

## Оглавление

1) Условие задачи.....	3
2) Схема программы.....	4
3) Описание программы.....	7
4) Текст программы.....	8
5) Заключение.....	9
6) Список литературы.....	10

## Оригинальный текст задачи

### Eeny Meeny

В самом темном уголке земли (или не совсем земли) < имя континента/острова удаленного, чтобы предохранить нарушение > обитает племя под названием «Eeny Meeny». Они получают это название в честь способа выбора руководителя на год. Оказывается, газетному репортеру, посещавшему племя, удалось привить несколько идей цивилизации, но очевидно он куда-то исчез прежде, чем завершил свою работу. Таким образом, племя больше не имело постоянного руководителя, срок правления (один год) подошел к концу. Они съели текущего руководителя, и быть избранными племенные участники (женщины также имели право быть избранными – один из благословений цивилизации племени приняло) становились в круг, выбиралось место для начала отсчета и главный знахарь начинал считать «Е», «е», «п», «у», «М», «е», «е», «п», «у», «М», «i», «п», «у», «М», «o!», «Е», «е», «п», «у», «М», «е», «е», «п», «у», «М», «i», «п», «у», «М», «o!», .... На каждой «o!», выбранный человек выталкивался из круга, который затем замыкался и всё продолжалось с его соседа (т.е. с того, кто был бы «Е» во всяком случае). Этот процесс продолжался до тех пор, пока не оставался один человек – новый руководитель.

Работа руководителя очень привлекательная, но Вам не хочется быть съеденным через год. Вам удалось обнаруживать, что счет в этом году начнется с Mxgobgwq (очень большой человек), так что Вы хотели бы узнать, где не стоять. Вы не знаете ни направления, ни сколько людей будет участвовать, но Вы можете оценить их число (это – несомненно, менее чем 150).

Напишите программу, которая определит «первую» (то есть ближайшую к Mxgobgwq) безопасную позицию, независимо от фактического количества людей и направления счета (по часовой стрелке или анти - по часовой стрелке).

#### **Ввод:**

Ввод состоит из серии строк, каждая строка содержит минимальное и максимальное количество людей имеющих право быть избранными. Ввод заканчивается строкой, содержащей два нуля (0 0).

#### **Вывод:**

Вывод состоит из серии строк, по одной для каждой строки ввода. Каждая строка состоит из единственного числа (номер позиции ближайшей к Mxgobgwq, которая не будет выбрана как руководитель для любого числа в данной области и для каждого направления). Если никакая позиция не является безопасной, тогда напечатайте «Недостаточно данных»

