

# Assignment1 : Random Algebraic Expression Generator

You are asked to write a PHP program to generate randomly an algebraic expression. If you forgot what an algebraic expression is, this is an example of an algebraic expression:  $6x - 5 + 7x - 3x^2$ . You can read more about the [algebraic expressions at Wikipedia](#).

The main idea is to generate randomly an algebraic expression with multiple terms, then a student will simplify that expression. For example, a student can simplify  $6x - 5 + 7x - 3x^2$  to  $-3x^2 + 13x - 5$ .

In fact, the goal is to write a web app that students can use to practice simplification of algebraic expressions.

In assignment1, your task is only to generate the expression. You do not need to work on the checking whether the answer of the user is correct or not.

## Requirement:

1. Create a php file and call it **lib.php**. In this file, define two functions:
  - a. **display\_html\_head**: this function has 1 input parameter which is the title of the HTML file. The function does not return any value. In this function, use HEREDoc to print doctype, <html>, <head>, <title>, and <body> of a HTML file. Note that when you call this function, you must pass the title of the document that you want to display inside <title> and </title> tags
  - b. **display\_html\_foot**: this function does not have any input parameter and it does not return any value. Using a HEREDoc, print end tags for </body> and </html>
2. Create another php file, as your main php code, and use **require** to add lib.php to this main code.
3. Define a global variable called **degree**. This variable stores the degree of the algebraic expression that we are going to generate (the greatest exponent in an algebraic expression is the degree of that expression. For example, degree of  $x+3$  is 1 and the degree of  $x^2 + 3x - 1$  is 2). We want to write a program that by changing the value of this variable, the program automatically generates an expression of that degree. Therefore, the final program will be fully parametric based on the degree parameter.
4. Define another global variable called **num\_of\_terms**. This variable stores the number of the terms that we want to generate in the expression. Num\_of\_terms depends on the degree of the expression. We use the following formula:

$$\text{num\_of\_terms} = 2 * \text{degree} + 3$$

It means that if the degree is 1, we will generate 5 terms (e.g.  $9x-1+x-5x+1$ ). If the degree is 2, we generate 7 terms (e.g.  $x+14x^2-6+x^2-2x-8x^2-4$ )

5. Define a function called **generate\_expression**. This function has only 1 input parameter, degree, and it returns a two-dimensional array called terms: in the first row of this 2D array we store coefficients and in the second row we store exponents.

**For example:** for the following randomly generated expression, the 2d-array below is filled in this function:

$$x + 14x^2 - 6 + x^2 - 2x - 8x^2 - 4$$

Coefficient	1	14	-6	1	-2	-8	-4
Exponent	1	2	0	2	1	2	0

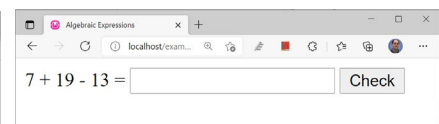
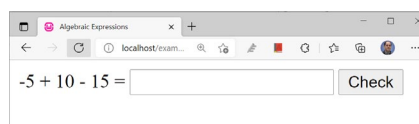
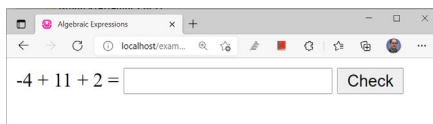
In this function use **rand()** pre-defined function of PHP to generate a random number for coefficients and exponents of the terms. Coefficient can be between -20 and 20 **excluding 0** (a coefficient cannot be zero). Exponents can be from 0 to degree value. For example, if the degree is 2, exponents can be from 0 to 2 (i.e., 0, 1, and 2).

# Assignment1 : Random Algebraic Expression Generator

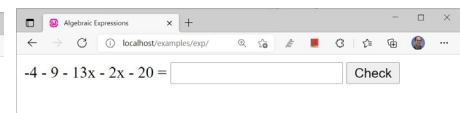
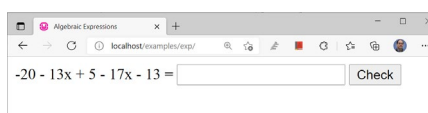
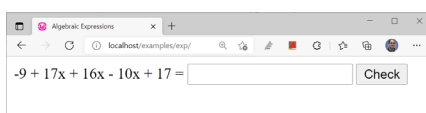
- Define a function called **meets\_requirements**. This function checks if the generated expression meets some requirements. This function has 2 input parameters, 2D-array terms, and degree, and it returns a Boolean value. To make the generated expression interesting for practice of the student, we require that there must be at least 2 terms from each degree in our expression. For example, the expression  $2x + 3 - 1 + 5 - 2$  does not meet the requirement because there is 1 term of degree 1 ( $2x$ ) and there are 4 terms of degree 0 (3, -1, 5, and -2). But  $2x + 3 - 1 + 5x - 2$  meets the requirement because there are at least two terms of degree 1, ( $2x$  and  $5x$ ) and three terms of degree 0 (3, -1, and -2). Note that your function must work for all degrees. It means if the current value of the global variable **degree** is 3, the generated expression meets requirements if there are at least 2 terms of degree 3, at least 2 terms of degree 2, at least 2 terms of degree 1 and at least 2 terms of degree 0. For example,  $2x^3 - 5x + 3 - 7x^2 + x - 4x^3 - 6 + 3x^2 + 5$  meets requirements, and your function must return true for this expression.
- Define a function called **print\_expression**. This function receives the 2D array of terms as input and prints the formatted expression on the HTML file. This function does not return any value. Some point to implement in your **print\_expression** function:
  - If the first term is positive, we do not put a plus sign before it. For example, we do not print **+3 - 5x + 7** instead we print **3 - 5x + 7**, however if the first term is negative, we display minus sign: **-6x + 6 - 2x**
  - When a coefficient is one, it is omitted (e.g.  $1x^2$  is written  $x^2$ ).
  - When the exponent (power) is one, it is omitted (e.g.,  $x^1$  is written  $x$ )
  - When the exponent is zero, the result is always 1 (e.g.,  $3x^0$  is written 3, since  $x^0$  is always 1).
  - You can use `<sup>` HTML tag for displaying an exponent.
  - You can use `abs` predefined function for finding absolute value of a coefficient.
- Define a function called **display\_expression**. This function has one parameter, degree, and it does not return any value. The task of this function is to generate an expression and then verify if it meets the requirements. If it meets, display it, else repeat until an expression that meets the requirement is generated. In this function, you call **generate\_expression**, **meets\_requirements**, and **print\_expression** inside a loop.
- On top of your code, after the global variables, call **display\_html\_head**, **display\_expression**, and **display\_html\_foot** functions. The details of how to call these functions with passing values to them is left to you.
- IMPORANT**: you must write your program such that by changing ONLY THE VALUE of degree global variable, your program generates a random expression of that degree. (You must not change any other part of your code.)

