Arrays, Strings, Types and Operators

A.1 Specifications

A polynomial of a single variable x with integer coefficients is an expression of the form:

$$p(x) = c_n x^n + \ldots + c_2 x^2 + c_1 x + c_0$$

where $c_0, c_1, c_2, \ldots, c_n$ are integers.

Consider a sparse implementation of polynomials up to the n^{th} degree that stores only the terms with non-zero coefficients. For example, the polynomial

$$p(x) = -3x^7 + 824x^5 - 7x^3 + x^2 + 9$$

can be represented using the following two arrays, one of which stores the coefficients and the other, the exponents.

coeff[ ]

−3	824	−7	1	9	0	0	0	0	…

exp[ ]

7	5	3	2	0	0	0	0	0	…

Implement the following functions:

void init_polynom(int coeff[ ], int exp[ ])
Initialize all coefficients and exponents of the polynomial to zero.

void get_polynom(int coeff[ ], int exp[ ])
Get inputs (which are integers) from the user using scanf() and store them in the polynomial.
Pre-condition: the polynomial must first be initialized using init_polynom( ).
The input will be of the form [number of terms] [coefficient k] [exponent k] **…** [coefficient 1] [exponent 1]
For example, the above polynomial has 5 terms and the input will be:

5 −3 7 824 5 −7 3 1 2 9 0

Assume that all inputs are valid and no error checking is required. The terms are input in the decreasing order of their exponents.

void polynom_to_string(int coeff[ ], int exp[ ], char s[ ])
Convert the polynomial to a string **s** that can be displayed on the standard output using printf().
For example, the above polynomial has the following string representation:

−3x^7+824x^5−7x^3+x^2+9

Any term whose coefficient is zero should not appear in the string unless the polynomial has only a single constant term of zero.
A coefficient (or exponent) value of 1 should not be printed (e.g., the term x^2 above).
There is no space between the characters in the string.

void eval_polynom(int coeff[ ], int exp[ ], double x, double *result)
Evaluate the polynomial for the value of x and store the result p(x) in variable **result**.

void add_polynom(co1[ ], ex1[ ], co2[ ], ex2[ ])
Add two polynomials and store the result in the first polynomial (arrays co1[ ] and ex1[ ]).
The terms of the resulting polynomial must be stored in the decreasing order of their exponents.

Hint: Save the result in a temporary polynomial and then copy it to the first polynomial.

Note: If a polynomial is initialized before use and we do not store terms with zero coefficients, then the end of a polynomial is marked by an array entry with a coefficient 0.

A.2 Implementation

- Use file **poly.c** as the template and add your code to the specified areas. Do not change anything else in the given template, e.g., function or variable names. You may declare your own variables and functions inside file **poly.c**.
- Do not use any C library function except scanf() and printf().
- Include sufficient comments in your code to facilitate code inspection.
- To compile both **poly.c** and the main program **polyMain.c**, use the following command:
```
   > cc poly.c polyMain.c
```
- Given the test program **polyMain.c** and the input file **poly_in.txt**, the output is as in file **poly_out.txt**.
- Submit only file **poly.c**.
- Do not submit file **polyMain.c** (the grader already has this file!).