1. EXERCISE (4.1.pl)
Define a variable called $x and put a number into it. Now write a subroutine called "increment" that adds 1 to this variable. Use this subroutine to increment the contents of $x three times. Print $x to see if it worked.

2. EXERCISE (4.2.pl)
Write a subroutine called "absolute" that calculates the absolute value of number and returns it, i.e. if it finds a negative number it should return its positive counterpart. If the number is positive it should simply return the number. Print the returned value to check if it worked.

3. EXERCISE (4.3.pl)
Write a subroutine that calculates the sum of an array (that you can define as you like) and returns it. Print the result to the screen.

4. EXERCISE (4.4.pl)
Modify 4.3.pl to immediately return the text "negative number found" if it finds a negative number in the array. If not it should still return the sum of the array.

5. EXERCISE (4.5.pl)
Modify 4.2.pl so you can use the subroutine on any variable by passing it as a parameter when calling the subroutine. Apply the subroutine to two different variables (e.g. $x and $y). Print the results of the subroutine to see if it worked.

6. EXERCISE (4.6.pl)
Define 4 variables $x, $y, $a and $b and fill them with numbers. Now write a subroutine that calculates the difference of two variables that get passed on as parameters. Inside the subroutine use private variables called $a and $b to do the calculation (i.e. copy over the contents of @_). Use the subroutine to calculate the difference between $x and $y first and then the difference between $a and $b that you defined in the very beginning. Check what happens if don't make sure that the $a and $b inside the subroutine are private.