## Screenshots

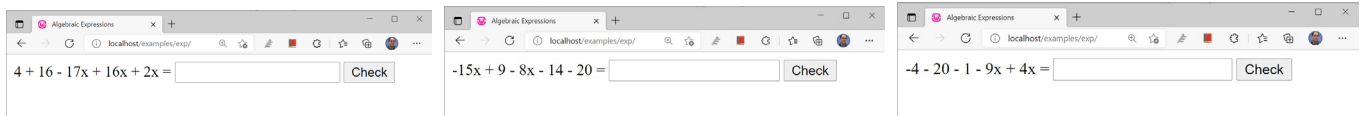
Some screenshots for degree=0:



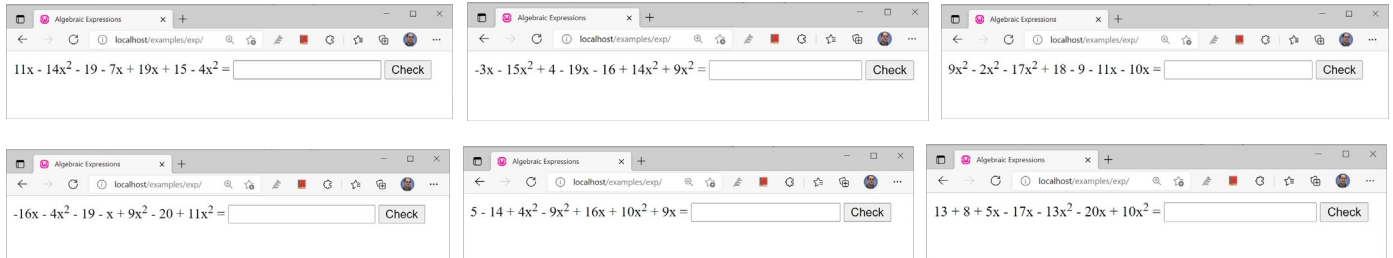
Some screenshots for degree=1:



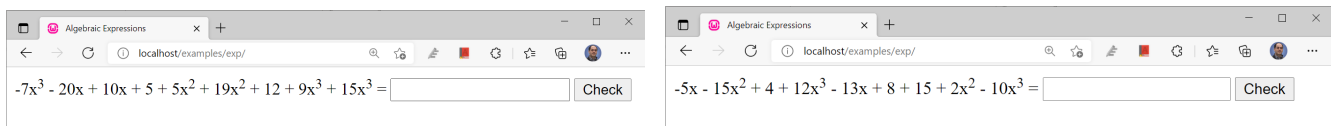
# Assignment1 : Random Algebraic Expression Generator



Some screenshots for degree=2:



Some screenshots for degree=3:



## Submission

Please submit the following items to the corresponding assignment folder on Blackboard prior to the deadline:

- A zip/archive folder containing all HTML, PHP source code files. Rename the zip file name as a#\_xy.zip where # is assignment number, x is your first name and y is your last name. For example, when I submit my work for assignment 1, I will rename the zip file as a1\_SaeedMirjalili.zip
- If you do not know [How to zip file\(s\)?](#)
- [Or search Google](#)
- [I found this video also useful for zipping and uploading your project](#)

## Marking Guideline

Task	Points	Granted
Use of lib.php and <b>require</b> in main.php	5	
display_html_head function has 1 parameter and uses HEREdoc	7	
display_html_foot function; no input, no return	5	
The program is parametric based on the variable degree	10	
<b>generate_expression</b> function according to the step 5	19	
<b>meets_requirements</b> function according to the step 6	19	
<b>print_expression</b> function according to the step 7	23	
<b>display_expression</b> function according to the step 8	12	
<b>Total</b>	<b>100</b>	<b>0</b>