Assignment D Similarity matrix

A similarity matrix contains the pairwise similarities between each element in two vectors. We have the vectors $x = [x_1, x_2, \ldots, x_N]$ and $y = [y_1, y_2, \ldots, y_M]$ and we define the following similarity measure:

$$\mathrm{s}(x_i, y_j) = \exp\left(-\delta(x_i - y_j)^2\right),$$

where δ is a parameter known as the length scale. The similarity matrix S is then defined as:

$$S = \begin{bmatrix} \mathrm{s}(x_1, y_1) & \mathrm{s}(x_1, y_2) & \cdots & \mathrm{s}(x_1, y_M) \\ \mathrm{s}(x_2, y_1) & \mathrm{s}(x_2, y_2) & \cdots & \mathrm{s}(x_2, y_M) \\ \vdots & \vdots & & \vdots \\ \mathrm{s}(x_N, y_1) & \mathrm{s}(x_N, y_2) & \cdots & \mathrm{s}(x_N, y_M) \end{bmatrix}.$$

The matrix S is thus an $N \times M$ matrix, where element (i, j) is the similarity between x_i and y_j.

Problem definition

Write a function named `similarityMatrix` that takes as input the two vectors x and y as well as the length scale δ and returns the similarity matrix S.

Solution template

```
function S = similarityMatrix(x, y, delta)
% Insert your code
```

Input	
`x`, `y`	Input data (vectors).
`delta`	Length scale (decimal number).

Output	
`S`	Similarity matrix.

Example

Consider the following input vectors:

$$x = [1.1,\ 1.2], \quad y = [1.3,\ 1.4,\ 1.5],$$

and say we have $\delta = 2$. The similarity matrix can then be computed as:

$$S = \begin{bmatrix} 0.9231 & 0.8353 & 0.7261 \\ 0.9802 & 0.9231 & 0.8353 \end{bmatrix}$$

where, for example, element (1,2) is computed as:

$$\mathrm{s}(x_1, y_2) = \exp\left(-2 \cdot (1.1 - 1.4)^2\right) = 0.8353$$

D

```
function S = similarityMatrix(x, y, delta)
    N = numel(x);
    for i = 1:N
        S(i,:) = exp(-delta*(x(i) - y).^2);
    end
end
```

Assignment E Coin return

You are designing a machine for giving change (coins) to customers in a supermarket. Given that a certain amount of money is to be paid out, you need to decide which coins to give. You know the denominations (values) of the different kinds of coins available in the machine, and can assume that there are sufficiently many coins available of each kind. You must use the following algorithm to select the coins:

1. Let x denote the amount to be paid.
2. If x is less than half the value of the smallest coin: Stop.
3. Pay out the largest available coin with a value less than or equal to x.
4. Subtract the value of the paid out coin from x.
5. Repeat from 2 .

Problem definition

Create a function named `coinReturn` that takes as input vector containing the values of coins available in the machine (ordered starting with the smallest value) as well as the amount to be paid out. The function must return a vector with the values of the coins that are paid out (in the order defined by the algorithm).

Solution template

```
function coinsToReturn = coinReturn(coinsInMachine, amount)
% Insert your code
```

Input	
`coinsInMachine`	The values of the kinds of coins available in the machine (vector).
`amount`	Amount to be paid out (decimal number).

Output	
`coinsToReturn`	The values of the coins that are paid out (vector).

Example

Consider paying out the amount 34.60, when the following kinds of coins are available:

The input `coinsInMachine` will be the vector [0.50, 1, 2, 5, 10, 20]. The algorithm should proceed as follows:

	Iteration 1	Iteration 2	Iteration3	Iteration 4	Iteration 5	Iteration 6
Remaining amount (x)	34.60	14.60	4.60	2.60	0.60	0.10
Coin paid out	20	10	2	2	0.5	Stop

Thus, the coins paid out will be $[20, 10, 2, 2, 0.5]$ which should be the output of the function.

E

```
function coinsToReturn = coinReturn(coinsInMachine, amount)
    y = (coinsInMachine(1) / 2);
    i = 1;
    while(amount >= y)
        coinsToReturn(i) = coinsInMachine(find(coinsInMachine <= amount, 1,'Last'));
        amount = amount - coinsToReturn(i);
        i = i + 1;
    end
end
```