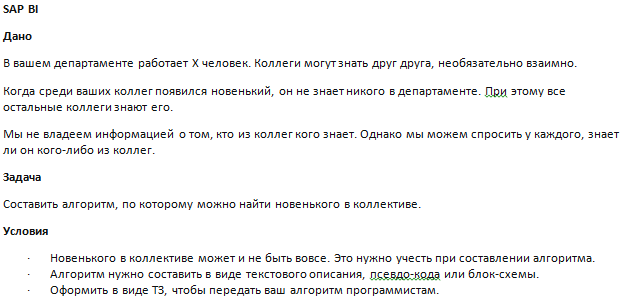
## Кейс №4

Наумов Александр

# Задание:



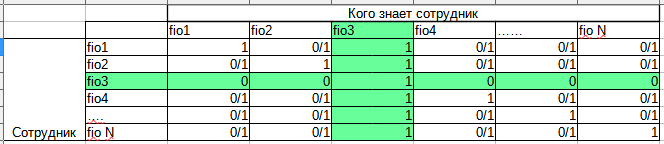
# Текстовое решение:

1. Нужно составить матрицу отношений между сотрудниками (новичком будет тот которого знают все (в столбике все 1), но он никого не знает, кроме себя (в строчке все 0, кроме себя)).

1 – знает

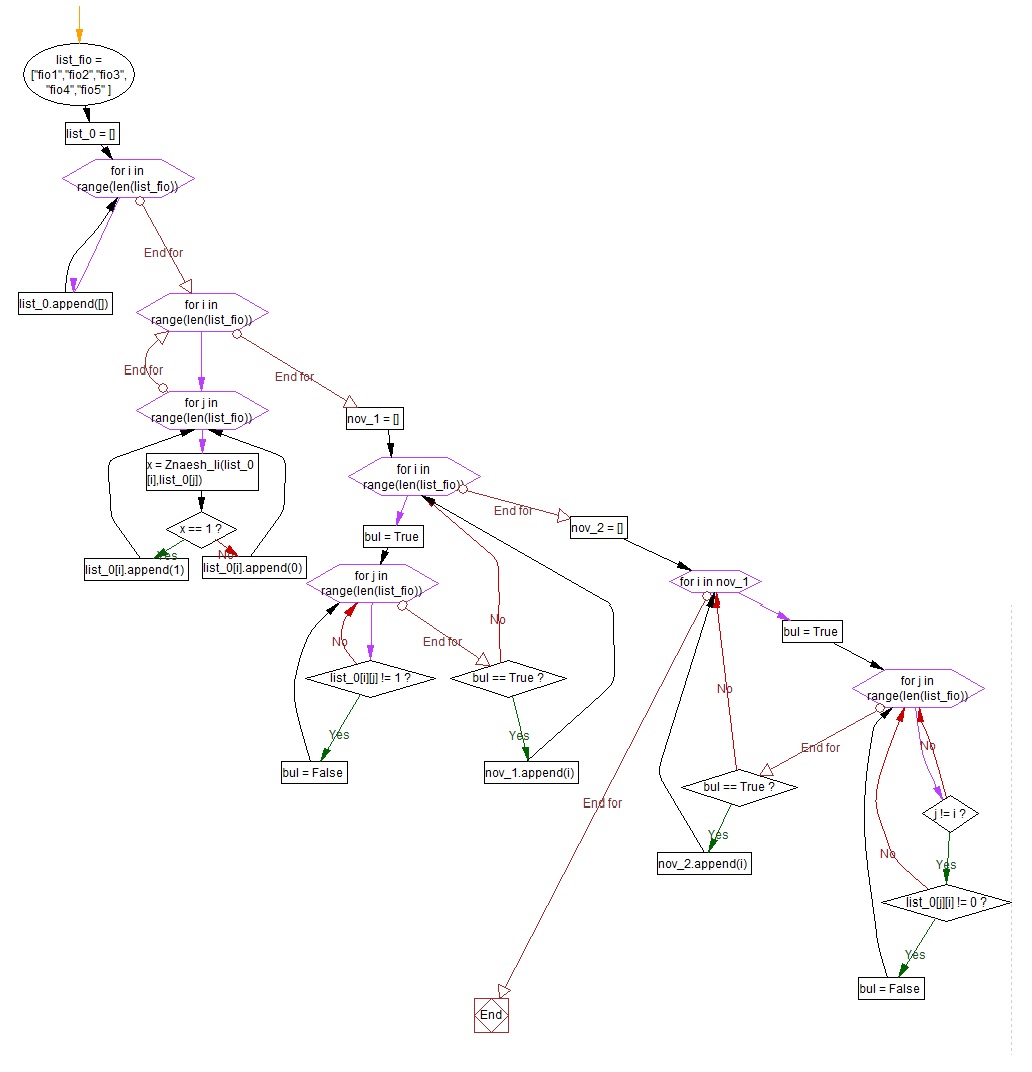
0 – незнает

0/1 – не имеет значения



1. В матрице найти сотрудников, которых знают все.
2. Среди сотрудников из пункта 2 найти сотрудника, который знает только себя, этот сотрудник и будет новичком.

# Блок-схема решения:



# Псевдокод решения:

**Обозначения:**

List\_fio - список фамилий сотрудников.

List\_0 – список с вложенными списками (матрица отношений между сотрудниками, 1 – знает, 0 – не знает)

Nov\_1 – список с индаксами людей, которых, знают все.

Nov\_2 – список индексов людей, которые состоят в списке nov\_1 и которые не знают никого.

Znaesh\_li (fio1, fio2) – функция возвращающая 1(единицу), если человек под fio1 знает человека под fio2, и возвращает 0(нуль), если не знает.

Fio1 – ФИО сотрудника типа string.

Fio2 – ФИО сотрудника типа string.

.append() – добавление элемента в список.

Len() – измерение длины списка.

**Код:**

list\_fio = ["fio1","fio2","fio3","fio4","fio5" … “fioN” ]

list\_0 = [ ]

for i in range(len(list\_fio)):

list\_0.append([ ])

for i in range(len(list\_fio)):

for j in range(len(list\_fio)):

x = Znaesh\_li (list\_0[i],list\_0[j])#знает ли итого, житый?

if x == 1:

list\_0[i].append(1)

else:

list\_0[i].append(0)

nov\_1 = [ ] #проверяем, есть ли человек которого знают все

for i in range(len(list\_fio)):

bul = True

for j in range(len(list\_fio)):

if list\_0[i][j] != 1:

bul = False

if bul == True:

nov\_1.append(i)

nov\_2 = [ ] #проверяем, человек из прошлого пункта знает кого то?

for i in nov\_1:

bul = True

for j in range(len(list\_fio)):

if j != i:

if list\_0[j][i] != 0:

bul = False

if bul == True:

nov\_2.append(i)

#теперь список nov\_2 содержит новичка если он есть, если нету то список пуст