Universidade Federal de Itajubá – UNIFEI

ECOE41 – Tópicos Especiais em Programação I

Introduction to Python

Classroom work

This work consists of the implementation of some methods for solving mathematical problems.

All functions must be developed in the same file with the main code that will be made available.

To make it clear how functions should be added to the main program file, the following example illustrates the implementation of a function that calculates the averages of several integers numbers. The function is already contextualized in the code provided.

```
'''
  Template for implementation of the work
'''

def mean(l):
  total=0
  for i in l:
    total += i
  return (float(total) / len(l))

'''
  Write your functions here!!!
'''


'''
  Main code
'''

# Fist problem: function mean

List=input().split(" ")
Numbers=list()

for elem in List:
  Numbers.append(int(elem))

print("%.1f\n" % (mean(Numbers)))
```

You have to write one (or more, if necessary) function for each problem that will be asked below. Notice that there is a place for functions (*Put your functions here!!!*) and a place for calling these functions (*Main code*).

The inputs must follow the pattern set in the examples. The first example illustrate this and is already done in the template file.

Be careful of the input and output patterns. It is very important that these patterns are respected. This will be one of the work marking criteria.

Write functions for:

01) Calculate the mean of the list of integers.

Input example:

```
13 18 13 14 13 16 14 21 13
```

Expected output:

```
15.0
```

02) Calculate the median of the list of integers. The median is the middle value, so first you will have to rewrite the list in numerical order.

Input example:

```
13 18 13 14 13 16 14 21 13
```

Expected output:

```
14.0
```

03) Calculate the mode of the list of integers. The mode is the number that is repeated more often than any other.

Input example:

```
13 18 13 14 13 16 14 21 13
```

Expected output:

```
13
```

04) Calculate the range of the list of integers. The range is the difference of the smallest value from the largest value in the list.

Input example:

```
13 18 13 14 13 16 14 21 13
```

Expected output:

```
8
```

05) Calculate the sum of two square matrices. Each matrix is represented by a list of list of integers.

For example, the matrix: $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

will be represented by: [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

Input example: (The first line of the input represents the matrix order)

```
3
1 2 3
4 5 6
7 8 9
1 1 1
1 1 1
1 1 1
```

Expected output:

```
2 3 4
5 6 7
8 9 10
```

06) Calculate the subtraction of two square matrices. Each matrix is represented by a list of list of integers.

Input example: (The first line of the input represents the matrix order)

```
3
1 2 3
4 5 6
7 8 9
1 1 1
1 1 1
1 1 1
```

Expected output:

```
0 1 2
3 4 5
6 7 8
```

07) Calculate the multiplication of two square matrices. Each matrix is represented by a list of list of integers.

Input example: (The first line of the input represents the matrix order)

```
2
1 2
3 4
1 2
3 4
```

Expected output:

```
7 10
15 22
```

08) Calculate the determinant of one square matrix of order 2. (Matrix of float numbers)

Input example:

```
1.0 2.0
-5.0 -3.0
```

Expected output:

```
7.0
```

09) Calculate the determinant of one square matrix of order 3. (Matrix of float numbers)

Input example:

```
5.0 0.0 1.0
-2.0 3.0 4.0
0.0 2.0 -1.0
```

Expected output:

```
-59.0
```

10) Calculate the determinant of one square matrix of order 4. (Matrix of float numbers)

Input example:

```
1.0 2.0 3.0 4.0
-1.0 0.0 3.0 5.0
5.0 6.0 2.0 3.0
5.0 -3.0 -5.0 -2.0
```

Expected output:

```
-111.0
```

11) Calculate the transpose of one matrix. (Matrix of float numbers)

For example, the transpose of the matrix: $\begin{bmatrix} 1.0 & 2.0 & 3.0 \\ 4.0 & 5.0 & 6.0 \\ 7.0 & 8.0 & 9.0 \end{bmatrix}$ is: $\begin{bmatrix} 1.0 & 4.0 & 7.0 \\ 2.0 & 5.0 & 8.0 \\ 3.0 & 6.0 & 9.0 \end{bmatrix}$

Input example: (The first line of the input represents the matrix order)

```
3
1.0 2.0 3.0
4.0 5.0 6.0
7.0 8.0 9.0
```

Expected output:

```
1.0 4.0 7.0
2.0 5.0 8.0
3.0 6.0 9.0
```

12) Calculate the cofactor matrix of a give square matrix with order less than or equal to 4. (Matrix of float numbers)

For example, the cofactor matrix of: $\begin{bmatrix} 1.0 & 2.0 & 3.0 \\ 4.0 & 5.0 & 6.0 \\ 7.0 & 8.0 & 9.0 \end{bmatrix}$ is: $\begin{bmatrix} 1.0 & 4.0 & 7.0 \\ 2.0 & 5.0 & 8.0 \\ 3.0 & 6.0 & 9.0 \end{bmatrix}$

Input example: (The first line of the input represents the matrix order)

```
3
1.0 2.0 3.0
0.0 4.0 5.0
1.0 0.0 6.0
```

Expected output:

```
24.0 5.0 -4.0
-12.0 3.0 2.0
-2.0 -5.0 4.0
```

13) Calculate the adjoint matrix of a give square matrix with order less than or equal to 4. (Matrix of float numbers)

For example, the transpose of the matrix: $\begin{bmatrix} 1.0 & 2.0 & 0.0 \\ 2.0 & 3.0 & 1.0 \\ 0.0 & 0.0 & 2.0 \end{bmatrix}$ is: $\begin{bmatrix} 6.0 & -4.0 & 2.0 \\ -4.0 & 2.0 & -1.0 \\ 0.0 & 0.0 & -1.0 \end{bmatrix}$

Input example: (The first line of the input represents the matrix order)

```
3
1.0 2.0 0.0
2.0 3.0 1.0
0.0 0.0 2.0
```

Expected output:

```
6.0 -4.0 2.0
-4.0 2.0 -1.0
0.0 0.0 -1.0
```

14) Calculate the inverse of one matrix with order less than or equal to 4. (Matrix of float numbers).

For example, the transpose of the matrix: $\begin{bmatrix} 2.0 & 1.0 \\ 5.0 & 3.0 \end{bmatrix}$ is: $\begin{bmatrix} 3.0 & -1.0 \\ -5.0 & 2.0 \end{bmatrix}$

All examples are inversible.

Input example: (The first line of the input represents the matrix order)

```
2
2.0 1.0
5.0 3.0
```

Expected output:

```
3.0 -1.0
-5.0 2.0
```

15) Using matrix representation of a Linear System and Gaussian Elimination (Gauss method), calculate the system of linear equations. (Matrix of float numbers).

For example, the soluction of the system: $\begin{aligned} x+2y+3z &= 4 \\ 5x+6y+7z &= 8 \\ 9x+10y+11z &= 12 \end{aligned}$ is: $[1,-3,2]$

All examples there is a single solution, that is, *Consistent Independent System*.

Input example: (The first line of the input represents the number of variables of the system. All the coefficients are integers, but the solution is a vector of float numbers)

```
3
1 2 3 4
5 6 7 8
9 10 11 12
```

Expected output:

```
1 -3 2
```