



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»
Кафедра «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ ПО УЧЕБНО-ВЫЧИСЛИТЕЛЬНОМУ ПРАКТИКУМУ

Выполнил студент: Светличная Алина Алексеевна

Группа: ИУ7-13Б

Проверил: Кузнецова Ольга Владимировна

подпись, дата

Оценка _____ Дата _____

2020 г.

Оглавление

1. Условие задачи	3-4
2. Схема алгоритма	5-11
3. Описание программы	12-13
4. Код.....	14-16
5. Результаты работы.....	17
6. Заключение	18
7. Список литературы.....	18

Условие задачи. Арбитраж

Предпосылки

Использование компьютеров в сфере финансов в последнее время отмечается противоречием в электронной торговле — используемые для перепродаж валют при чрезвычайно малых колебаниях в цене, что является незаконным во многих Wall Street фирмах. Этика электронной торговли - новая область со многими нерешенными проблемами.

Проблема

Арбитраж — это перепродажа нескольких валют при небольших различиях в конверсионных расценках среди них, с целью получения прибыли. Например, если \$ 1.00 стоит 0.7 английского фунта стерлинга, 1,00£ - 9,5 французских франков, и 1,00F - 0.16\$, то торговец может начинать с \$ 1.00 и заработает $1 \times 0.7 \times 9.5 \times 0.16 = 1.064$ доллара, таким образом, получив прибыль 6.4 процентов.

Вы будете писать программу, которая определяет, может ли последовательность обменов валюты давать прибыль, как описано выше.

Для того чтобы был успешный обмен, он должен начинаться и заканчиваться той же самой валютой, но при этом может рассматриваться любая стартовая валюта.

Ввод

Входной файл состоит из одного или большее количество таблиц преобразования. Вы должны решить проблему обмена для каждой из таблиц во входном файле.

Каждой таблице предшествует целое число n , дающее размер таблицы. Максимальный размер — 20, минимальный — 2.

В таблице нет диагональных элементов (они приняты за единицу). Таким образом, первый ряд таблицы представляет собой преобразования между первой и $n-1$ страной, то есть, количество валюты страны i ($2 \leq i \leq n$), что быть куплено с одной единицей валюты 1 страны.

Таким образом, каждая таблица состоит из $n+1$ линий во входном файле: n числа стран и n линий, содержащих конверсионные расценки.

Для каждой таблицы во входном файле Вы должны определить, существует ли последовательность обменов, имеющих прибыль больше 1 процента (0.001). Если

последовательность существует, Вы должны напечатать последовательность обменов, которая кончается прибылью. Если имеется более одной последовательности, которые заканчиваются прибылью больше 1 процента, Вы должны напечатать последовательность минимальной длины (то есть, одна из последовательностей, в которой используется наименьшее количество обменов валютами, чтобы получить прибыль), которая лексически самая маленькая. Поскольку IRS (Внутренняя Служба Доходов Соединённых Штатов) следит за длиной операционных последовательностей, все получающие прибыль последовательности должны состоять из n или меньшего количества сделок, где n — размер таблицы. Последовательность 1 2 1 представляет два преобразования.

Если прибыль существует, Вы должны печатать последовательность обменов, которые кончаются прибылью. Последовательность напечатана как ряд i целых чисел, представляющий i^{th} линию таблицы преобразования (i страна). Первое целое число в последовательности - страна, от которой прибыльная последовательность начинается. Этим же числом также заканчивается последовательность.

Если не существует прибыли от n или меньшего количества сделок, то строка
no arbitrage sequence exists

