Exam 2 – Review Problems – Part I

Instructions. There are two parts to these review problems. You should attempt the problems in this part (Part I) without a computer. Part II is in a Jupyter Notebook. Feel free to use your class materials and the course website to attempt the problems in Part II.

Problem 1. Suppose that we made the following class:

```
import numpy as np

class MatrixPairs:
    """Here we make a class whose objects consist of pairs of matrices."""
    def __init__(self, A, B):
        self.A = A
        self.B = B

    def mult(self):
        """Here we make a method to multiply the matrices."""
        return self.A * self.B

    def add(self):
        """Here we make a method to add matrices."""
        return self.A + self.B
```

What would happen if we ran the following:

a. `mypair1 = MatrixPairs(np.array([[1, 2], [2, 3]]), np.array([[1, -2], [-2, 3]]))`
`mypair1.mult()` (Is this matrix multiplication?)

b. `mypair2 = MatrixPairs(2, 3)`
`mypair2.mult()`

c. `mypair1.add()`

d. `print(MatrixPairs.__doc__)` (We didn't cover this in class. Can you guess?)

Problem 2. Suppose you ran the following code that makes a new file. After each block of code, what would be written in that file?

a.

```
end = 5
filename = 'squares.txt'
with open(filename, 'w') as file_object:
    for i in range(1, end + 1):
        a = str(i ** 2)
        file_object.write(f"{a}  ")
```

b.

```
filename = 'squares.txt'
with open(filename, 'a') as file_object:
    for i in range(end + 1, 2 * end + 1):
        a = str(i ** 2)
        file_object.write(f"{a}  ")
```

c.

```
filename = 'squares.txt'
with open(filename, 'w') as file_object:
    file_object.write(f"\nThe squares up to {2*end}.")
```

Problem 3. A spreadsheet is made using the code below. After you run the code, what is in the spreadsheet?

```
import xlwings as xw

wb = xw.Book()
sht = wb.sheets['Sheet1']
sht.range('A1').value = 21

for i in range(2, 23):
    j = i - 1
    sht.range(f'A{i}').value = sht.range(f'A{j}').value + 1
```

Problem 4. What is the result of the following code?

```
import matplotlib.pyplot as plt
import matplotlib.image as img

M = np.zeros((100, 100, 3), dtype='uint8')

for i in range(0, 50):
    for j in range(0, 100):
        M[i, j, 0] = 255

plt.imshow(M)
```

Problem 5. A permutation matrix is a square matrix that has exactly one 1 in each row and column and the rest are zeros. It's a fact that some power of any permutation matrix is the identity matrix. Below is code that will compute the minimal power needed to produce the identity matrix for a given permutation matrix. Point out all three errors in the code.

Note. Two of the errors involve things we didn't explicitly cover in class.

```
import numpy as np

A = np.zeros([4, 4])
A[0, [1]] = 1
A[1, [3]] = 1
A[2, [0]] = 1
A[3, [2]] = 1

i = 1
I4 = np.eye(4)
while True:
    B = A ** i
    if B == I4:
        print('The minimal power to get the identity is i.')
        break
    else:
        i += 1
```