

Учреждение образования  
«Белорусский государственный университет информатики и  
радиоэлектроники»  
Кафедра информатики

# Отчёт

Лабораторная работа №1  
По учебной дисциплине *Методы оптимизации и управления*

Выполнил:

студент группы №853504

Кузьма В.В.

Проверил:

доцент кафедры информатики

Дугинов О.И.

Минск 2021

## ЗАДАНИЕ

Дана квадратная матрица и её обратная матрица.

Необходимо заменить в матрице  $i$ -ый столбец на вектор и найти обратную матрицу к новой матрице

### Примеры работы

```
A = np.array([[1, 0, 5], [2, 1, 6], [3, 4, 0]])
A_inv = np.array([[-24, 20, -5], [18, -15, 4], [5, -4, 1]])
x = np.array([[2], [2], [2]])
i = 2
print(solve(A_inv, x, i))
```

```
[[ -0.85714286  0.71428571  0.14285714]
 [ 1.28571429 -1.07142857  0.28571429]
 [-0.14285714  0.28571429 -0.14285714]]
```

```
A = np.array([[1, -1, 0], [0, 1, 0], [0, 0, 1]])
A_inv = np.array([[1, 1, 0], [0, 1, 0], [0, 0, 1]])
x = np.array([[1], [0], [1]])
i = 3
print(solve(A_inv, x, i))
```

```
[[ 1.  1. -1.]
 [ 0.  1.  0.]
 [ 0.  0.  1.]]
```

## КОД ПРОГРАММЫ

```
def solve(A_inv, x, k):
    n = len(A_inv)
    k -= 1 # для лучшей индексации
    # ШАГ 1
    l = np.dot(A_inv, x) # вычисляем вектор l
    li = l[k][0] # находим i-ую компоненту вектора l
    if (li == 0): # проверка компоненты на равенство нулю
        return "Матрица не может быть обратимой"
    # ШАГ 2
    l1 = np.copy(l) # получаем вектор l с волной
    l1[k][0] = -1 # заменяем i-ую компоненту вектора l с волной -1
    # ШАГ 3
    l2 = np.dot(-1/li, l1) # получаем вектор l с шапочкой
    # ШАГ 4
    E = np.eye(n) # создаем единичную матрицу
    Q = np.copy(E)
    Q[:,k] = l2.transpose() # создаем матрицу Q
    # ШАГ 5
    z = np.eye(n)
    for i in range(n):
        for j in range(n):
            z[i][j] = Q[i][i]*A_inv[i][j]
            if (i != k):
                z[i][j] += Q[i][k]*A_inv[k][j]
    return z
```