Типовой ввод

3

1.2 0.89

0.88 5.1

1.1 0.15

4

3.1 0.0023 0.35

0.21 0.00353 8.13

200 180.559 10.339

2.11 0.089 0.06111

2

2.0

0.45

Типовой вывод

121

1241

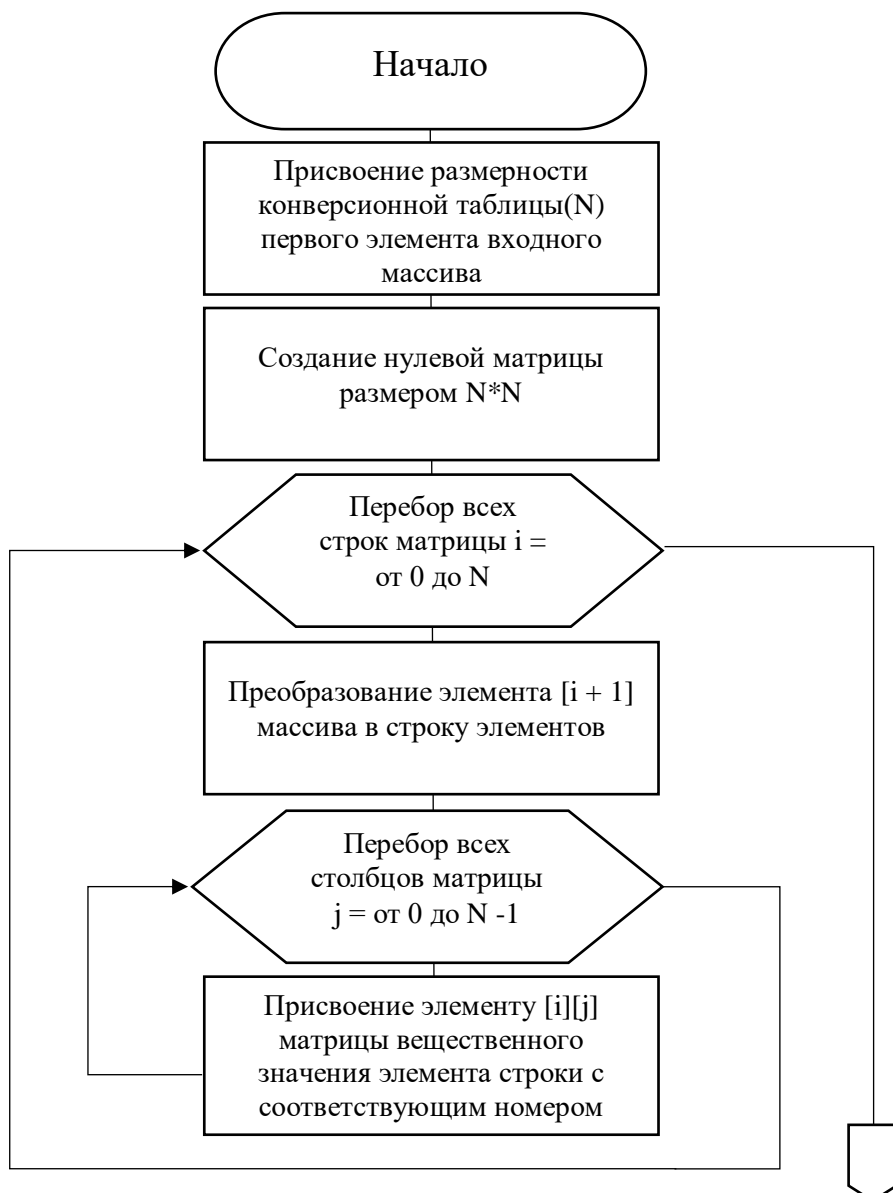
no arbitrage sequence exists

Схема алгоритма

Алгоритм №1 «Перенос данных из файла входных значений в программу»

