

COSI 10A Recitation #7



Agenda: Lists

Review: Lists

Lists

object that **stores a list of values**

element: One value in a list.

index: A 0-based integer to access an element from a list.

<i>index</i>	0	1	2	3	4	5	6	7	8	9
	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
<i>value</i>	12	49	-2	26	5	17	-6	84	72	3

↑				↑					↑
element 0				element 4					element 9

Review: Lists

List initialization

name = [**value**, **value**, ... **value**]

numbers = [12, 49, -2, 26, 5, 17, -6]

<i>index</i>	0	1	2	3	4	5	6
<i>value</i>	12	49	-2	26	5	17	-6

Review: Lists

Accessing list elements

```
numbers = [0] * 8
```

```
numbers[0] = 3
```

```
numbers[1] = 99
```

```
numbers[2] = 6
```

<i>numbers</i>	<i>index</i>	0	1	2	3	4	5	6	7
	<i>value</i>	3	99	6	42	0	0	11	0

```
x = numbers[0]
```

```
numbers[x] = 42
```

```
numbers[numbers[2]] = 11 # use numbers[2] as index
```

Review: Lists

Lists and for loops

It is common to use for loops to access list elements

```
for i in range(0, 8):  
    print(str(numbers[i]) + " ", end='')  
print()  # output: 0 4 11 0 44 0 0 2
```

Sometimes we assign each element a value in a loop.

```
for i in range(0, 8):  
    numbers[i] = 2 * i
```

<i>index</i>	0	1	2	3	4	5	6	7
<i>value</i>	0	2	4	6	8	10	12	14

Review: Lists

Can a list have a list as an element?

YES!!

```
data = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
data = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
  
print(data[1][1])  
print()  
  
for i in range(len(data)):  
    for j in range(len(data[i])):  
        print(data[i][j], end=" ")  
    print()
```

5

1	2	3
4	5	6
7	8	9

Problem 1

P1. Write a function `calcDet` that calculates the determinant of a 2x2 matrix. The `calcDet` function needs to accept the 2x2 matrix and return the determinant of the given matrix. You may assume that 2x2 matrix always given to `calcDet` function.

$$|A| = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$$

Example Input:

Example Output:

Matrix:

Determinant: 10

[4 7]

[2 6]

<https://codeshare.io/NKNKrr>

P1. Write a function `calcDet` that calculates the determinant of a 2x2 matrix. The `calcDet` function needs to accept the 2x2 matrix and return the determinant of the given matrix. You may assume that 2x2 matrix always given to `calcDet` function.

$$|A| = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$$

```
def calcDet(matrix):  
    # Calculate the determinant of a 2x2 matrix  
    a, b = matrix[0][0], matrix[0][1]  
    c, d = matrix[1][0], matrix[1][1]  
    return a * d - b * c  
  
def main():  
    # Example 2x2 matrix  
    matrix = [  
        [4, 7],  
        [2, 6]  
    ]  
  
    # Calculate determinant  
    det = calcDet(matrix)  
    print(f"Determinant: {det}")  
  
# Run the main function  
main()
```


Problem 2

P2. Write a program that calculates the inverse of the given matrix. Your program should have at least three functions, `main`, `calcDet`, and `inverse`. You can use the `calcDet` from P1. The `inverse` function needs to accept the 2x2 matrix and return the inverse of the given matrix. You may assume that 2x2 matrix always given to `inverse` function.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Example Input:

Matrix:

[4 7]

[2 6]

Example Output:

Inverse Matrix:

[0.6, -0.7]

[-0.2, 0.4]

P2. Write a program that calculates the inverse of the given matrix. Your program should have at least three functions, `main`, `calcDet`, and `inverse`. You can use the `calcDet` from P1. The `inverse` function needs to accept the 2x2 matrix and return the inverse of the given matrix. You may assume that 2x2 matrix always given to `inverse` function.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

```
def inverse(matrix):
    # Calculate the inverse of a 2x2 matrix using calcDet
    det = calcDet(matrix)

    a, b = matrix[0][0], matrix[0][1]
    c, d = matrix[1][0], matrix[1][1]
    # The inverse matrix formula
    inverse_matrix = [
        [d / det, -b / det],
        [-c / det, a / det]
    ]
    return inverse_matrix

def main():
    # Example 2x2 matrix
    matrix = [
        [4, 7],
        [2, 6]
    ]

    # Calculate inverse
    inv_matrix = inverse(matrix)
    print("Inverse Matrix:")
    for row in inv_matrix:
        print(row)

# Run the main function
main()
```

Questions?