1. **Product of numbers**

Write a JAVAscript function Prod() that receives an arbitrary amount of numbers and returns their product. For example, if you type show(Prod(2,3,4,5)), you should receive 120 (which is 2*3*4*5). If no argument is given (Prod()), the output should be 1.

2. **Min and Max**

Consider the algorithm seen in class to find the location of the Minimum element:

```
Procedure FindMin(x_1x_2...x_n)
mini:=1; min:=x_1

for i:=2 to n do

    if x_i<min then min:=x_i; mini:=i

output mini
```

a) Argue that the comparison of elements in the if statement occurs n-1 times in total.
b) Rewrite this algorithm as **FindMax** so to find the location of the Maximum instead of the Min.

c) Write a new algorithm that finds BOTH the Min and the Max using at most $\frac{3n}{2}$ comparisons.

3. **Mystery algorithm**

Consider the following algorithm:

```
Input $T, x_1, x_2, \ldots, x_n$
$T_i := 0$
for $i := 1$ to $n$ do
    if $T = x_i$ then $T_i := i$
output $T_i$
```

a) Run this algorithm on inputs $T=3, x_1=2, x_2=5, x_3=3$, and provide a complete trace of execution.

b) Describe in words, what this algorithm is doing. (clarify what happens when some $x_i$‘s are identical)

c) What is to be understood when the output is 0?
4. **JavaScript addition**

Write a JavaScript function `add` that inputs two strings `x, y` and output another string `z` such that if you consider the numerical value of the characters of `x` and similarly for `y` and `z`, we have the relation `z = x + y`.

Example:

```javascript
add("500000000303030303033030303000000000333333333333", "500000000303030303033030303000000000033333333333333")
```

should output the string

```
z = "1000000000606060606060606060600000000666666666666"
```

which is the numerical sum of the values represented by `x` and `y`. Your function should work for arbitrarily long numbers. We don’t care what happens if the inputs are not of the right format. A few examples will be made available to you. Your function should be executed on these inputs and return the correct output.

5. **Multiplication**

Write an algorithm to multiply two numbers (base 10) of `n` digits each and output a number of `2n` digits:

```
x_n x_{n-1} ... x_1 \times y_n y_{n-1} ... y_1 = z_{2n} z_{2n-1} z_{2n-2} ... z_1.
```