



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение высшего  
образования  
«МИРЭА - Российский технологический университет»

**РТУ МИРЭА**

---

Институт искусственного интеллекта  
Кафедра общей информатики

**ОТЧЕТ**  
**ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 12**  
**Элементы алгоритмизации и процедурного программирования**  
**по дисциплине**  
**«ИНФОРМАТИКА»**

Выполнил студент группы ИМБО-01-22

Ким К.С.

Принял  
Ассистент

Павлова Е.С.

Практическая работа выполнена

«\_» декабря 2022 г.

Подпись студента

«Зачтено»

«\_» декабря 2022 г.

Подпись преподавателя

Москва 2022

## **СОДЕРЖАНИЕ**

1 ПОСТАНОВКА ЗАДАЧИ .....	3
1.1 Персональный вариант .....	3
2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ .....	4
2.1 Блок-схемы алгоритмы программы .....	4
2.2 Код программы .....	13
2.3 Примеры тестирования .....	18
3 ВЫВОДЫ.....	20
4 ИНФОРМАЦИОННЫЙ ИСТОЧНИК .....	21

# **1 ПОСТАНОВКА ЗАДАЧИ**

Требуется разработать блок-схему алгоритма и написать программу обработки данных в соответствии с выбранным и согласованным с преподавателем вариантом. При этом требуется контролировать типы и диапазоны вводимых данных, а также предусмотреть обработку других исключительных ситуаций (если они есть), например, ситуацию деления на ноль. Блок-схема должна быть полной, то есть должна описывать и процесс диалога с пользователем, и контроль вводимых данных, и подпрограммы вычислений с обработкой возможных исключительных операций. Блок-схема должна изображаться по ГОСТу. При обнаружении ошибки ввода или ошибки вычислений программа должна информативно уведомлять пользователя о причине ошибки. Если ошибка произошла на этапе ввода данных, то программа должна просить пользователя повторить ввод.

## **1.1 Персональный вариант**

2.8. Создать квадратную матрицу размера  $M \times M$ , где  $M$  является целым числом из диапазона  $[2, 5]$ . Конкретный размер матрицы задается пользователем. Матрица содержит только целые числа из диапазона  $[1, 100]$ , которые могут быть как случайными, так и вводиться пользователем. Отсортировать по возрастанию элементы, принадлежащие или лежащие выше побочной диагонали матрицы, остальные элементы умножить на минус один. Результаты обработки матрицы вывести на экран.

## 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

### 2.1 Блок-схемы алгоритмов программы

Блок-схема алгоритма, решающего поставленную задачу показана на рис.

1–9.