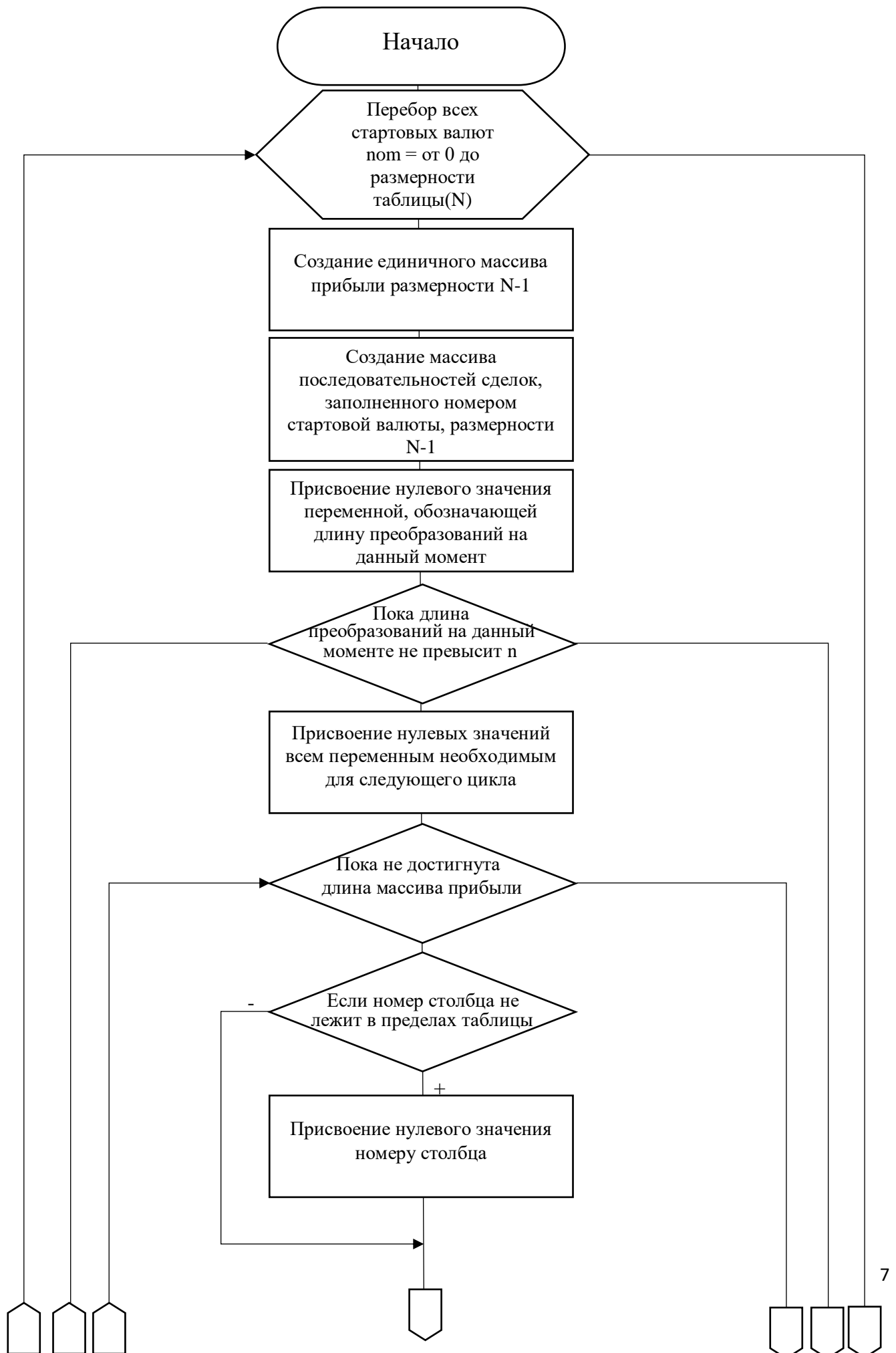


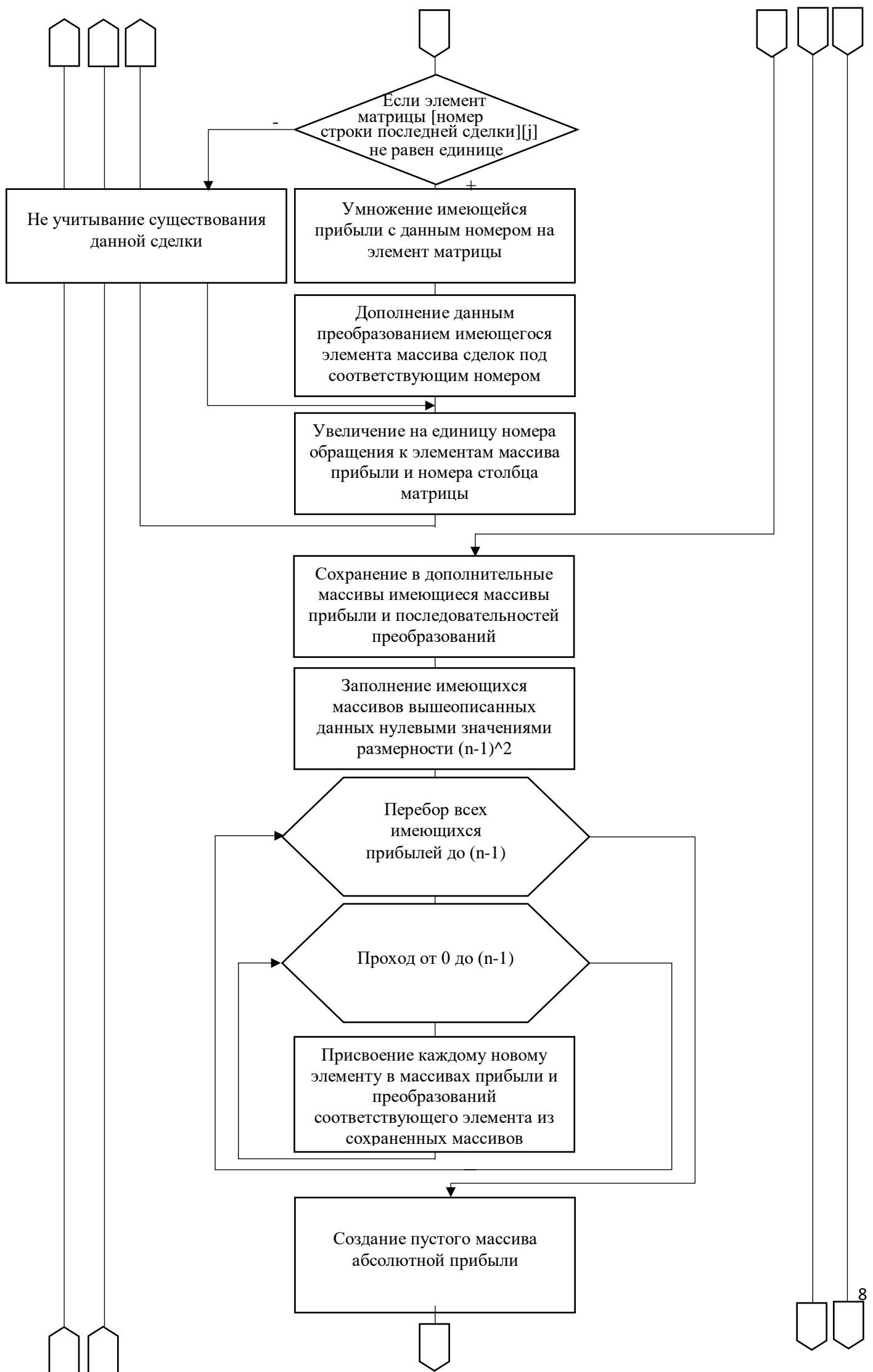
Алгоритм №2 «Преобразование входных значений к удобному виду (матрице)»

