# Метод Гаусса оптимального исключения

Написать программу для решения СЛАУ методом Гаусса оптимального исключения.

## Код программы

myA = [  
 [5, 7, 6, 5],  
 [7, 10, 8, 7],  
 [6, 8, 10, 9],  
 [5, 7, 9, 10]  
]  
  
myB = [  
 23,  
 32,  
 33,  
 31]  
  
  
# вывод системы на экран  
def fancy\_print(A, B, selected):  
 for row in range(len(B)):  
 print("(", end='')  
 for col in range(len(A[row])):  
 print("\t{1:10.2f}{0}".format(" " if (selected is None or selected != (row, col)) else "\*", A[row][col]),  
 end='')  
 print("\t) \* (\tX{0}) = (\t{1:10.2f})".format(row + 1, B[row]))  
  
  
# перемена местами двух строк системы  
def swap\_rows(A, B, row1, row2):  
 A[row1], A[row2] = A[row2], A[row1]  
 B[row1], B[row2] = B[row2], B[row1]  
  
  
# деление строки системы на число  
def divide\_row(A, B, row, divider):  
 A[row] = [a / divider for a in A[row]]  
 B[row] /= divider  
  
  
# сложение строки системы с другой строкой, умноженной на число  
def combine\_rows(A, B, row, source\_row, weight):  
 A[row] = [(a + k \* weight) for a, k in zip(A[row], A[source\_row])]  
 B[row] += B[source\_row] \* weight  
  
  
# решение системы методом Гаусса (приведением к треугольному виду)  
def gauss(A, B):  
 column = 0  
 while column < len(B):  
  
 # ищем максимальный по модулю элемент в с+1 столбце  
 current\_row = None  
 for r in range(column, len(A)):  
 if current\_row is None or abs(A[r][column]) > abs(A[current\_row][column]):  
 current\_row = r  
 if current\_row is None:  
 print("решений нет")  
 return None  
  
 if current\_row != column:  
 # переставляем строку с найденным элементом повыше  
 swap\_rows(A, B, current\_row, column)  
  
 # нормализуем строку с найденным элементом  
 divide\_row(A, B, column, A[column][column])  
 # fancy\_print(A, B, (column, column))  
  
 # обрабатываем нижележащие строки  
 for r in range(column + 1, len(A)):  
 combine\_rows(A, B, r, column, -A[r][column])  
  
 column += 1  
  
 # матрица приведена к треугольному виду, считаем решение  
 X = [0 for b in B]  
 for i in range(len(B) - 1, -1, -1):  
 X[i] = B[i] - sum(x \* a for x, a in zip(X[(i + 1):], A[i][(i + 1):]))  
  
 print("\nПолучили ответ:")  
 print("\n".join("X{0} =\t{1:10.2f}".format(i + 1, x) for i, x in enumerate(X)))  
  
 return X  
  
  
print("Исходная система:")  
fancy\_print(myA, myB, None)  
  
gauss(myA, myB)

## Результат

