CS102 – Algorithms and Programming II
Programming Assignment 2
Spring 2024

ATTENTION:

- Compress all Java program source files (.java) files into a single zip file.
- The name of the zip file should follow the below convention:
  CS102_SecX_Asgn2_YourSurname_YourName.zip
- Replace the variables "SecX", "YourSurname" and "YourName" with your actual section, surname, and name.
- You may ask questions on Moodle and during your section's lab.
- Upload the above zip file to Moodle by the deadline (if not, significant points will be taken off). You will have a chance to update and improve your solution by consulting with the TAs and tutors during your section's lab.

GRADING WARNING:

- Please read the grading criteria provided on Moodle. The work must be done individually. Code sharing is strictly forbidden. We are using sophisticated tools to check the code similarities. The Honor Code specifies what you can and cannot do. Breaking the rules will result in disciplinary action.

Adding More Functionality to Polynomials

You will use the Polynomial class that you implemented in the first Programming Assignment for this second assignment:

1. Implement the following operations:

- **add(Polynomial p2):** Sums current polynomial (polynomial for which the method is called) and polynomial p2, and returns the result as a new polynomial.
- **sub(Polynomial p2):** Subtracts polynomial p2 from polynomial for which the method is called.
- **mul(Polynomial p2):** Multiplies current polynomial (polynomial for which the method is called) and polynomial p2 and returns the result as a new polynomial.

For polynomials P(x), Q(x) result of addition, subtraction and multiplication operations are:

$$P(x) = 3 + 4x + 5x^2 + 2x^3$$
$$Q(x) = 2 + 4x + 1x^2$$

$$P(x) + Q(x) = 5 + 8x + 6x^2 + 2x^3$$
$$P(x) - Q(x) = 1 + 4x^2 + 2x^3$$
$$P(x) * Q(x) = 6 + 20x + 29x^2 + 28x^3 + 13x^4 + 2x^5$$

2. Implement the following compose, div, and findEqual methods. You can use the add, sub, and mul methods you already implemented to simplify these methods.

- **compose(Polynomial p2):** Returns the composition of the current polynomial with p2.

For polynomials P(x), Q(x) result of compositon (P(Q(x))) is:

$P(x) = 3 + 4x + 1x^2$
$Q(x) = 2 + 1x$

$P(Q(x)) = 3 + 4\ (2 + 1x) + 1\ (2 + 1x)^2$
$P(Q(x)) = 15 + 8x + 1x^2$

- **div(Polynomial p2):** Divides current polynomial with p2 and returns the quotient.

$P(x) = 3 + 4x + 1x^2 + 3x^3 + 2x^5$
$Q(x) = 2 + 1x$

For P(x), Q(x), the result of a division operation, P(x) / Q(x), is found as follows:

1. Find leading term (the non-zero term with highest degree) of the P(x) and Q(x).

   $lead(P(x)) = 2x^5$
   $lead(Q(x)) = x$

2. Find polynomial T(x) such that:

   $T(x) = lead(P(x)) / lead(Q(x)) = 2x^4$

3. Subtract T(x) * Q(x) from P(x) and assign the result to P(x).

   $P(x) - Q(x) * T(x) = 3 + 4x + 1x^2 + 3x^3 + -x^4$

4. Add T(x) to the result and repeat this process until the degree of P(x) is higher than the degree of Q(x).

Dividend (P(x))	T(x) = lead(P(x)) / lead(Q(x))
$3 + 4x + 1x^2 + 3x^3 + 2x^5$	x^4
$3 + 4x + 1x^2 + 3x^3 - 1x^4$	$-x^3$
$3 + 4x + 1x^2 + 5x^3$	$5x^2$
$3 + 4x - 9x^2$	$-9x$
$3 + 22x$	22
-41	

   Result of P(x) / Q(x) is $22 – 9x + 5x^2 – x^3 + x^4$. Note that the remainder is ignored.

- **findEqual(Polynomial p2):** Finds the common x values in the range [1, 200] that make this polynomial equal to polynomial p2. This method should return an int array

of the values that satisfy the solution. Do not forget that in your PolynomialTester class, you need to print the values of such x if the result is a non-empty array.

For sample polynomials P(x), Q(x) results of P(x).findEqual(Q(x)) are as follows:

```
P(x) = 10
Q(x) = 2 + 1x
P(x).findEqual(Q(x)) = {8}

P(x) = 1x
Q(x) = 1x
P(x).findEqual(Q(x)) = {1, 2, 3, ..., 199, 200} // all possible x

P(x) = 1x + 1x^4
Q(x) = 16 + 1x
P(x).findEqual(Q(x)) = {2}
```

3. Modify the PolynomialTester class from the first assignment to test your Polynomial class for the above operations.

Please reuse the available methods as much as possible instead of repeating the same code in different methods. For example, instead of calculating the degree of the polynomial for addition, you must use the method defined for this purpose. You can implement subtraction by calling polynomial addition and multiplication methods instead of repeating the code of the add method. For the compose method, you must call the multiplication method repeatedly.

Preliminary Submission: You will submit an early version of your solution before the final submission. This version should at least include the following:

- The methods add, sum, mul, and compose should be completed.

You will have time to complete your solution after you submit your preliminary solution. You can consult the TAs and tutors during the lab. Do not forget to make your final submission at the end.

Even if you finish the assignment in the preliminary submission, you should submit for the final submission on Moodle.

Not completing the preliminary submission on time results in a 50% reduction of this assignment's final grade.