Computer Graphics Lab - CS352

Weekly Assignment Report

Objective Questions

Question 1:

Answer- A) [5, 7, 9]

Question 2:

Answer- D) Both A and B

Question 3:

Answer- A) np.sqrt()

Question 4:

Answer- A) Adding two arrays of different shapes automatically

Question 5:

Answer- A) Computes the sum of each column

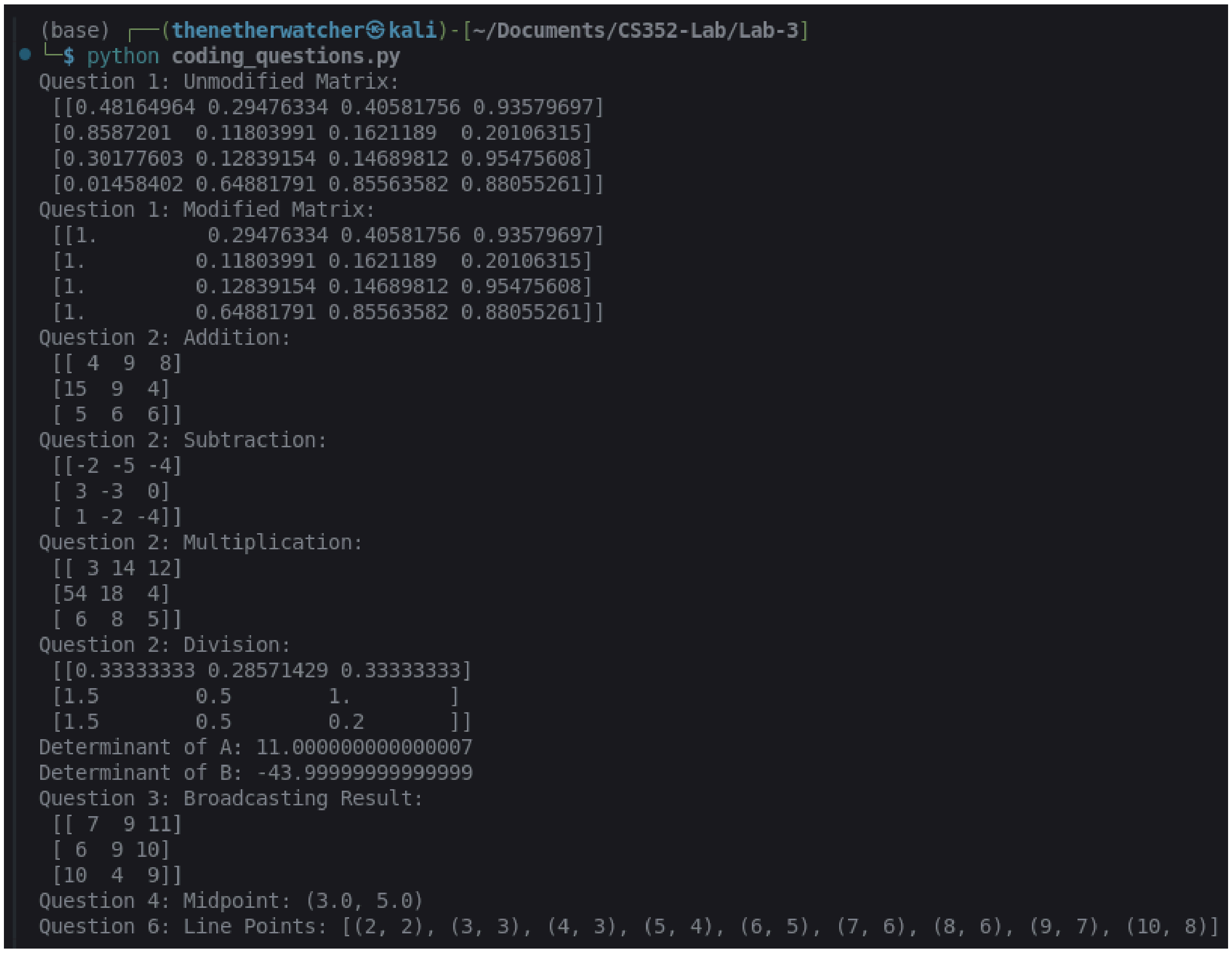
Coding Questions

Python

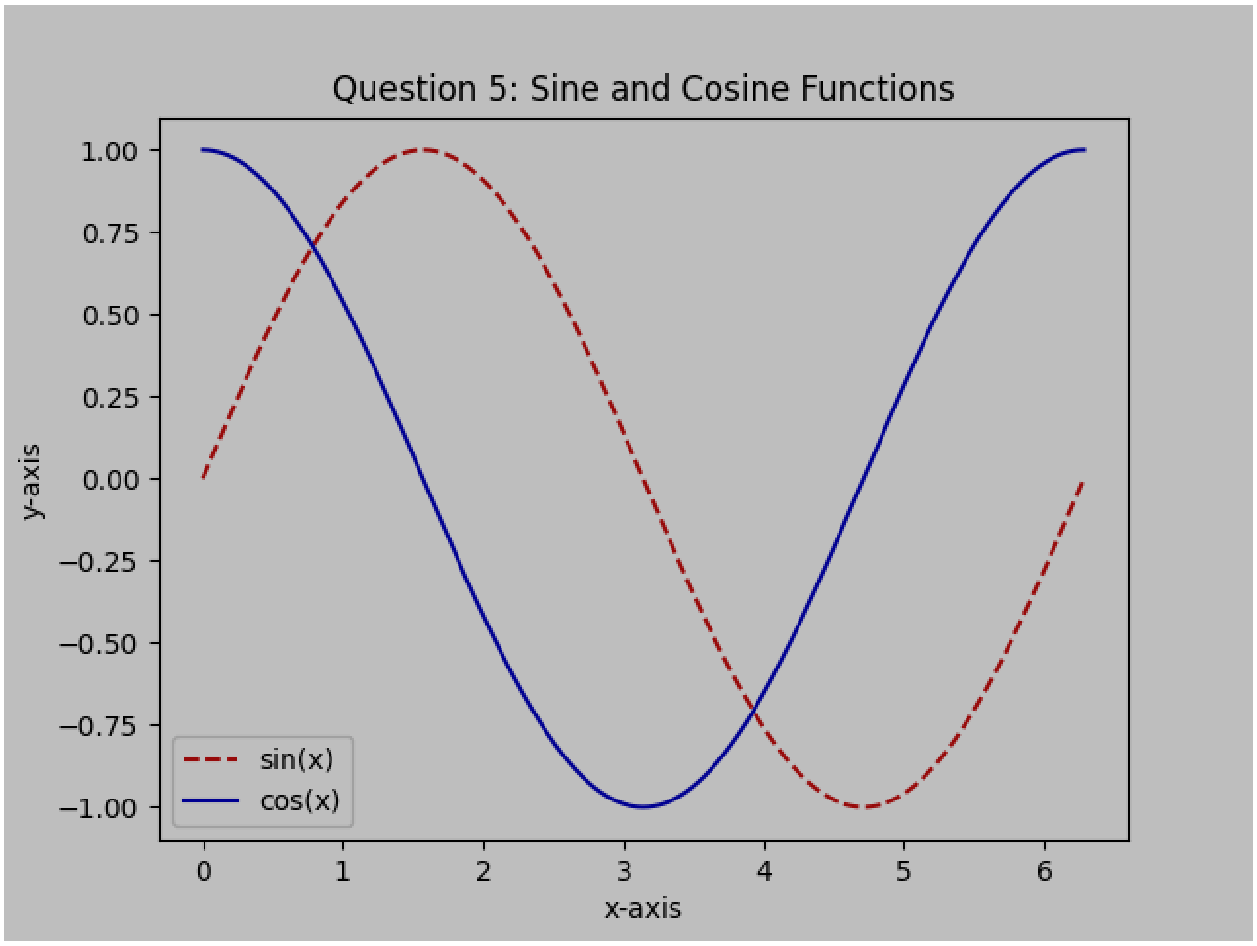
```python
import numpy as np

import matplotlib.pyplot as plt


# Question 1: Array Creation and Indexing

matrix = np.random.rand(4, 4)

print("Question 1: Unmodified Matrix:\n", matrix)

matrix[:, 0] = 1

print("Question 1: Modified Matrix:\n", matrix)


A = np.random.randint(1, 10, (3, 3))

B = np.random.randint(1, 10, (3, 3))


add = A + B

subtract = A - B

multiply = A * B

divide = A / B


print("Question 2: Addition:\n", add)

print("Question 2: Subtraction:\n", subtract)

print("Question 2: Multiplication:\n", multiply)
print("Question 2: Division:\n", divide)


print("Determinant of A:", np.linalg.det(A))

print("Determinant of B:", np.linalg.det(B))


# Question 3: Broadcasting in NumPy

matrix = np.random.randint(1, 10, (3, 3))

array = np.array([1, 2, 3])

result = matrix + array  # Broadcasting

print("Question 3: Broadcasting Result:\n", result)


# Question 4: Midpoint Theorem

def calculate_midpoint(x1, y1, x2, y2):

    Mx = (x1 + x2) / 2

    My = (y1 + y2) / 2

    return Mx, My


midpoint = calculate_midpoint(2, 3, 4, 7)

print("Question 4: Midpoint:", midpoint)


# Question 5: Plotting with Matplotlib

x = np.linspace(0, 2 * np.pi, 10000)

y_sin = np.sin(x)
```