#### **Пример ввода**

80 150

40 150

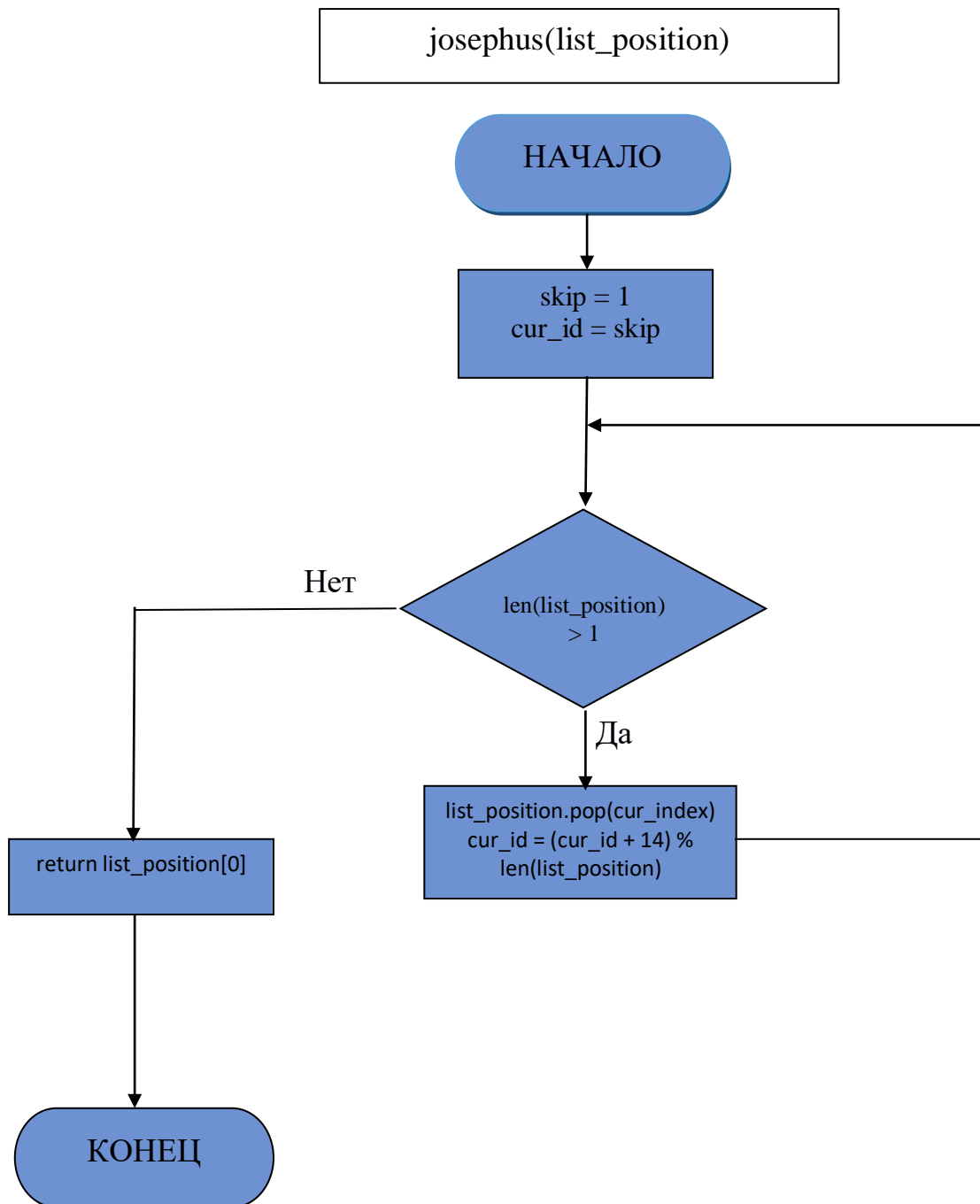
0 0

