SDM366 Spring 2024 **Homework # 1**
Assigned February 23, 2024 Due: Friday, March 1, 2024

---

1. *Python basics (15)*: If you are not familiar with python, you can refer to `https://www.pythontutorial.net/python-basics/` or other tutorial links for help. Then you should be able to complete the following questions:

   (a) Write a program to display the current date and time. (5)

   (b) Write a program to print a specified list after removing the 0th, 4th and 5th elements. The given list is ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']. (5)

   (c) Define a class called *Student* that includes the student's name and age information. In addition, you should provide a method to display these information. Create an object for this class and call the display method. (5)

Solution:

Listing 1: (a)

```
import time

current_time = time.localtime()
formatted_time = time.strftime("%Y-%m-%d %H:%M:%S", current_time)
print("Current date and time:", formatted_time)
```

Listing 2: (b)

```
colors = ["Red", "Green", "White", "Black", "Pink", "Yellow"]
indexes_to_remove = {0, 4, 5}
colors = [item for idx, item in enumerate(colors) if idx not in
    indexes_to_remove]
print(colors)
```

Listing 3: (c)

```
class Student:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def display_info(self):
        print(f"name: {self.name}")
        print(f"age: {self.age}")

student = Student('Ray', 21)
student.display_info()
```

2. *Array Manipulation and Basic Numpy Operations (20)*: Perform a series of operations using NumPy to demonstrate understanding of array manipulation techniques including reshaping, sorting, slicing, and so on. You can refer to `https://numpy.org/doc/stable/user/absolute_beginners.html` for help.

   (a) **Reshape (5)**: Create a 1D array with elements from 1 to 10, and then reshape it into a 2x5 matrix. Print the result.

   (b) **Slice (5)**: Slice the 2x5 matrix from (a), consisting of the last two rows and the last two columns. Print the result.

   (c) **Sort (5)**: Sort the given array in descending order. The given array is [2, 1, 5, 3, 7, 4, 6, 8]. Print the result.

   (d) **Insert (5)**: Given an array [1, 2, 4, 5], insert integer 3 between 2 and 4, and append 6 at the end of the array. Print the result.

   Solution:

Listing 4: (a)

```
import numpy as np

arr_a = np.arange(1, 11)
arr_a = arr_a.reshape((2, 5))
print(arr_a)
```

Listing 5: (b)

```
arr_b = arr_a[-2:, -2:]
print(arr_b)
```

Listing 6: (c)

```
arr_c = [2, 1, 5, 3, 7, 4, 6, 8]
arr_c.sort(reverse=True)
print(arr_c)
```

Listing 7: (d)

```
arr_d = np.array([1, 2, 4, 5])
arr_d = np.insert(arr_d, 2, 3)
arr_d = np.append(arr_d, 6)
print(arr_d)
```

3. *Linear Algebra (45)*: In this class, it is important to use Python to complete the linear algebra task. Let's get familiar with it now.

$$A = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 2 & 2 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} -2 & -1 & 1 \\ 1 & 5 & 4 \\ 1 & -1 & -2 \\ 1 & 2 & 1 \end{bmatrix}$$

(a) Print the two matrices A and B. (5)

(b) Print the second row of A and the third column of B. (10)

(c) Print the results of A + B and A - B. (10)

(d) Construct a new 4 × 6 matrix [A, B] by appending B to the right of matrix A. (10)

(e) Compute $A^T B$ (9)

Solution:

Listing 8: (a)

```
A = np.array([[1, -1, 0],
              [1, 2, 2],
              [-1, 0, -1],
              [0, 1, 0]])

B = np.array([[-2, -1, 1],
              [1, 5, 4],
              [1, -1, -2],
              [1, 2, 1]])

print(A)
print(B)
```

Listing 9: (b)

```
print(A[1, :])
print(B[:, 2])
```

Listing 10: (c)

```
print(A + B)
print(A - B)
```

Listing 11: (d)

```
matrix_AB = np.hstack((A, B))
print(matrix_AB)
```

Listing 12: (e)

```
print(A.T @ B)
```

4. *Matplotlib (20)*: Plot a unit circle, and then plot 10 plus signs "+" uniformly distributed on the unit circle. Show the result.

Solution:

Listing 13: problem 4

```
import numpy as np
import matplotlib.pyplot as plt

theta = np.linspace(0, 2 * np.pi, 100)
x = np.cos(theta)
y = np.sin(theta)

num_points = 10
theta_points = np.linspace(0, 2 * np.pi, num_points, endpoint=False)
x_points = np.cos(theta_points)
y_points = np.sin(theta_points)

plt.figure(figsize=(6, 6))
plt.plot(x, y, label='Unit Circle')

plt.scatter(x_points, y_points, color='red', s=100, marker='+', label=
    'Points')

plt.xlabel('X')
plt.ylabel('Y')
plt.title('Unit Circle with Uniformly Distributed Points')
plt.legend()
plt.grid(True)
plt.axis('equal')

plt.show()
```