```
Input: Augmented n x n+1 matrix Augmented_matrix
        Output: square nxn matrix A,colum vector b, solution array x with steps
begin LuSimpleMethod

        auxialiry_matrix <- Augmented_matrix
        for i from 0 to n-1 do
                pivot_number = auxialiry_matrix[0][0] --> l x l+1 matrix
                if (pivot_number = 0) then
                        for j from 0 to l-1 do
                                if (auxialiry_matrix[j][0] = 0) then
                                        switch auxialiry_amtrix[j][0]
                                        and auxialiry_matrix[0][0]
                if (pivot_number = 0) and (i = n-2) then
                        break
                fj <- auxialiry_matrix[0]
                column_vector <- columnFrompivotnumber(auxialiry_matrix)
                multiplier <- column_vector/pivot_number
        fi <- auxiliary_matrix[1:]
        fi <- fi - (multiplier*fj)
        if (i = 0) then
                Augmented_matrix[i+1:] <- fi
        else:
            Augmented_matrix <- complitFirstColumnWithZeros(fi)
        auxiliary_matrix <- cutFisrtRowAndFisrtColumn(fi)
        solution_array[i+1] <- Augmented_matrix
        solution_array_l[i+1] <- triangular_boton(Augmented_matrix)
        solution_array_u[i+1] <- triangular_top(Augmented_matrix)
    matrix_A <- deleteLastColumn(Augmented_matrix)
    vector_b <- getLastColumn(Augmented_matrix)
    matrix_A, vector_b, solution_array, solution_array_u, solution_array_l
end LuSimpleMethod
```