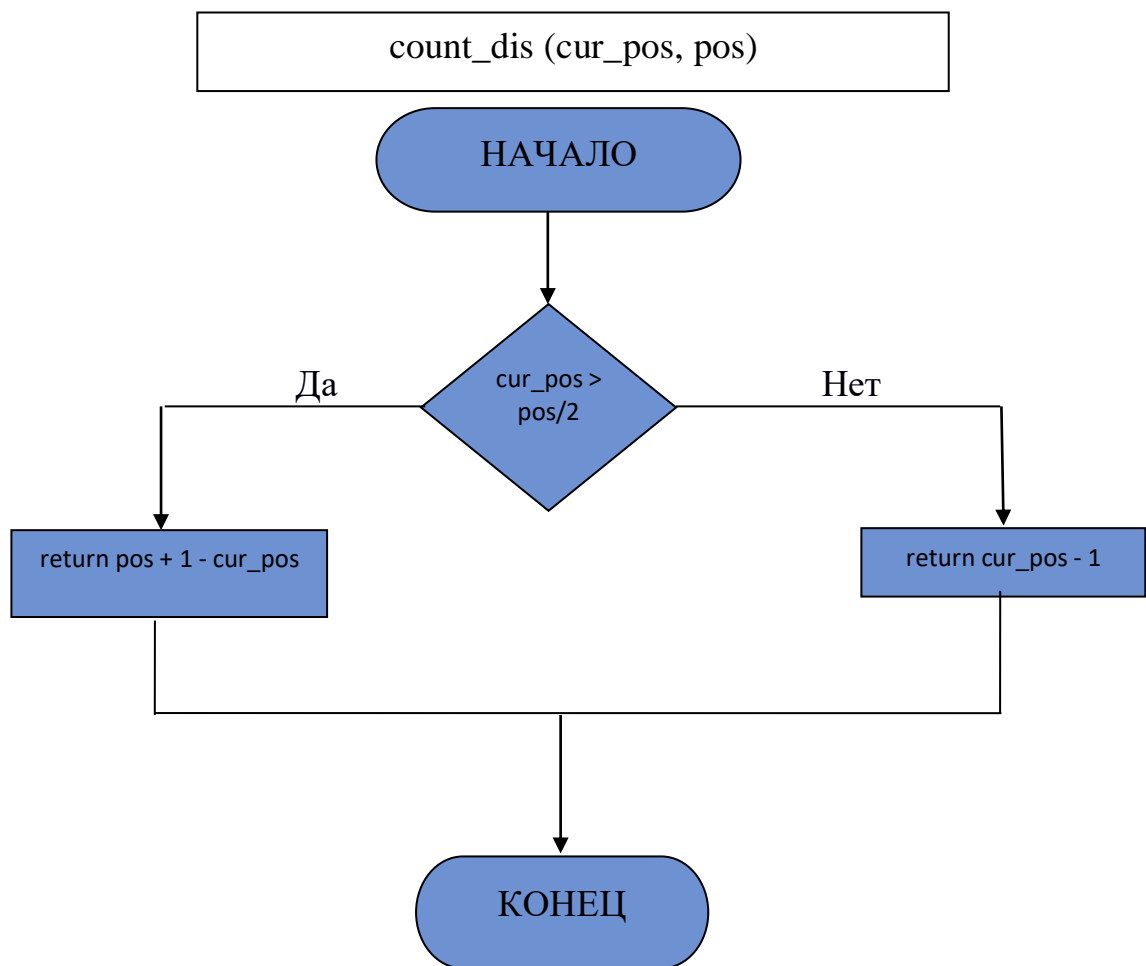
#### **Пример вывода**

1

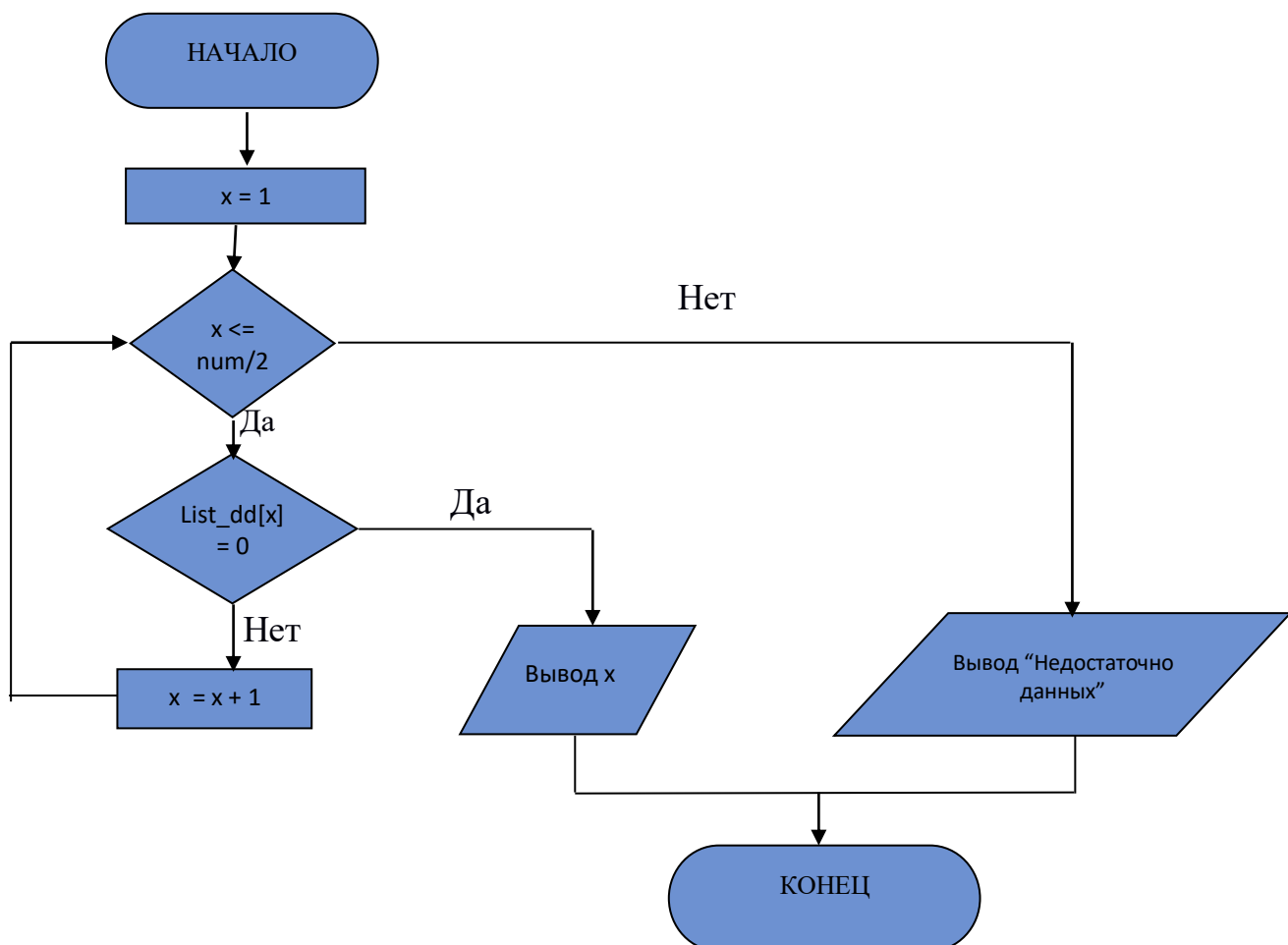
Недостаточно данных

