1. Write a MATLAB function for calculating the Euclidean distance between two two-dimensional arrays A and B given as input arguments. A is of size $N \times n$, and B is of size $M \times n$. The function should return a matrix D of size $N \times M$ where element $d(i, j)$ is the Euclidean distance between row i of A and row j of B.

2. Write an inline function that will calculate $6x - 4y + xy + cos^2(x - k)$. Assume that x and y may scalars, vectors or matrices and that all operations should be carried out element-wise.

3. Write a MATLAB function for checking if a given point (x, y) is within the square with a bottom left corner at (p, q) and side s. The input arguments are x, y, p, q and s, and the output is either true (the point is in the square) or false (the point is not in the square).

4. Given a function name, return the names and values of the input arguments.

   e.g. function_name(arg1, arg2) is a function definition, then return input names as arg1 and arg2 alongwith their values.

5. This Challenge is to utilize Global variables.

   Global variables are risky as the subroutine may inadvertently and unwantingly update them. Globals also tend to slow execution time.

   The Challenge is to implement and become aware of capability and risk.

   The global variables in this Challenge are gOffset and global_x.

   Given two input variables [a,b] output [gOffset+a global_x*b]

   Input: [a,b] (Two real values)

   Output: [gOffset+a global_x*b]

6. This Challenge is to implement the Matlab Persistent variable capability.

Given a sequence of Calls to a function return the current input summed with the prior function call input.

On the first call return the current input.

Input: a (value)

Output: b where b=a+"previous a"

Example:

```
    Input  Sequence: 1 2 -3 3 5
    Output Sequence: 1 3 -1 0 8
```