University of Science and Technology of Hanoi *** **Final Exam** **Algorithms and Data Structures** Sheet: 01 No of pages: 02		***Academic year**: 2024–2025* ***Date**: 08/01/2025* ***Time**: 75 minutes* ***Important instructions*** *(according to lecturer's decision)* 1. No documents or communication devices are allowed. 2. Copying or using Internet will lead to heavy penalty	
Pathway	CS, DS, MATH, and MST	**Lecturer**	**Dr. Đoàn Nhật Quang**
Student name		**Student's ID**	

Follow this instruction:
- Create a folder "YOURNAME_STUDENTID" in the Desktop.
*- Create the source files **question1.c** (or cpp) and **question2.c** for the corresponding problems.*
*- **Remove the executable files** (.exe) and **zip** all your source codes, and submit it in Google classroom. Submitting executable files or inappropriate files leads to 0.*

Question 1 (10 pts**)**
In this problem, a natural number is given. The objective is to find a digit and swap its position in the number to obtain the smallest possible value.
Note: The smallest value must have the same number of digits as in the original value; thus, 0 cannot be used.
- Example: 51029 -> the possible smallest value is 15029 after one swap.
- Propose an algorithm using **iteration** and implement it in C/C++ to complete the task. (3 pts)
- Propose and implement an alternative algorithm using **recursion** for this task. (6 pts)
- Calculate the complexity of the two approaches. Justify your answer by making comments on the code. (1 pt)

Hint: You can use an array to store all digits. The problem becomes sorting and searching on an array.

Question 2 (7 pts)
In this problem, we design and implement data structures in C/C++ to work on a list of 3D vectors.
- Propose a data structure for a 3D vector (x, y, z) with x, y, z are real. (1 pt)
- Propose a List data structure to store a collection of 3D vectors (the list capacity, n, can be pre-defined). (1 pt)
- Write a function to calculate the Euclidean distance between a pair of vectors in the list. (2 pts)
  Given two vectors: $v_1 = (x_1, y_1, z_1)$ and $v_2 = (x_2, y_2, z_2)$ then their distance is expressed by:
  $$d(v_1, v_2) = \sqrt{(x_1 - x_2)^2 + (y - y_2)^2 + (z_1 - z_2)^2}$$
- Write a function to display the vectors in a list and the distance matrix. (2 pts)
  Note: the distance matrix is square and of the size n × n. Each element of the matrix (i,j) is the distance between two vectors v_i and v_j.
- Write a main function to test all the above functions. (1 pt)

Question 3 (3 pts**)**

Odd-even sorting is a variant of Bubble Sort such that by comparing all odd/even indexed pairs of adjacent elements in the list, if a pair is in the wrong order (the first is larger than the second), the elements are switched. The pseudo-code is as follows:

```
Algorithm: Odd-even sorting
Input: an array arr of the length n
sorted = 0;
while (sorted == 0) {
  sorted = 1;
  for (i = 1; i<n-1; i+=2) {
    if (arr[i] > arr[i+1]) {
      swap arr[i] and arr[i+1];
      sorted = 0;
    }
  }
  for (i = 0; i<n-1; i+=2) {
    if (arr[i] > arr[i+1]) {
      swap arr[i] and arr[i+1];
      sorted = 0;
    }
  }
}
```

- Initialize an array of the size $n \geq 6$. Array values are initialized based on your choice.
- Explain the sorting process and display the results at each iteration of the sorting process.

Note: This question does not require any implementation in C/C++. You need to make comments on your file.

--END --