Programming for Data Science (13/1/2025)

Upload the solutions to the programming exercises to the following link:
https://evo.di.unipi.it/student/courses/16/exams/3PqQBxV

Exercise 1. (Math 1, on paper)

Let $f : \mathbb{R} \to \mathbb{R}$ be a function defined as:

$$f(x) = \begin{cases} 2x + 1 & \text{if } x \geq 0, \\ -3x + 4 & \text{if } x < 0. \end{cases}$$

1. Determine if $f(x)$ is injective.
2. Determine if $f(x)$ is surjective.

Exercise 2. (Math 2, on paper)

Let A be a 3×3 matrix defined as:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}.$$

1. How many ways can you select an ordered list of three elements from A such that: no two elements are from the same row; no two elements are from the same column.
2. Calculate the determinant of A
3. Convert A in upper triangular form through Gaussian elimination (at least the first step of the procedure)

Exercise 3. (Python 1) Write a Python program that reads a sequence of integers S from the terminal. The sequence is to be read number by number from the terminal until the user inserts a -1. Implement the following three functions over the sequence:

1. **Longest Subsequence**: Determine the length of the longest subsequence where all numbers are either strictly increasing or strictly decreasing.
2. **Subsequence Sum**: Compute the maximum sum of any contiguous (and even overlapping) subsequence in S.
3. **Median Value**: Find the median value of the sequence S (if the length is odd, the median is the middle value; if even, it's the average of the two middle values).

Constraints:

- The sequence S contains between 1 and 20 integers.
- Consecutive integers that are equal cannot belong to the same subsequence. In this case, an error message has to be written on the terminal and the last inserted number has to be skipped.

Example Input: 10 5 2 3 8 6 4 9 7
Example Output:

- Longest subsequence length: 5
- Maximum sum of contiguous subsequence: 77 (10,5,2; 2,3,8; 8,6,4; 4,9; 9,7)
- Median value: 6

Exercise 4. (Python 2) Write a Python program to analyze the frequency of characters in a given string. Take a string s as input from the terminal. Assume the string contains only lowercase letters and is non-empty. Implement the solution using both an iterative and a recursive approach, and make use of a dictionary to store the frequency counts.

Constraints:

- The string length |s| is between 1 and 100.
- Both implementations (iterative and recursive) must return a dictionary where keys are characters, and values are their frequencies.