has_next(self):...
    def get_next(self):...

if __name__ == '__main__':
    z = OurZip(list(range(10, 15)), list(range(100)), 'Mostafa')
    while z.has_next():
        print(z.get_next())
"""
(10, 0, 'M')
(11, 1, 'o')
(12, 2, 's')
(13, 3, 't')
(14, 4, 'a')
```

Problem #2: Implement our zip: v2

- In this variant, we will keep going up to the longest sequence. Replace missing values with None



```
    z = OurZip(list(range(10, 15)),
               list(range(10)), 'Mostafa')
    while z.has_next():
        print(z.get_next())
"""
(10, 0, 'M')
(11, 1, 'o')
(12, 2, 's')
(13, 3, 't')
(14, 4, 'a')
(None, 5, 'f')
(None, 6, 'a')
(None, 7, None)
(None, 8, None)
(None, 9, None)
```

Problem #3: How many primes

- Read a matrix. In next line, read integer Q, for Q queries.
    - In the next lines: read queries: sr sj r c
    - Each queries is a grid with **top left** (sr, sc) and #rows & #cols
    - For each query, print how many prime numbers in the requested sub-matrix.
- Input ⇒ Output
    - 3
    - 8 2 9 5
    - **3 2** 27 6
    - **7 8** 29 22
    - 2 ⇒ 2 queries
    - 1 0 2 2 ⇒ 3 (primes 3, 2, 7 in rectangle (0, 1) (2, 1))
    - 0 1 2 3 ⇒ 3 (primes 2, 5, 2 in rectangle (0, 1) (1, 3))

Problem #4: Greedy Robot

- Read an integer matrix (all **distinct** values)
- A robot starts at cell (0, 0).
- Take the value in the current cell and moves.
  - It can move only **one step** to either: *Right, Bottom or the diagonal.*
  - It always selects the destination cell that has **maximum value**.
- Print the total values the robot collects

3
1 2 3
4 **5** 6
7 8 **9**
⇒ (0, 0) (1, 1), (2, 2) ⇒ 15

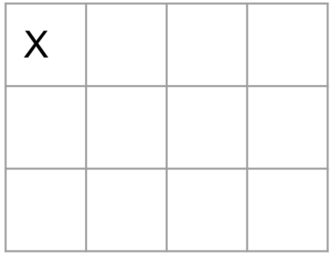
3
1 2 3
5 4 9
7 6 8
⇒ (0,0)⇒(1,0)⇒(2,0)⇒(2,1)⇒(2,2)
⇒27

2
1 2 3 4 5
6 7 8 9 10
⇒ 35

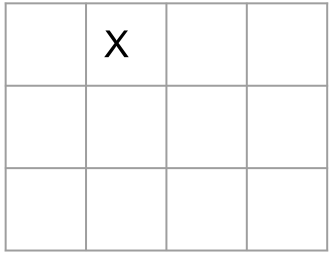
Problem #5: Active Robot

- Read a line that starts with integer values N M
  - It represents a grid NxM, where a robot starts at (0, 0)
- Then the remaining of the line is **several** commands
- Each command is 2 values
  - **Directio**: up, right, down, left
  - Steps: the number of steps to take in the direction. Steps [1, 1000000000]
