# Department of Computer Science

**CS250: Data Structure and Algorithms**

**Class: BSCS-10C**

# Lab 4: Practical Application of Linked List

**Open-ended Lab**

**Date: 08th October 2021**

**Time: 09am - 12:00pm**

# Instructor: Prof. Dr. Faisal Shafait Lab Engineer : Aftab Farooq

# Lab 4: Singly Linked List & its Practical Implementation

**Introduction**

This lab will introduce students with the practical implementation of linked list with its operations.

**Objectives**

This Lab Task will introduce students with the practical application of linked list with its operations.

**Tools/Software Requirement**

Visual Studio C++

**Description**

**Singly Linked List as implemented in Lab 03**

A Linked List, is a data structure consisting of a group of nodes which together represent a sequence. Under the simplest form, each node is composed of two parts i.e. data part and a reference part (also known as, a link) to the next node in the sequence. This structure allows efficient insertion or removal of elements from any position in the sequence.

## Singly-linked-list.svg

## The basic operation consists of

* Creating the list.
* Initialize pointers to NULL.
* Inserting nodes at beginning, last and specified location.
* Delete nodes from beginning, last and specified location.
* Traversing the list.
* Destroying the list.

**Representing a polynomial using a linked list**

A polynomial can be represented in an array or in a linked list by simply storing the coefficient and exponent of each term. However, for any polynomial operation, such as addition or multiplication of polynomials, you will find that the linked list representation is more easier to deal with. First of all note that in a polynomial all the terms may not be present, especially if it is going to be a very high order polynomial. Consider,



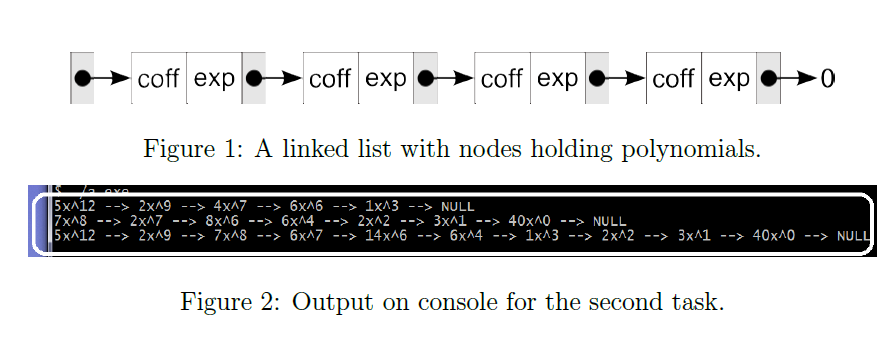
Now this 12th order polynomial does not have all the 13 terms (including the constant term). It would be very easy to represent the polynomial using a linked list structure, where each node can hold information pertaining to a single term of the polynomial.

Each node will need to store,

- the exponent, and

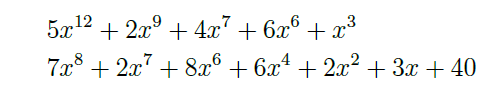
- the coefficient for each term.

Thus, we need to define a node class to hold two integers , exp and coff.



**Lab Task:**

Addition of two polynomials. Consider addition of the following polynomials,



The resulting polynomial is going to be,



Now notice how the addition was carried out. Let us say the result of addition is going to be stored in a third list.

\_ we start with the highest power in any polynomial.

\_ if there is no item having the same exponent, we simply append the term to the new list, and continue with the process.

\_ if we found that the exponents are matching, we simply add the coefficients and then store the term in the new list.

\_ If one list gets exhausted earlier and the other list still contains some lower order terms, then simply append the remaining terms to the new list.

Execute the completed code, and the output should be something similar one in Figure 2.

**Lab Grading:**

|  |  |
| --- | --- |
| **Task** | **Marks** |
| Lab Viva/Quiz | 5 |
| Comments/ Identation | 2 |
| Solution Document | 2 |
| Output Screen Shots | 1 |
| -- | -- |
| Total | 10 |

**Deliverables**

Compile a single word document by filling in the solution part and submit this Word file on LMS. This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva related to the tasks. The viva has a weightage of 5 marks. Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems discuss it by emailing it to [aftab.farooq@seecs.edu.pk](mailto:aftab.farooq@seecs.edu.pk)

**Note:** Students are required to upload the lab on LMS before deadline.

Use proper indentation and comments. Lack of comments and indentation will result in deduction of marks.