CS218- Data Structures
Programming Assignment No. 2
Fall 2019

Instruction

This is the second programming assignment for the course CS218- Data Structures in the offering Fall 2019. The assignment comprises of two problems. It is suggested that you should start working on the assignment at your earliest. This seems a good amount of intellectual work required to complete it. Each question should be solved in one program file names as per suggested scheme. Your student number dash assignment number and problem number, e.g K182122-A2P1.cpp
Your program should take input from the input file as per the direction of the input format. The output should be produced in an output file as per the required format. These two files should be read from the current directly where your source code is residing.
The assignment is for individual and there should not be any case of cheating. You can have discussion about any problem and approach among yourself but do not share code and instruction for any problem.

Due Date: November 02, 2019 21:00PM

Problem No. 1: Polynomial equation

Polynomial equations are equations that have multiple terms made up of numbers and variables. You are required to design a solution in order to represent polynomial for computer manipulation.
The main requirement is to make use of linked list to hold each term of the polynomial for example:

Linked List1: $4X^7+2X^6-X^3+4X^2+3=0$
Linked List2: $2X^{8+}3X^3-2=0$
Resultant linked list: $2X^8+4X^{7+}2X^6+2X^3+4X^2+1=0$

For above given list1 the first node of the list1 contains a structure of Co-efficient= +4, Base= X and Exponent = 7 and in the very same fashion rest of the terms are stored. The polynomial linked list is always stored in descending order of power and in single variable (Base).

a. You are required to write a program that can add two instances of this lists and store its result in another list. Within the input file each list equation is entered. You have to read each equation and form a linked list of them first; then form resultant linked list by adding the equations up.

INPUT FILE	**OUTPUT FILE**
$4X^7-2X^6-X^3+4X^2+3=0$ $2X^6+3X^2-2=0$	$4X^7-X^3+7X^2+1=0$

b. Differentiation is the action of computing a derivative. The derivative of a function y = f(x) of a variable x is a measure of the rate at which the value y of the function changes with respect to the change of the variable x. Next you are required to calculate the derivative of the resultant linked list created in above part. The output file created in above part will be used as input file to read off for this part.

INPUT FILE	**OUTPUT FILE**
$4X^7-X^3+7X^2+1=0$	$28X^6-3X^2+14X=0$

Problem No. 2 Largest Palindrome in a Linked List

A palindrome is a word, phrase, number or sequence of words that reads the same backward as forward. Punctuation and spaces between the words or lettering is allowed. Hence there are multiple categories of Palindromes. For example: Single Word Palindrome like Madam, Multiple Words Palindromes like: “Step on no pets” and “was it a cat I saw” Number Palindromes like 191 and 123321. Given an Integer Array of some content, an array palindrome is defined as the sub-array whose contents concatenated form a palindrome. For example, A [5] = {1, 21,12,1,9} contains a palindrome from index 0-3 – by seeing 121121 together. Similarly, we can have a palindrome in a linked list, example given below:

Linked List L1(Head) → 12451 → 11342 → 11211 → 4318 → 12 → 1 →X

You need to find the largest such palindrome possible in the content of the linked list, assume the singly linked list is used to store content. You program should decide about the longest possible palindrome.

Input Format

The input file gives you, number of nodes in the singly linked list as positive integer (n), the next lines n lines give you integers to hold in the n nodes. For the given example there are 6 nodes in the given list.

```
6
12451
11342
11211
4318
12
1
```

Output Format

The output file contains the longest possible palindrome using the content of the list from any position to any other position within the node or spanning the nodes.

```
11211
```