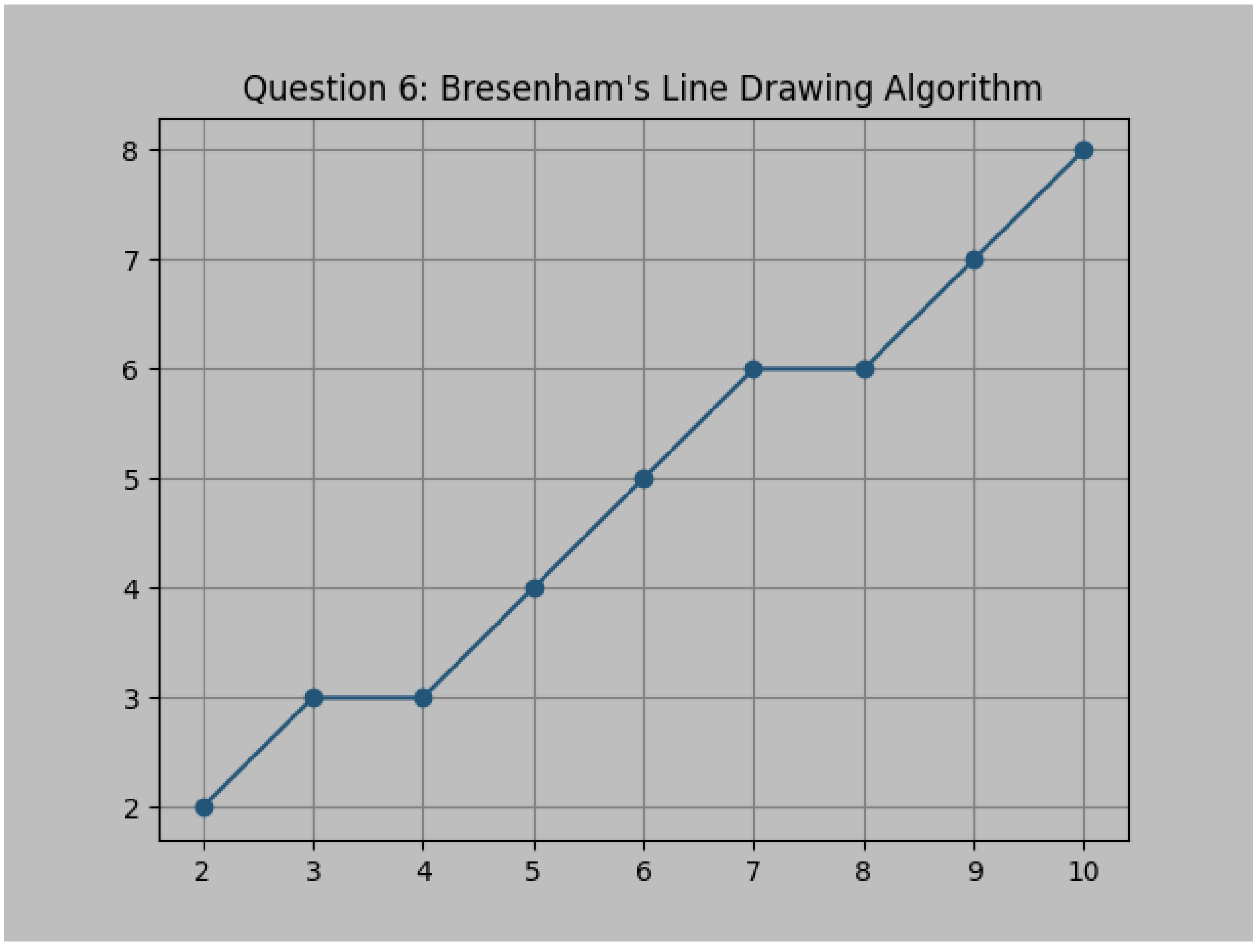
```
y_cos = np.cos(x)


plt.figure()

plt.plot(x, y_sin, 'r--', label='sin(x)')

plt.plot(x, y_cos, 'b-', label='cos(x)')   # Blue solid line with triangles

plt.xlabel('x-axis')

plt.ylabel('y-axis')

plt.title('Question 5: Sine and Cosine Functions')

plt.legend()

plt.show()


# Question 6: Bresenham's Line Drawing Algorithm

def bresenham(x1, y1, x2, y2):

    points = []

    dx = abs(x2 - x1)

    dy = abs(y2 - y1)

    sx = 1 if x1 < x2 else -1

    sy = 1 if y1 < y2 else -1

    err = dx - dy


    while True:

        points.append((x1, y1))

        if x1 == x2 and y1 == y2:
            break
        e2 = 2 * err
        if e2 > -dy:
            err -= dy
            x1 += sx
        if e2 < dx:
            err += dx
            y1 += sy

    return points


line_points = bresenham(2, 2, 10, 8)
print("Question 6: Line Points:", line_points)


# Plotting the line
x_coords, y_coords = zip(*line_points)
plt.figure()
plt.plot(x_coords, y_coords, marker='o')
plt.title("Question 6: Bresenham's Line Drawing Algorithm")
plt.grid(True)
plt.show()
```

