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

begin partialGaussianMethod

        auxialiry_matrix <- Augmented_matrix
        for i from 0 to n-1 do
                pivot_number <- auxialiry_matrix[0][0]
--> l x l+1 matrix
                pivot_column <- getFirstColumn(auxialiry_matrix)
                pivot_column <- absoluteValueInColumn(pivot_column)
                pos_max_pivot <- getIndexMaxValueFromColumn(pivot_column)
                if (pos_max_pivot != 0) then
                        switchColmn auxialiry_matrix[0][0] and
                            auxialiry_matrix[pos_max_pivot][0]
                        switchColmn Augmented_matrix[0][0] and
                            Augmented_matrix[pos_max_pivot][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
    matrix_A <- deleteLastColumn(Augmented_matrix)
    vector_b <- getLastColumn(Augmented_matrix)
    matrix_A, vector_b, solution_array

end simpleGaussianMethod
```