  - If the robot hits the wall during the move, it **circulates** in the matrix.
  - For every command, print where is the robot now
- Input
  - 3 4 **right 1** **down 2** **left 2** **up 3**
- Output
  - (0, 1) (2,1) (2, 3) (2, 3)

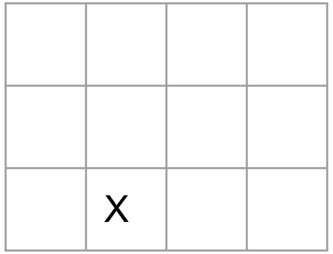
Problem #5: Active Robot



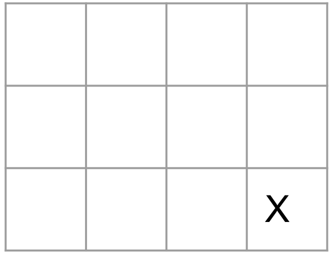
right 1 step ⇒
New pos (0, 1)



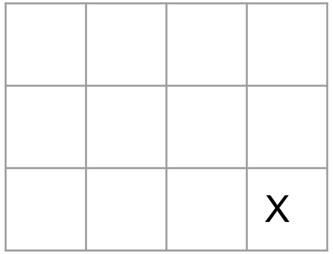
down 2 steps ⇒
New pos (2, 1)



left 2 steps ⇒
New pos (2, 3)
Circulation

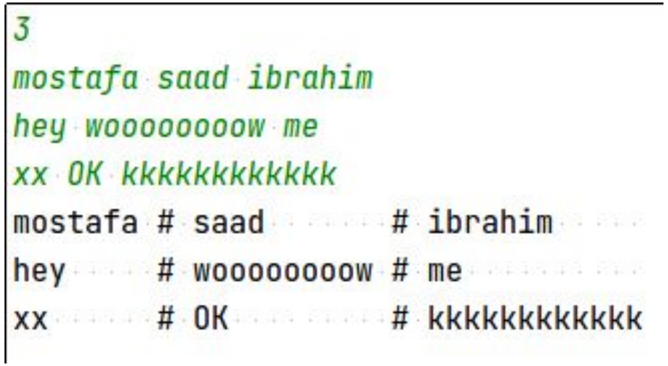


up 3 steps ⇒
New pos (2, 3)
Circulation



Problem #6: Matrix pretty print

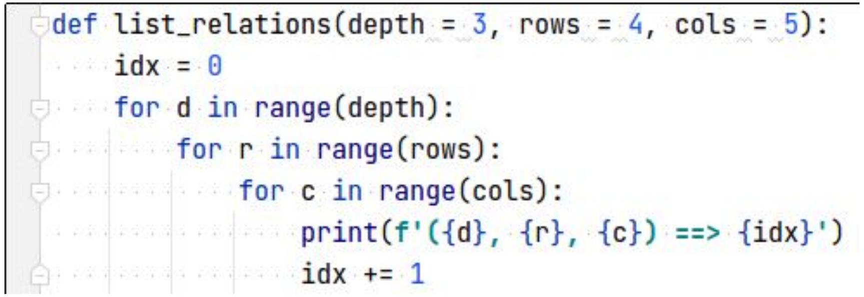
- Read a matrix of strings (no spaces, same # of columns)
- We would to **pretty print** the matrix such that
  - Each column is left justified based on the length of the longest string in the column
  - Seperate each 2 columns with ‘ # ‘
  - You will need to study: Python String ljust() Method
- Given the matrix, transform it to a new list of strings (one per row) Using 2 lines of code Hint: Use comprehension lists



```
3
mostafa saad ibrahim
hey wooooooooow me
xx OK kkkkkkkkkkk
mostafa # saad        # ibrahim
hey     # wooooooooow # me
xx      # OK          # kkkkkkkkkkk
```

Problem #7: Flatten 3D lists

- Read a line that starts with 3 numbers: DEPTH, ROWS, COLS the dimensions of 3D list
  - List of list of list
- Then the **remaining** of the line will be either:
  - 1 d r c (means convert from 3D to 1D)   or
  - 2 idx   (means convert from idx to 3D)
  - Can you generalize to higher dimensions? E.g. 6D
- Input ⇒ Outputs
  - 3 4 5   1   1 0 0   ⇒ 20
  - 3 4 5   2   20   ⇒ 1 0 0
  - 3 4 5   1   1 1 1   ⇒ 26
  - 3 4 5   1   2 3 2   ⇒ 57
  - 3 4 5   1   2 0 0   ⇒ 40
  - 3 4 5   2   59   ⇒ 2 3 4



```python
def list_relations(depth = 3, rows = 4, cols = 5):
    idx = 0
    for d in range(depth):
        for r in range(rows):
            for c in range(cols):
                print(f'({d}, {r}, {c}) ==> {idx}')
                idx += 1
```

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”