## Схема программы

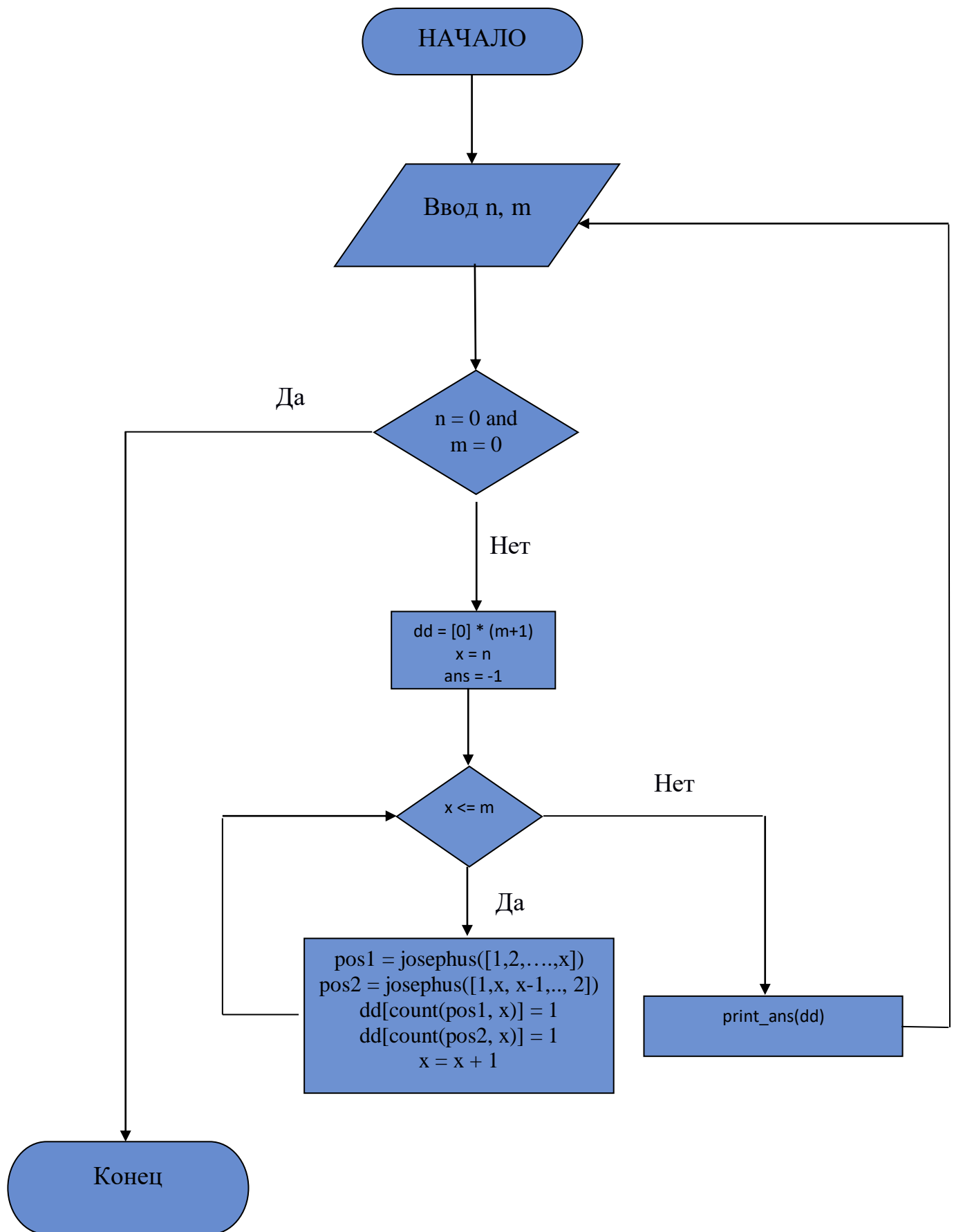




print\_ans(list\_dd, num)