Output:



```
(base) ┌──(thenetherwatcher㉿kali)-[~/Documents/CS352-Lab/Lab-3]
└─$ python coding_questions.py
Question 1: Unmodified Matrix:
 [[0.48164964 0.29476334 0.40581756 0.93579697]
 [0.8587201  0.11803991 0.1621189  0.20106315]
 [0.30177603 0.12839154 0.14689812 0.95475608]
 [0.01458402 0.64881791 0.85563582 0.88055261]]
Question 1: Modified Matrix:
 [[1.         0.29476334 0.40581756 0.93579697]
 [1.         0.11803991 0.1621189  0.20106315]
 [1.         0.12839154 0.14689812 0.95475608]
 [1.         0.64881791 0.85563582 0.88055261]]
Question 2: Addition:
 [[ 4  9  8]
 [15  9  4]
 [ 5  6  6]]
Question 2: Subtraction:
 [[-2 -5 -4]
 [ 3 -3  0]
 [ 1 -2 -4]]
Question 2: Multiplication:
 [[ 3 14 12]
 [54 18  4]
 [ 6  8  5]]
Question 2: Division:
 [[0.33333333 0.28571429 0.33333333]
 [1.5        0.5        1.        ]
 [1.5        0.5        0.2       ]]
Determinant of A: 11.000000000000007
Determinant of B: -43.99999999999999
Question 3: Broadcasting Result:
 [[ 7  9 11]
 [ 6  9 10]
 [10  4  9]]
Question 4: Midpoint: (3.0, 5.0)
Question 6: Line Points: [(2, 2), (3, 3), (4, 3), (5, 4), (6, 5), (7, 6), (8, 6), (9, 7), (10, 8)]
```


Question 5: Sine and Cosine Functions
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
y-axis
0
1
2
3
4
5
6
x-axis
sin(x)
cos(x)


Question 6: Bresenham's Line Drawing Algorithm
8
7
6
5
4
3
2
2
3
4
5
6
7
8
9
10