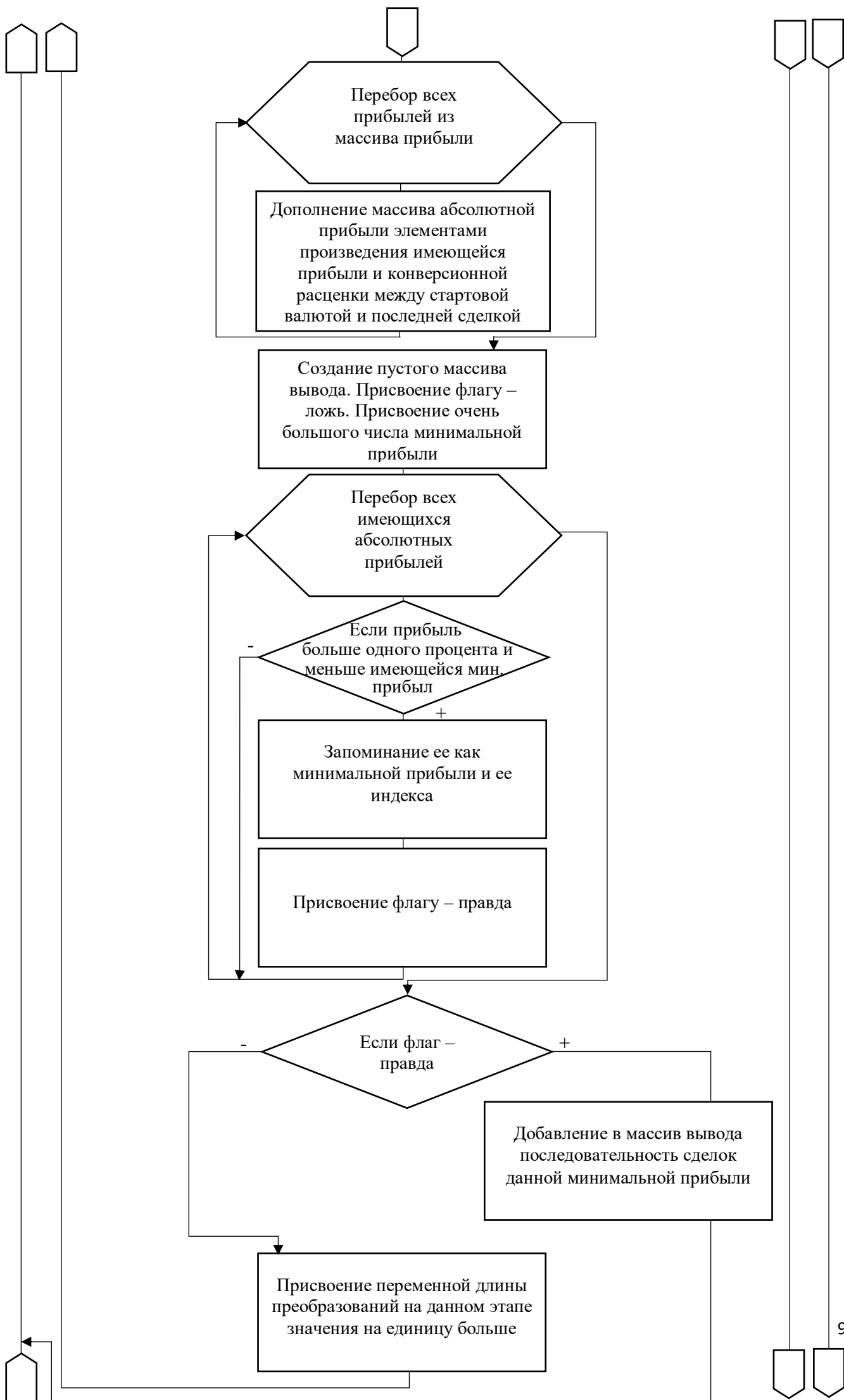


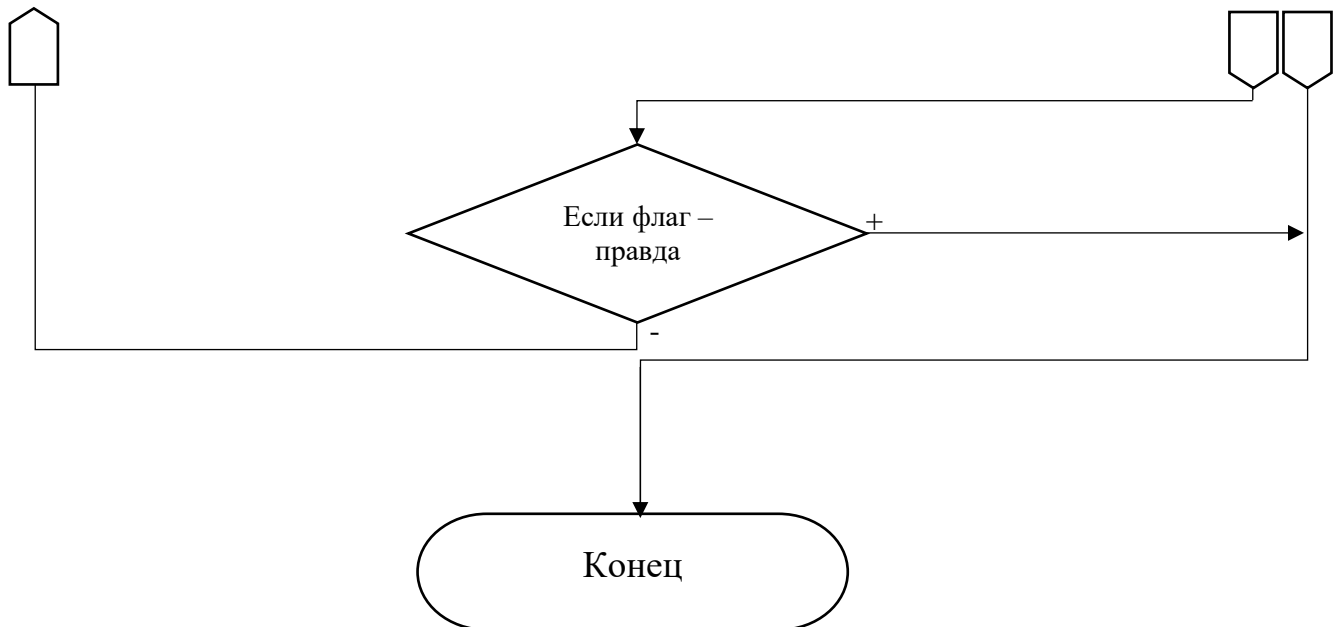


Алгоритм №3 «Обработка данных одной конверсионной таблицы. Получение прибыли»

