**Explanation of the algorithm for addPoly:**

To get the sum of two different Polynomials, I considered 3 situation: two Polys having the exact same size, the first Poly being larger than the second Poly, and the second Poly being larger than the first Polly.

The result Poly which is the sum of two Polys should have the size of the larger one.

If the two Polys are the same size, simply add each coefficient of the same degree of two Polys and assign it to the coefficient of the exact degree of result Poly.

If one of them is larger, first, add two Polys’ coeffs and assign it to the result coeff for the size of the smaller Poly, then, assign the rest of the result coeff array with the larger Poly’s coeffs.

Finally, return the result Poly which is the sum of two Polys.

**Explanation of the algorithm for multPoly:**

The result Poly which is the product of two Polys should have the size of the sum of two Polys.

First, all the coefficients in the coefficient array need to be initialized to 0.

Multiply two coefficients of the two polys and add it to the coefficient of the result Poly.

Finally, return the result Poly which is the product of two Polys.

**Interesting parts of my program:**

To print the polynomial (outPutPoly), many rules need to be considered and implemented using if, else if, else.

First, the constant part of polynomial is the only part that adding plus sing(+) does not need to be considered as even if it’s positive, no sign needs to be added.

Second, after the constant part, even if the coefficient is positive of the particular degree, if the coefficient before that degree is 0 (which would’ve been skipped – not printed), plus sign(+) does not need to be added.