

Report for Data Structure Assignment #1

2015100059, 통계학과, 윤창원

(Codes are written in C on the Microsoft Visual Studio 2017)

The task from the assignment #1 was to make a complete program that receives two polynomials from users and adds those two polynomials. Since the implementation of the 'ADT polynomials' and 'Add function' was given in the class, the main task was to implement the receiving part and representing the result in a proper manner.

The main problem for receiving polynomials was that we don't know how many terms the user will put into the program. (To be precise, this program assumes that the polynomials' exponents are all greater than or same with 0) To handle this issue, Dynamic Memory Allocation was used.

1. Create a temporary pointer "parr" that will store inputs from the user.
2. Through the malloc function, give "parr" memories to store two (s=2) float inputs, one for a coefficient and the other for an exponent.
3. Use realloc function to give the "parr" two more spaces (which is one term in the polynomial) every time the user enters another term.
4. Attach the condition "If the exponent is 0" to end the input. (due to this condition, it is allowed to enter a term whose coefficient and exponent are both 0)

After having all the inputs stored in the "parr", they are moved to original data structure for polynomials named "terms". Since the inputs were stored in order of coefficient first and exponent later for every term, inputs with indices of even numbers are moved to "term.coef" and those with odd numbers are moved to "terms.expon". And for the exponents, their type was converted to 'int' to match with the original data structure.

The next part was to print the result of function "padd" which is the polynomial $C(x)$. Since it should be presented using the proper

expression for polynomials, the terms were divided into 3 parts.

The first term: Since it doesn't need an additional sign, print it as it is.

The last term: If the exponent is 0, only print the coefficient according to its sign. If it's not, follow the case below.

Rest of the terms: Put "+" or "-" signs according to their coefficients.

The result of the program was satisfying and it handled all the possibilities in the proper manner.