National Yang Ming Chiao Tung University   October 24, 2021
Department of Electrical Engineering   Instructor: Mark P.-H. Lin
Data Structures, Fall 2021   TA: Tina Hsiao

Programming Assignment #3

Linked Lists

1 Problem Description

Polynomials appear in many fields of mathematics, science, and engineering. For example, they are used in calculus and numerical analysis to approximate other functions. A polynomial is an expression consists of variables and coefficients, which involves at least the operations of addition, subtraction, and multiplication of variables. In this programming assignment, you are asked to implement these polynomial operations using the linked-list data structure.

2 Input Format

The input file gives the description of all the aforementioned polynomial operations, including *add*, *subtract*, and *multiply*. In addition to those common operations, two special operations, *append* and *delete*, are further added to manipulate and maintain the polynomials in use, which are stored in a list. The detailed input format of each operation is given below.

- **Append polynomial** [Operation ID: 0]
  The operation indicates that you will need to append a new polynomial to the end of your polynomial list. In addition to the operation ID, which is 0, it also provides the information of a three-variable polynomial, including the number of terms in the polynomial, followed by the coefficient and the exponents of the three variables, x, y, and z, in each term. Both coefficients and exponents are integers, where the coefficients range between $-10^6 - 10^6$, and the exponents range between $0 - 10^6$. The format of the append operation is illustrated below with an example on the right side.

Append Operation Format	Append
	$3x^2y - 2xy^4z^3 + 6$
0 [number of terms in the polynomial]	
[coefficient] [exponent of x] [exponent of y] [exponent of z]	0 3
[coefficient] [exponent of x] [exponent of y] [exponent of z]	3 2 1 0
[coefficient] [exponent of x] [exponent of y] [exponent of z]	-2 1 4 3
$\vdots$	6 0 0 0

Based on the above format, different terms of a polynomial are given in the following sequence: A term with the larger exponent of x appears before those with smaller exponents of x. If two terms have the same exponent of x, one with the larger exponent of y appears before the other with the smaller exponent of y. If two terms have the same exponents of both x and y, one with the larger exponent of z appears before the other with the smaller exponent of z.

- **Add polynomials** [Operation ID: 1]
  The operation has the format "1 [a] [b]", which indicates that you need to add the ath and the bth polynomials, which are stored in your polynomial list, and then append the result to the polynomial list, where $0 \leq a, b <$ the size of the polynomial list.

- **Subtract polynomials** [Operation ID: 2]
  The operation has the format "2 [a] [b]", which indicates that you need to subtract the bth polynomial from the ath polynomial in the list, and then append the result to the list, where $0 \leq a, b <$ the size of the polynomial list.

- **Multiply polynomials** [Operation ID: 3]
  The operation has the format "3 [a] [b]", which indicates that you need to multiply the ath and the bth polynomials in the list, and then append the result to the list, where $0 \leq a, b <$ the size of the polynomial list.

- **Delete polynomial** [Operation ID: 4]
  The operation has the format "4 [a]", which indicates that you need to remove the ath polynomial from the list, where $0 \leq a <$ the size of the polynomial list.

Based on the aforementioned operation formats, a sample input "sample.in" is given below with the comment of each line on the right side. The input file will avoid any overflow after performing each operation. The total number of the whole operations in the input file, including *append*, *add*, *substract*, *multiply*, and *delete*, could be as large as 10^{10}.

Sample Input	Comments
0 4	Append $x + y + z + 1$
1 1 0 0	
1 0 1 0	
1 0 0 1	
1 0 0 0	
0 1	Append x
1 1 0 0	
2 0 1	$(x + y + z + 1) - (x)$
3 1 1	$(x) \times (x)$
4 0	Delete $x + y + z + 1$
4 0	Delete x

3 Output Format

After performing all the operations given in the input file, you have to output the remaining polynomials, which are still stored in the polynomial list of your program, according to the following output format, where the sequence of different polynomials must follow the same sequence in the list, and the sequence of different terms in the same polynomial must follow the same sequence described in the "Append" operation.

Output Format

[number of terms in the 1st polynomial]
[coefficient] [exponent of x] [exponent of y] [exponent of z]
[coefficient] [exponent of x] [exponent of y] [exponent of z]
[coefficient] [exponent of x] [exponent of y] [exponent of z]
$\vdots$
[number of terms in the 2nd polynomial]
[coefficient] [exponent of x] [exponent of y] [exponent of z]
[coefficient] [exponent of x] [exponent of y] [exponent of z]
[coefficient] [exponent of x] [exponent of y] [exponent of z]
$\vdots$
$\vdots$

A sample output "sample.out" is given below with the comments on the right side, which results from "sample.in" in the previous section.

Sample Output	Comments
3	$y + z + 1$
1 0 1 0	
1 0 0 1	
1 0 0 0	
1	x^2
1 2 0 0	

4 Command-line Parameter

In order to correctly test your program, you are asked to add the following command-line parameters to your program.

[executable file name] [input file name] [output file name]

(e.g., StudentID.exe sample.in sample.out)

5 Submission Information

Your program must be written in the C/C++ language, and can be compiled on the Linux platform. The source files of your program must be named with "[your student ID].h" and "[your student ID].cpp". The executable file name of your program must be "[your student ID].exe". To submit your program, please archive both executable and source files of your program into a single zip file, named "[your student ID].zip", and upload to E3.

6 Due Date

The zip file must be submitted through E3 before 23:59, November 6, 2021.

7 Grading Policy

The programming assignment will be graded based on the following rules:

- Pass sample input with compilable source code (50%)
- Pass five hidden test cases (50%)

The submitted source codes, which are copied from or copied by others, will NOT be graded. There will be 25% penalty per day for late submission.