ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

LAB 1 **SUMMER SEMESTER, 2021-2022**

CSE 4810: Algorithm Engineering

1) In the following code, find the asymptotic complexity of the function "**function()**". Clearly explain how you arrived to your solution.

```
void function(string a, int l, int r)
{
    if (l == r)
        cout<<a<<endl;
    else
    {
        for (int i = l; i <= r; i++)
        {
            swap(a[l], a[i]);
            function(a, l+1, r);
            swap(a[l], a[i]);
        }
    }
}
int main()
{
    string ss = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    int n = ss.size();
    function(ss, 0, n-1);
    return 0;
}
```

2) Construct an algorithm to find whether a given point falls within a given triangle or not. Write it
in a clear and understandable manner in the form of a Pseudocode. After coming up with the solution find its asymptotic complexity.

3) Find the asymptotic complexity of the following code.

```
def process(numbers):
    # numbers is a list of numbers
    for i in range(len(numbers)-1):
        for j in range(len(numbers)-i-1):
            if numbers[j] > numbers[j+1]:
                numbers[j], numbers[j+1] = numbers[j+1], numbers[j]
    return numbers
```

4) Construct an $O(n^2)$ algorithm for the following problem. Can you optimize the algorithm to make it faster than $O(n^2)$? How?

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.

You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

Example 1:

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].
```

Example 2:

```
Input: nums = [3,2,4], target = 6
Output: [1,2]
```

Example 3:

```
Input: nums = [3,3], target = 6
Output: [0,1]
```

5) Construct an algorithm to check if two given rectangles overlap.