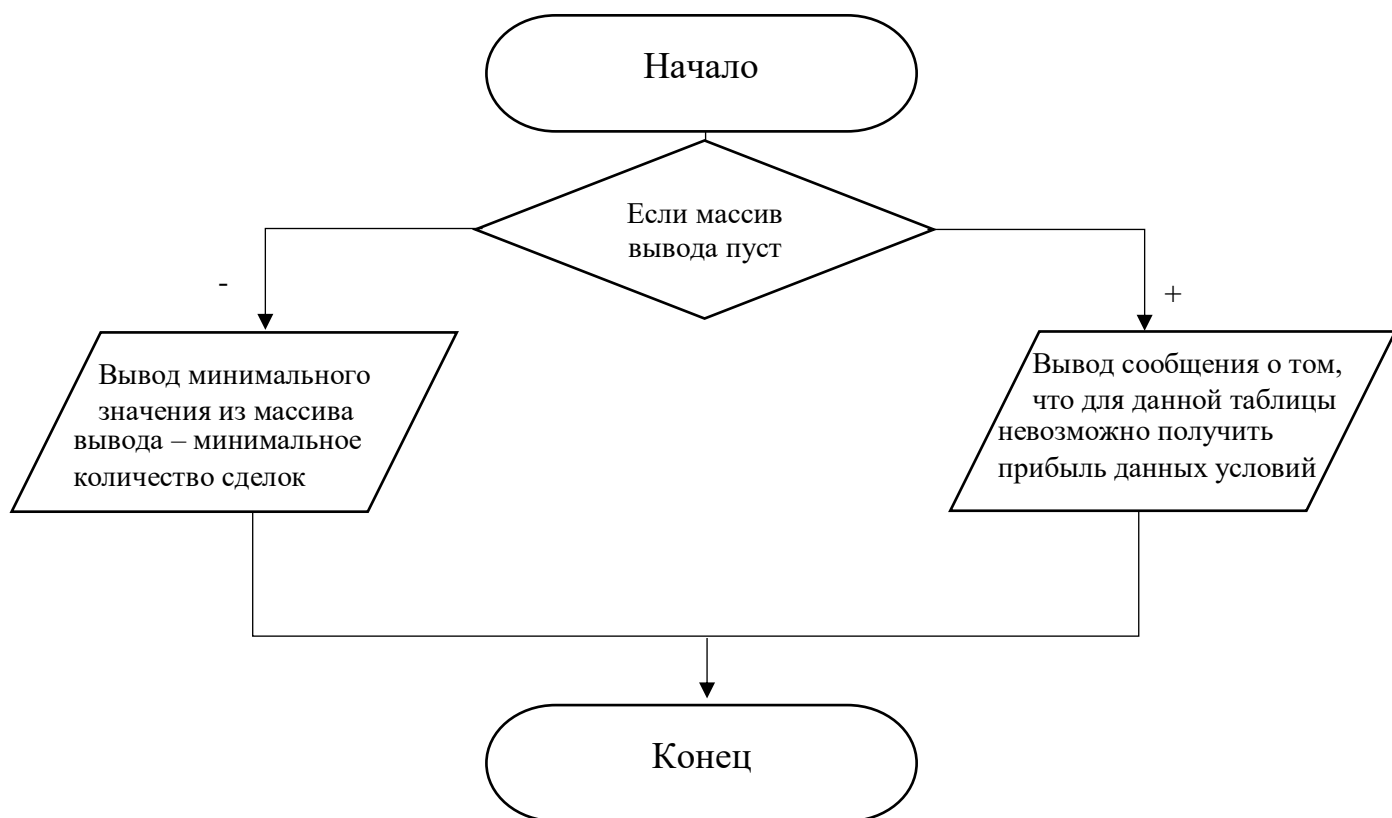




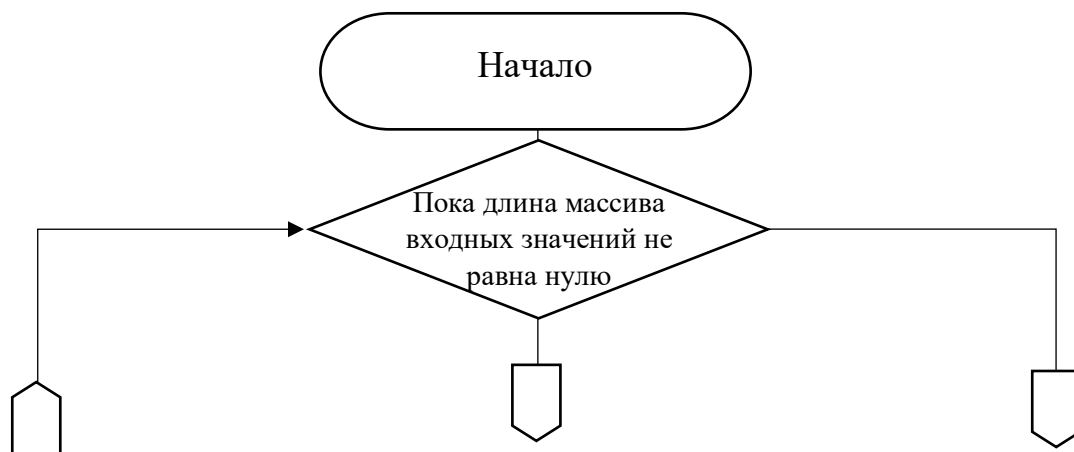


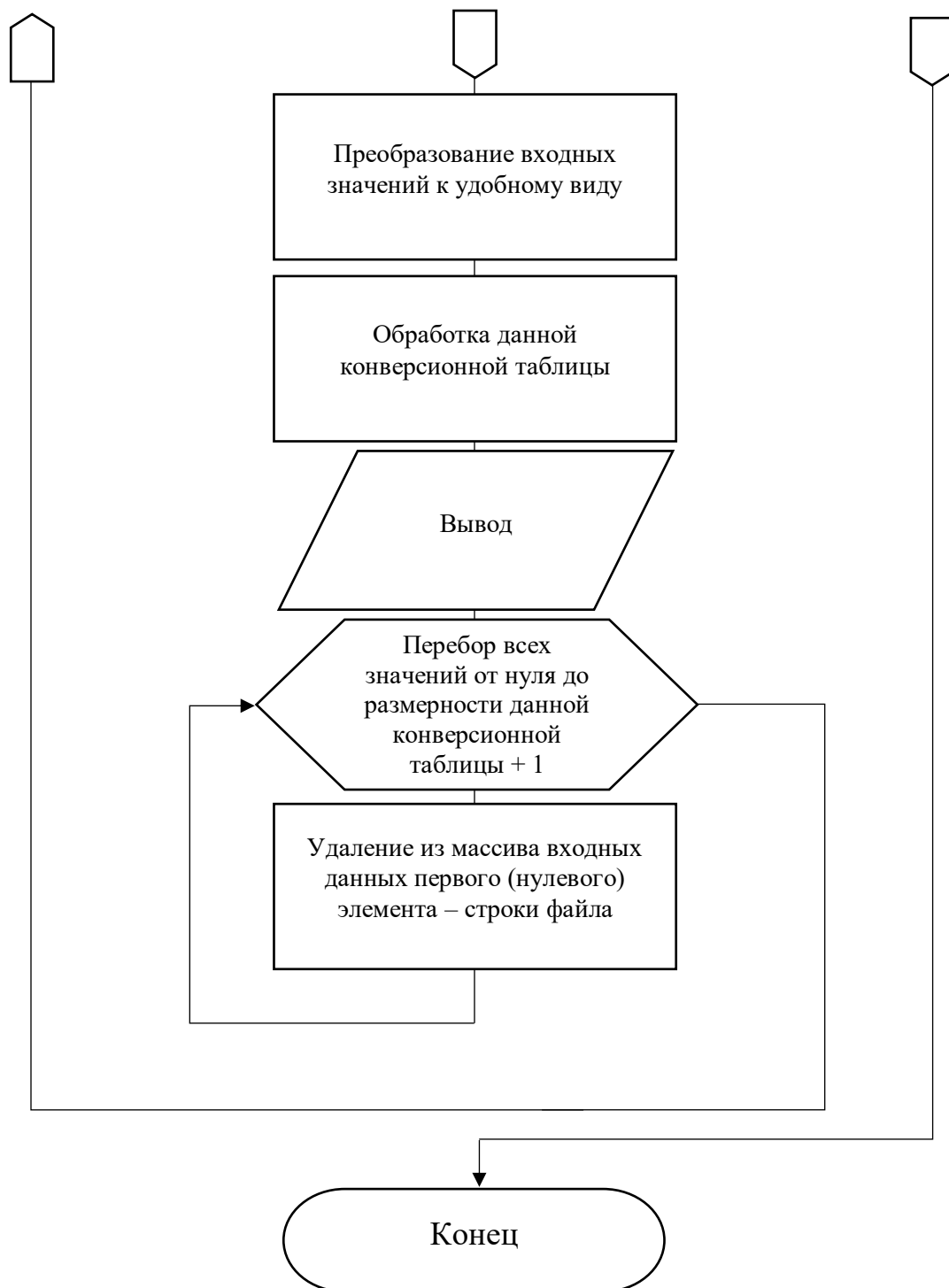


Алгоритм №4 «Вывод данных»



Алгоритм №5 «Обработка всех конверсионных таблиц»





Описание программы

Первым шагом программа построчно считывает все элементы файла входных значений в один массив. Учитывая выбранную стратегию «отрезания» обработанных элементов, можно утверждать, что программа обработает все входящие конверсионные таблицы тогда и только тогда, когда в массиве входных значений не останется элементов, а, следовательно, можно запустить цикл, при котором программа будет выполнять основные действия до тех пор, пока длина массива входных значений не окажется равна нулю.

Вторым важным шагом является верная обработка массива входных значений. По условию нулевой элемент данного массива обозначает размерность одной конверсионной таблицы, определенными функциями последующие элементы входного массива необходимо преобразовать в конверсионную матрицу, добавляя на элементы главной диагонали единицы.

Также по условию необходимо найти такую прибыль, которая будет достигаться за количество преобразований меньшее, чем размерность данной конверсионной матрицы, а значит, вновь резонен цикл, который будет совершать дальнейшие действия до тех пор, пока количество преобразований на данный момент не превысит размерность таблицы.

И последним важным циклом является прохождение по всем стартовым валютам.

