Computer Engineering Department
Muğla Sıtkı Koçman University

**CENG 2002 Programming Language Concepts, Spring 2020**
**MAKE UP EXAM**

**DUE DATE: 21.06.2020, 23:59**
**This is an individual work. No team work is allowed.**
**Similarity check will be applied to submitted codes.**

# UNIVARIATE POLYNOMIAL ALGEBRA and DERIVATIVE TOOL

In your third homework, you are going to write a polynomial algebra tool. Please follow the steps given below in order to prevent any point reduction from your grade.

Let's first remember the basics:

**Polynomial** (poly-nomial: many-terms) is an expression which has constants, variables and positive exponents, e.g.

$$3x^5 + x^4 + 2x^3 - 2x + 5$$

The above expression has five terms and the degree of 5 (the largest exponent).

We call each term of polynomials as **monomials**. Therefore, monomial is a polynomial with just one term, e.g. $3x^5, x^4, 2x$ and $5$ are all monomials of the above polynomial.

Polynomial expressions can be combined using mathematical operations, e.g. addition, multiplication and we can get their derivatives.

Our tool will be able to add and multiply polynomials whereas it can take their first order derivatives. Your main function will take two polynomial terms in string form. For instance, let's assume that our polynomials are the one above together with $2x^5 + 3x^3 + 7x + 2$. Then you will create two polynomials as shown in the following:

```
char polynomial1[] = "3x5+1x4+2x3-2x1+5x0";

char polynomial2[] = "2x5+3x3+7x1+2x0";
```

First of all, you will split each polynomial into its monomials and print them (Print 1). You can tokenize your monomials using strtok function and the delimeter list of "+ –". At this point, you should have the information which delimeter is used to tokenize, + or -, since it is going to be the sign of your monomial. I can give you a hint to solve this step.

Assume that the identifier of your polynomial is **p** and your tokenize function will never touch it because it is constant. Now duplicate this string **p** into **dup1** (you can use *strdup function*). You will tokenize dup1 using strtok function and each token will be kept in a string called *token*. At each step, the delimeter that you used can be found using *p[token – dup1 -1]*.

Now you need to tokenize each monomial into its coefficient and its exponent. E.g. 3x5 should be splitted as 3 and 5. Do not forget to convert them to integer from string. You will store them either in a linked list or in a stack, you are free to choose your data structure and also the way you store your data. However, I may share my solution idea with you. The node of your data structure can keep a single integer for the coefficient and a pointer to the next node in the structure. You can define a max size for the degree of your polynomial, initialize it with all empty content where each node will refer to an exponent starting from 0 to max degree. Then you can easily insert the coefficients to corresponding nodes. Now your data structures are ready.

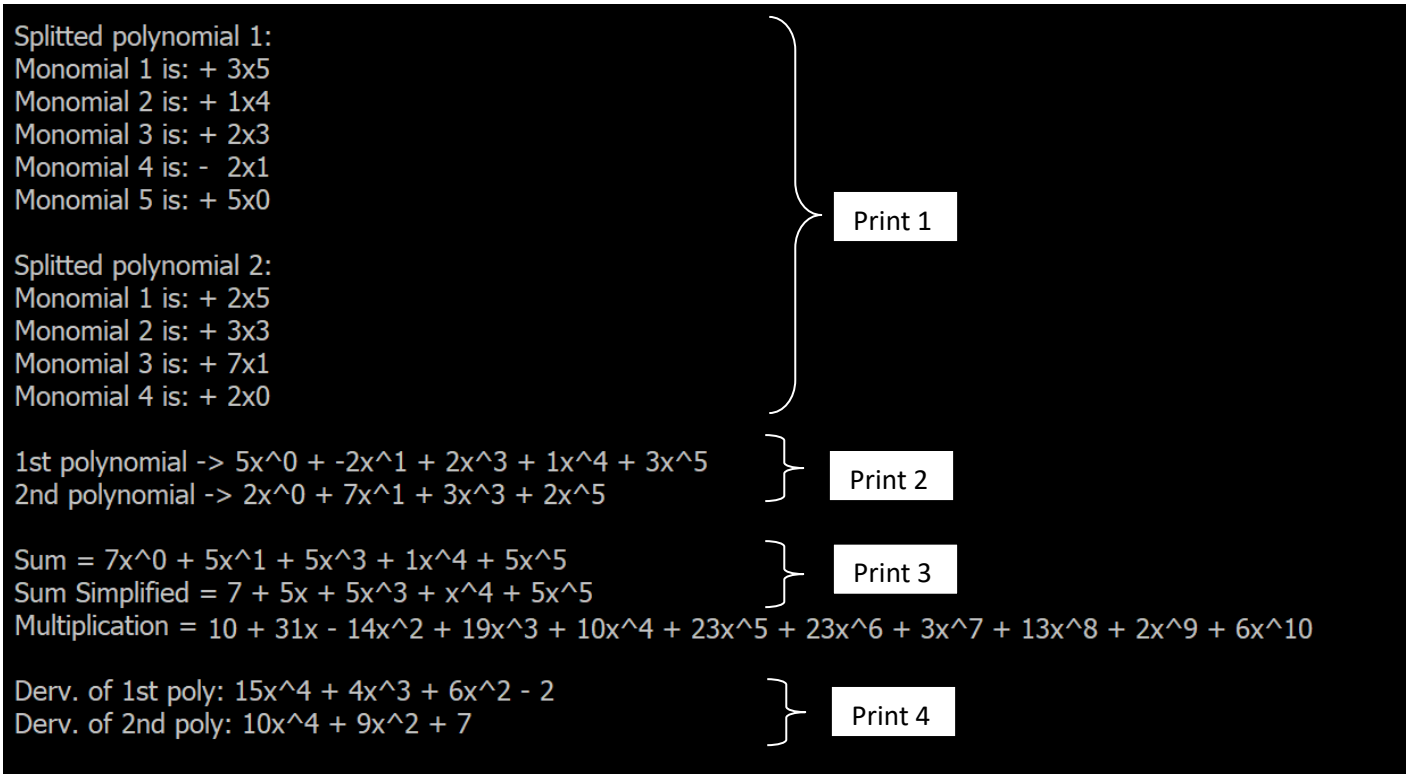
In the second step, you will print the polynomials as shown in Print 2 part. Write a print function for this aim which can take a pointer to the head of your structure.

In the third step, you will add your polynomials up, print it in a normal form and also in a simplified form, you will multiply them and print the result (no need to show in a simplified form this time). For this aim, you need to write three new functions:

- add
- multiply
- printSimplify

add and multiply functions can get two node pointers to the head of your structures and operate on them.

Finally, you will compute the derivative of each polynomial and print them. Write a new function, called derivative, for this purpose.




In any case, if you become unsure of your summation and multiplication, you can use a tool to validate your result, e.g. http://www.wolframalpha.com