Data Structures and Algorithms Lab

SE-F22

LAB-03

Issue Date: February 22, 2024

Start Time: 8:45AM

Submission Time: 11:45AM

Total Marks: 50

The objective of this lab is to:

Determine the operational working of functions applied on polynomials and creating Array ADT and performing various operations using it.

Instructions:

1) Follow the question instructions very carefully, no changes in function prototypes are allowed.
2) You could solve the following growth functions on paper.
3) Anyone caught in an act of plagiarism would be awarded an "F" grade in this Lab.

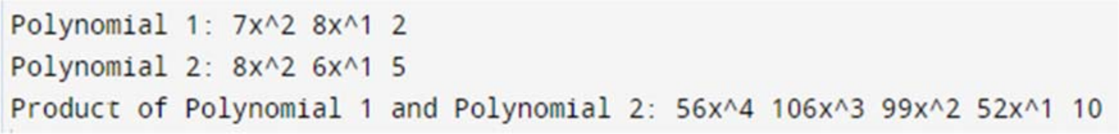
Task 01(Polynomial Multiplication) [15 Marks]

Assuming that you have created the Polynomial ADT as discussed in the class. Your task is to implement a function that will find the product to a polynomial with another polynomial.

Function Prototype:

```
Polynomial multiply(const Polynomial& other) const
```

Example:



```
Polynomial 1: 7x^2 8x^1 2
Polynomial 2: 8x^2 6x^1 5
Product of Polynomial 1 and Polynomial 2: 56x^4 106x^3 99x^2 52x^1 10
```

Task 02 (Array ADT) [10 Marks]

In this task you are required to implement the Array ADT. Following is the structure that you have to follow for your implementation:

```
class Array
{
private:
   T* data;
   int size;
   int capacity;
public:
   Array();
   Array(int size);
   ~Array();
   int getSize() const;
   void reSize(int newCapacity);
   T& operator[](int index);
   void insert(int index, const T& value);
   void remove(int index);
   void sort();
   void display();
};
```

Task 03 (Find Elements with Target Sum) [10 Marks]

You are given an array ‘nums’ of any size and an integer ‘k’. Your task is to return another array let’s say ‘result’ such that *result* contains the 2 elements from nums that sum up to the value of k.

Example:

Input: nums = {5,9,-9,28,3,4}, k = 8
Output: {5,3}

Function Prototype:

```
Array<T>  findElements(Array<T> nums, int k)
```

Task 04 (Maximum Sum Subarray) [15 Marks]

You are given an array ‘nums’. You are required to find the subarray that has the maximum sum in the whole array.

Example:

Input: { -2,1,-3,4,-1,2,1,-5,4}
Output: 6
Explanation: The subarray 4,-1,2,1 forms the maximum sum 6 that is highest than the sum of any other Subarray. Hence, output is 6.

Function Prototype:

```
int maximumSum(Array<T>  nums)
```

"Programming isn't about what you know; it's about what you can figure out."

– Chris Pine