```
matrix_function.mix_matrix: Method in matrix_function that
        mix to b array and a array to find a A matrix
matrix_function.soltion: Method in matrix_function that do
        a progressive or backward substitutionv to find an array
total_gaussian_method.totalGaussianMethod: Method in
        total_gaussian_method that found the solution of matrix
        Ax = B by total gaussian method.
matrix_function.sort: Method in matrix_function that organize
        in case of changing rows

Input: array x, array y
Output: array coefficient

begin splineLineal

sizeX <- size(x)
sizeY <- size(y)

if (sizeX != sizeY):
        break;
int m <- 2*(sizeX -1)
A = identityMatrix(m)

int count <- 0
while (count < m) do:
        A[count][count] <- 0
end while

int counter <- 0
int counterRow <- 0
array b
int count <- 0
while (count < sizeX) do:
        array vec_x <- x[count]
        A[count][counter] <- vec_x
        A[count][counter+1] <- 1
        counter <- counter + 2
        if (count == 0):
                counter <- 0
        b.add(y[count])
        counterRow <- counterRow + 1
        count <- count + 1
end while
counter <- 0
count <- 0
while (count < m) do:
        A[counterRow][counter] <- x[count]
        A[counterRow][counter+1] <- 1
        A[counterRow][counter+2] <- -(x[count])
        A[counterRow][counter] <- -1
        counter <- counter + 2
        counterRow <- counterRow + 1
        b.add(0)
end while
array arr
count <- 0
while (count < m) do:
        array arr2
        int countAux <- 0
        while (countAux < m) do:
                arr2.add(A[count][countAux])
        arr.add(arr2)
        end while
end while
a, b, matrix <- matrix_function.mix_matrix(A,b)
a,b,dic,movement <- total_gaussian_method.totalGaussianMethod(matrix)
x = matrix_function.soltion(a,b)
x = matrix_function.sort(x,movement)
array aux
count <- 0
while (count < size(x)):
        aux.add(x[count])
end while
array coefficient
array plotter
count <- 0
while (count < size(aux)-1) do:
        array aux2
        aux2.add(aux[count])
        aux2.add(aux[count+1])
        count <- count + 2
        coefficient.add(aux2)
end while
count <- 0
return coefficient

end splineLineal
```