Далее необходимо создать единичный массив, предназначенный для прибыли, и нулевой массив, нужный для записи последовательности преобразований для данной прибыли. Проходясь по всей конверсионной матрице, умножаем значение имеющейся прибыли в массиве на значения матрицы, получая все возможные варианты преобразований, а также приписываем в массив сделок соответствующую сделку, которая характеризуется номером столбца элемента. После по последней цифре элементов массива преобразований определяем строку конверсионной таблицы, необходимую для конечного преобразования в начальную валюту, умножаем соответствующее значение на имеющуюся прибыль. Если конечная прибыль превышает 1% и самая маленькая в имеющемся массиве прибыли, то данная последовательность преобразований является искомой и записывается в переменную

выводы, далее размерность преобразований будет лишь увеличиваться, а, следовательно, обработку этой таблицы можно закончить. Если же прибыль не удовлетворяет условию, то ее необходимо сохранить, избегая шага перевода в стартовую валюту, для дальнейшей обработки более большими преобразованиями. Однако если и за весь цикл не будет найдена такая последовательность, то программа выведет соответствующее значение, индикатором данной ситуации будет служить пустая переменная вывода.

Код

```
file = open('Входные значения.txt')
lines = file.readlines()
while len(lines) != 0:
    n = int(lines[0])
    mas = [[0 for i in range(n)] for j in range(n)]
    for i in range(n):
        lines[i + 1] = list(lines[i + 1].split())
        for j in range(n - 1):
            mas[i][j] = float(lines[i + 1][j])
    for i in range(0, n):
        isok = False
        for j in range(0, n):
            if i == j:
                isok = True
                sdvig = mas[i][j]
                mas[i][j] = 1
            if i != j and isok:
                mas[i][j], sdvig = sdvig, mas[i][j]
    for nom in range(n):
        pbl = [1] * (n - 1)
        pere = [nom + 1] * (n - 1)
        n1 = 0
        while n1 < n:
            k = 0
            j = 0
            while k < (len(pbl)):
                if j > (n - 1):
                    j = 0
                if mas[int(pere[k] % 10) - 1][j] != 1:
                    pbl[k] = pbl[k] * mas[int(pere[k] % 10) - 1][j]
                    pere[k] = pere[k] * 10 + (j + 1)
                else: k += 1
                k += 1
                j += 1
            zpb1 = pbl
```

```

zpere = pere
pbl = [0] * (n-1) ** 2
pere = [0] * (n-1) ** 2
m = 0
for k1 in range(n-1):
    for j in range(n-1):
        pbl[m] = zpbl[k1]
        pere[m] = zpere[k1]
        m += 1
pr = []
for k1 in range(len(pbl)):
    pr.append(pbl[k1] * mas[int(pere[k1] % 10) - 1][nom])
vv = []
isok = False
minpr = 1000
for i in range(len(pbl)):
    if pr[i] > 1.001 and pr[i] < minpr:
        minpr = pr[i]
        mini = i
        isok = True
if isok:
    vv.append(pere[mini] * 10 + nom + 1)
if isok:
    break
n1 += 1
if isok:
    break
if vv == []:
    print('no arbitrage sequence exists')
else:
    vv = min(vv)
    vv = list(str(vv))
    for i in range(len(vv)):
        print(vv[i], end = ' ')
    print()
for i in range(n + 1):

```

```
lines.pop(0)
```


Результаты работы

Вид входного файла	Полный вывод (результат работы программы)
2 2.0 0.45 3 1.2 0.89 0.88 5.1 1.1 0.15 4 3.1 0.0023 0.35 0.21 0.00353 8.13 200 180.559 10.339 2.11 0.089 0.06111	no arbitrage sequence exists 1 2 1 1 2 3 1
2 1009 0.001	1 2 1
2 10009 0.0001	no arbitrage sequence exists
7 0.1 1.2 27 5.65 0.894 9.1 0.93 4.5 6.003 5.65 0.1 0.0001 0.91 2.3 4.12 0.12 5.432 34.7 0.75 5.6 73.1 12.5 23.4 0.1 1.23 1.34 1.56 1.78 1.32 1.65 0.91 0.92 0.56 3.4 0.06 0.34 17.1 18.3 4.5 0.12 4.6 7.0001	1 3 1
3 0.2 3.2 0.2 3.2 0.2 3.2	2 3 2

Заключение

В результате выполненного практического задания я научилась работать с текстовыми файлами на языке программирования Python, обрабатывать числовые матрицы и самостоятельно писать работоспособную программу для конкретных целей. И все это приближает меня к полному освоению выбранной профессии.

Список литературы

1. Ютуб-канал «egoroff_channel» видеоролик «Работа с файлами в Python. Чтение и запись данных» – https://www.youtube.com/watch?v=oRr_bEXJbV0
2. Интернет-ресурс «Python 3 для начинающих» - <https://pythonworld.ru/>
3. Курс лекций преподавателя программирования С.В. Борисова