

AM5801: Computational Lab

Assignment 2

Date: August 21, 2025

Deadline: August 31, 2025

Max mark: 50

---

1. Write a python program to represent a polynomial in a single variable x in the form of a linked list, and write functions to perform the following:

(a) *node \* read\_poly();*

Read a polynomial into a linked list, and return a pointer that points to the first element of the list. Every node of the link list will represent one term of the polynomial and will contain three fields: the power of x, the coefficient value, and a pointer to the next node. For example, consider the polynomial  $f(x) = 2x^5 + 3.2x^3 - 15$ . The linked list will contain three nodes with the values (5,2), (3,3.2), and (0,15). (5)

(b) *void print\_poly(node \* head);*

Print a polynomial in suitable formatted form. This function will take a single parameter as argument, which is a pointer to the first element of the list representing the polynomial. The output should be printed as:  $f(x) = 2x^5 + 3.2x^3 - 15$  (5)

(c) *node \* add\_poly(node \* head1, node \* head2);*

Add two polynomials pointed by the pointers head1 and head2, and return a new polynomial after adding them. (5)

(d) *node \* multiply\_poly(node \* head1, node \* head2);* Multiply two polynomials function to multiply two polynomials pointed by the pointers head1 and head2, and return the result. (10)

In the main function, declare a structure node representing a node of the linked list for the polynomial representation. Then read two polynomials P and Q and print the input polynomials, their sum (i.e., P+Q), and also their product. You must use the functions *read\_poly*, *print\_poly*, *add\_poly*, and *multiply\_poly* for the purpose.

2. A biochemistry laboratory processes samples from different experiments provided in the table. Write Python programs for the following tasks:

(a) Implement a queue data structure to manage sample processing order. The queue should support adding new samples, removing processed samples, and viewing the next sample to be processed. (8)

SampleID	Experiment	ProcessTime	Status
S001	ProteinFolding	45	Pending
S002	DNASequencing	120	Complete
S003	EnzymeAssay	30	Pending
S004	CellCulture	180	Processing
S005	ProteinFolding	60	Complete
S006	RNAExtraction	75	Pending
S007	EnzymeAssay	40	Processing
S008	DNASequencing	95	Complete

- (b) Create a hash table using linear probing for collision resolution to store samples indexed by SampleID. The hash table must handle insertion, search, and deletion operations. Use a table size of 11 and implement your own hash function. (10)
- (c) Develop a function that retrieves all samples belonging to a specific experiment type from your hash table and displays them in the order they were queued for processing. (7)