Основная программа

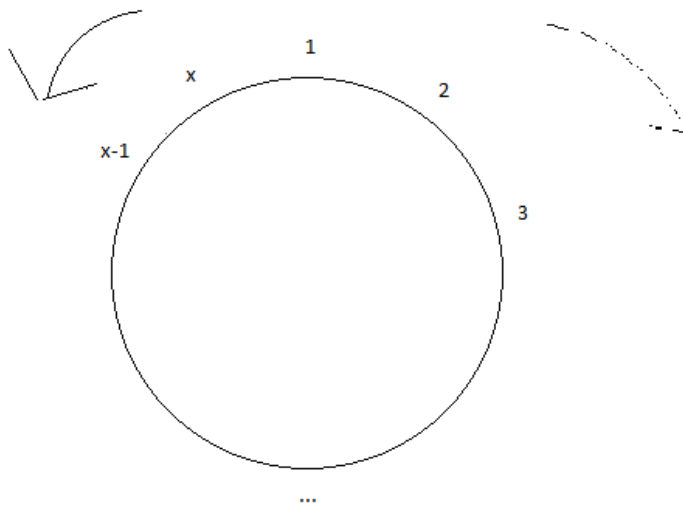


## Описание программы

В безопасном положении, независимо от фактического количества людей и направления счета (по часовой стрелке или анти - по часовой стрелке), расстояние до Мхgobgwq не превышает  $\lceil n / 2 \rceil$ , поскольку количество участников никогда не превышает  $n$ .

Сначала найти все позиции, которые будут выбраны для нового руководителя, и вычислите их расстояние до Мхgobgwq. Затем на всех расстояниях до Мхgobgwq меньше, чем  $(n-1) / 2$ , если позиция не выбрана, которое будет ближайшей безопасной позицией к Мхgobgwq.

Чтобы найти позицию, которая выбрана для нового руководителя, с  $x$  участниками, укажите Мхgobgwq в позиции 1, создать массив  $list\_position = [1, 2, 3, \dots, x]$  при счете по часовой стрелке и  $list\_position = [1, x, x-1, \dots, 2]$  при обратном отсчете по часовой стрелке. Затем удаляйте элементы, пока не останется 1 элемент, его значением будет выбранная позиция.



(По часовой стрелке будет 1, 2, 3, ..., x, 1, 2, ... ->  $list\_position = [1, 2, 3, \dots, x]$ )

По анти-часовой стрелке будет 1, x, x - 1, ..., 2, 1, x, x-1, ...

->  $list\_position = [1, x, x-1, \dots, 2]$ )

## Текст программы

```
# Найти номер позиции нового руководителя
def josephus(list_position):
    skip = 14
    cur_id = skip
    while len(list_position) > 1:
        list_position.pop(cur_id)
        cur_id = (cur_id + 14) % len(list_position)
    return list_position[0]

# Вычислить расстояние от Mxgobgwq до нового руководителя
def count_dis(cur_pos, pos):
    if cur_pos > pos // 2:
        return pos + 1 - cur_pos
    return cur_pos - 1

# Определить и напечатать номер безопасной позиции, которая ближайшей к
Mxgobgwq
def print_ans(list_dd, n):
    for x in range(1, n//2+1):
        if list_dd[x] == 0:
            print(x)
            return
    print("Недостаточно данных")

while 1:
    n, m = map(int, input().split())
    if not n and not m:
        break
    dd = [0] * (m+3)
    for x in range(n, m+1):
        pos1 = josephus([i for i in range(1, x+1)])
        pos2 = josephus([1] + [i for i in range(x, 1, -1)])
        dd[count_dis(pos1, x)] = 1
        dd[count_dis(pos2, x)] = 1
    print_ans(dd, n)
```



## **Заключение**

В ходе работы удалось разобраться со структурой графов, рассмотреть алгоритмы работы с ними, разработать программу, реализующую алгоритм, решающий поставленную задачу, а также написать отчёт о проделанной работе.

Я уверен, эти знания понадобятся мне в будущем обучении.

## Список литературы

- Лекционные конспекты Борисов С.В.
- Лутц Марк Python. Карманный справочник, 5-е изд. :Пер. с англ. – М. : ООО «И.Д. Вильямс», 2015. – 320с. : ил. – Парал. тит. англ