

2D lists problem solving

Iterate the list without range function

```
odi = [100, 99, 20]
test = [200, 150, 90, 180]
t20 = [20, 50, 100]
```

```
score = [odi, test, t20]
```

```
print(score)
```

```
[[100, 99, 20], [200, 150, 90, 180], [20, 50, 100]]
```

```
odi
```

```
[100, 99, 20]
```

Iterate on odi

This for loop iterates on all elements of odi list

```
for i in odi:
    print(i)
```

```
100
```

```
99
```

```
20
```

```
score
```

```
[[100, 99, 20], [200, 150, 90, 180], [20, 50, 100]]
```

```
for i in score:
    print(i)
```

```
[100, 99, 20]
```

```
[200, 150, 90, 180]
```

```
[20, 50, 100]
```

final code without range function

```
for i in score:
    for j in i:
        print(j, end=" ")
    print()
```

```
100 99 20
```

```
200 150 90 180
```

```
20 50 100
```

Challenge: Print maximum in odi, test and t20 separately

```
print(score)
```

```
[[100, 99, 20], [200, 150, 90, 180], [20, 50, 100]]
```

```
max(odi)
```

```
100
```

```
max(test)
```

```
200
```

```
max(t20)
```

```
100
```

Using following loop we are getting lists inside of score

```
for i in score:  
    print(max(i))
```

```
100
```

```
200
```

```
100
```

Question:

You are a data scientist at ICC and you need to find the runs scored Sachin and Ganguly in partnerships, also store them in a new list

*# len of both lists is same: Row * Col*

```
sachin = [  
    [1, 2, 3],
```

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

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

```
# 0 4 6
# 8 0 12
# 14 16 0
# 20 22 24
```

```
# Traversing on sachin
```

```
for i in range(len(sachin)):
    for j in range(len(sachin[i])):
        print(sachin[i][j], end=" ")
    print()
```

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

```
# indexes of Sachin's run
```

```
for i in range(len(sachin)):
    for j in range(len(sachin[i])):
        print(i, j, end=" ")
    print()
```

```
0 0 0 1 0 2
1 0 1 1 1 2
2 0 2 1 2 2
3 0 3 1 3 2
```

```
# Traversing on ganguly
```

```

for i in range(len(ganguly)):
    for j in range(len(ganguly[i])):
        print(ganguly[i][j], end=" ")
    print()

```

```

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

```

index for Ganguly's run

```

for i in range(len(ganguly)):
    for j in range(len(ganguly[i])):
        print(i, j, end=" ")
    print()

```

```

0 0 0 1 0 2
1 0 1 1 1 2
2 0 2 1 2 2
3 0 3 1 3 2

```

```

for i in range(len(ganguly)):
    for j in range(len(ganguly[i])):
        print(sachin[i][j], end=" ")
    print()

```

```

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

```

```

for i in range(len(ganguly)):
    for j in range(len(ganguly[i])):
        print(ganguly[i][j], end=" ")
    print()

```

```

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

```

Adding 2 matrix

```

for i in range(len(ganguly)):
    for j in range(len(ganguly[i])):

```

```

        print(sachin[i][j], ganguly[i][j], end=" ")
    print()

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

for i in range(len(ganguly)): # This is for rows
    for j in range(len(ganguly[i])): # this is for columns
        print(sachin[i][j] + ganguly[i][j], end=" ")
    print()

0 4 6
8 0 12
14 16 0
20 22 24

```

Make a list after adding them

```

for i in range(len(ganguly)): # This is for rows
    row = []
    for j in range(len(ganguly[i])): # this is for columns
        total = sachin[i][j] + ganguly[i][j]
        row.append(total)
    print(row)

[0, 4, 6]
[8, 0, 12]
[14, 16, 0]
[20, 22, 24]

```

Final code

```

partnership = []
for i in range(len(ganguly)): # This is for rows
    row = []
    for j in range(len(ganguly[i])): # this is for columns
        total = sachin[i][j] + ganguly[i][j]
        row.append(total)
    #print(row)
    partnership.append(row)
print(partnership)

[[0, 4, 6], [8, 0, 12], [14, 16, 0], [20, 22, 24]]

```

Check for Identity Matrix

- You are given a $N \times N$ square integer matrix A. You have to tell whether A is an identity matrix or not.
- Identity matrix is a special square matrix whose main diagonal elements are equal to 1 and all other elements are 0.

Input:

- First and only argument is an integer matrix A.

Output:

- Return 1 if A is an identity matrix, else return 0.

```
A = [[1, 1],  
      [0, 1]]
```

Iterate on the list

```
for i in range(len(A)):  
    for j in range(len(A)):  
        print(i, j, end=" ")  
    print()
```

```
0 0 0 1  
1 0 1 1
```

print diagonal elements

A

```
[[1, 1], [0, 1]]
```

```
for i in range(len(A)):  
    for j in range(len(A)):  
        if i == j:  
            print(A[i][j])
```

```
1  
1
```

```
for i in range(len(A)):  
    for j in range(len(A)):  
        if i == j:  
            print(i, j)
```

```
0 0
1 1
```

A

```
[[1, 1], [0, 1]]
```

```
A = [[1, 0, 0],
      [0, 1, 0],
      [0, 0, 1]]
```

```
def identity(A):    # A is the list for which we have to check
    n = len(A)
    for i in range(n):
        for j in range(n):
            # Check for diagonal elements if it is 1 or not
            if i == j and A[i][j] != 1:
                return 0
            # check for non diagonal elements if they are 0 or not
            if i != j and A[i][j] != 0:
                return 0
    # After going through whole list if return is not hit then am
    Identity
    return 1

print(identity(A))

1
```

Doubts

```
B = [1, 2, 3]
A=[]
def f(x):
    return x*2
for i in B:
    A.append(f(i))
print(A)

[2, 4, 6]
```