

## Lab 10 Multi-dimensional Lists

Please finish the following program design and submit via the platform **before Dec. 5th** and the source code should be saved as the name **StudentID\_Name\_10\_1.py**, such as 2023333519001\_王澜洁\_10\_1.py, 2023333519001\_王澜洁\_10\_2.py, ... etc. and then compress them with the name **StudentID\_Name\_10.rar** (or **StudentID\_Name\_10.zip**) such as 2023333519001\_王澜洁\_10.rar (or 2023333519001\_王澜洁\_10.zip).

1. (Sum elements column by column) Write a function that returns the sum of all the elements in a specified column in a matrix using the following header:

**def sumColumn(m, columnIndex):**

Write a test program that reads a  $3 \times 4$  matrix and displays the sum of each column. Here is a sample run:

```
Enter a 3-by-4 matrix row 0: 1.5 2 3 4
Enter a 3-by-4 matrix row 1: 5.5 6 7 8
Enter a 3-by-4 matrix row 2: 9.5 1 3 1
Sum of the elements at column 0 is 16.5
Sum of the elements at column 1 is 9
Sum of the elements at column 2 is 13
Sum of the elements at column 3 is 13
```

2. (Sum the major diagonal in a matrix) Write a function that sums all the numbers of the major diagonal in an matrix of integers using the following header:

**def sumMajorDiagonal(m):**

The major diagonal is the diagonal that runs from the top left corner to the bottom right corner in the square matrix. Write a test program that reads a matrix and displays the sum of all its elements on the major diagonal. Here is a sample run:

```
Enter a 4-by-4 matrix row 0: 1 2 3 4
Enter a 4-by-4 matrix row 1: 5 6.5 7 8
Enter a 4-by-4 matrix row 2: 9 10 11 12
Enter a 4-by-4 matrix row 3: 13 14 15 16
Sum of the elements in the major diagonal is 34.5
```

3. (Largest rows and columns) Write a program that randomly fills in 0s and 1s into a matrix, prints the matrix, and finds the rows and columns with the most 1s. Here is a sample run of the program:

```
1 1 1 0
0 1 1 1
1 1 1 1
1 0 1 0
The largest row index: 2
The largest column index: 2
```

4. (Explore matrix) Write a program that prompts the user to enter the length of a square matrix, randomly fills in 0s and 1s into the matrix, prints the matrix, and finds the rows, columns, and major diagonal with all 0s or all 1s. Here is several sample runs of the program:

```
Enter the length of a square matrix: 4
0001
1010
0010
0100
No same numbers on a row
No same numbers on a column
No same numbers on the major diagonal
No same numbers on the sub-diagonal
```

```
Enter the length of a square matrix: 4
1101
0101
1101
1000
All 0's on column 2
No same numbers on a row
No same numbers on the major diagonal
No same numbers on the sub-diagonal
```

```
Enter the length of a square matrix: 3
000
100
010
All 0's on row 0
All 0's on column 2
All 0's on major diagonal
All 0's on sub-diagonal
```

5. (Row sorting) Implement the following function to sort the rows in a twodimensional list. A new list is returned and the original list is intact.

```
def sortRows(m):
```

Write a test program that prompts the user to enter a matrix of numbers and displays a new row-sorted matrix. Here is a sample run:

Enter a 3 by 3 matrix row by row:

0.15 0.875 0.375

0.55 0.005 0.225

0.30 0.12 0.4

The row-sorted list is

0.15 0.375 0.875

0.005 0.225 0.55

0.12 0.3 0.4