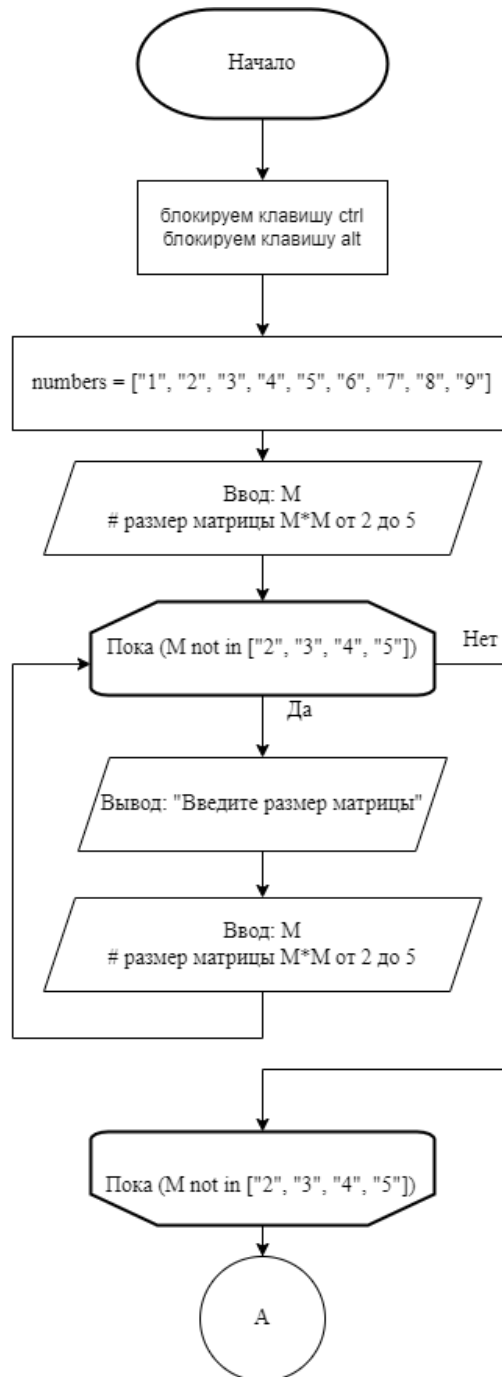


Рисунок 1 – Блок-схема программы, часть 1

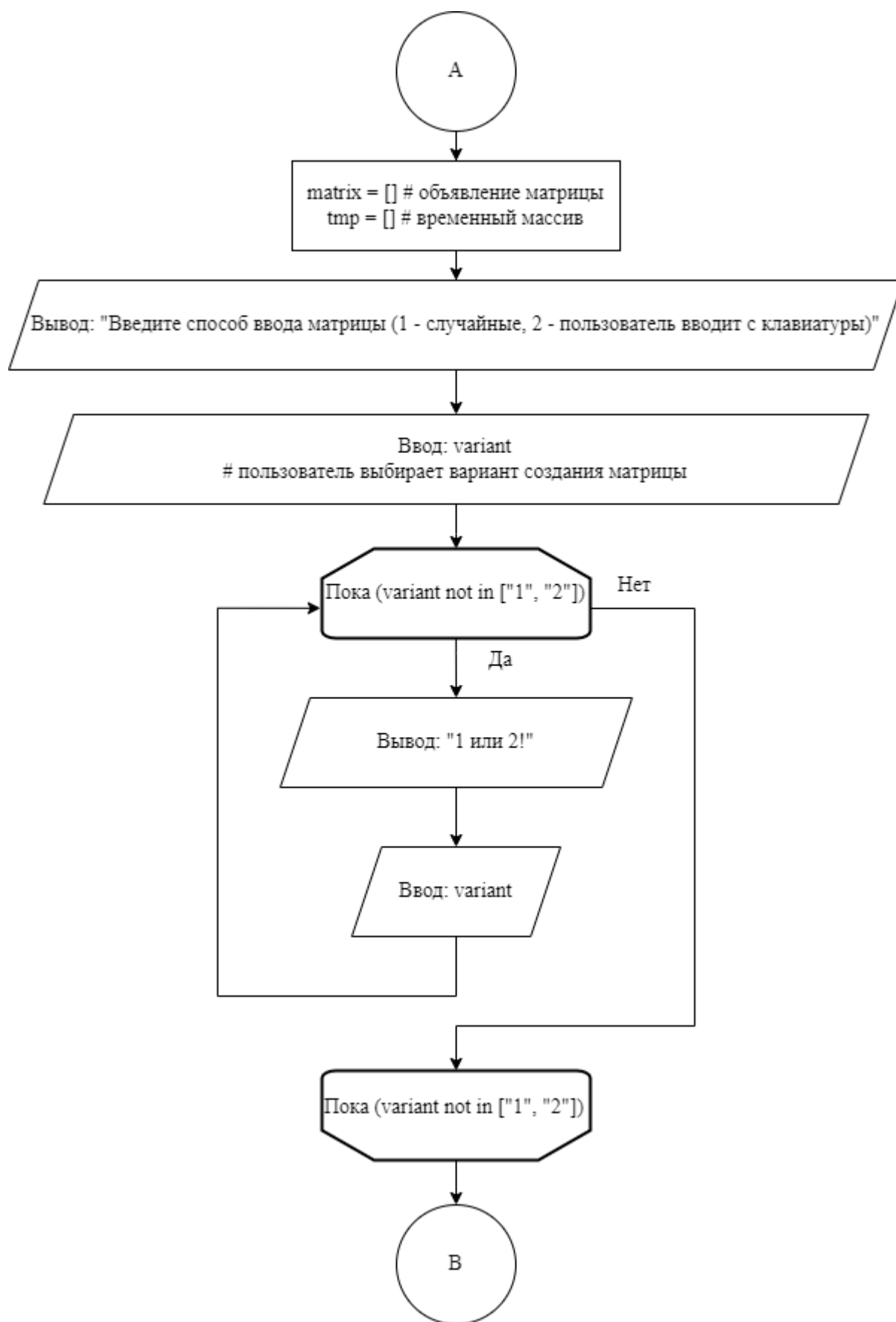


Рисунок 2 – Блок-схема программы, часть 2

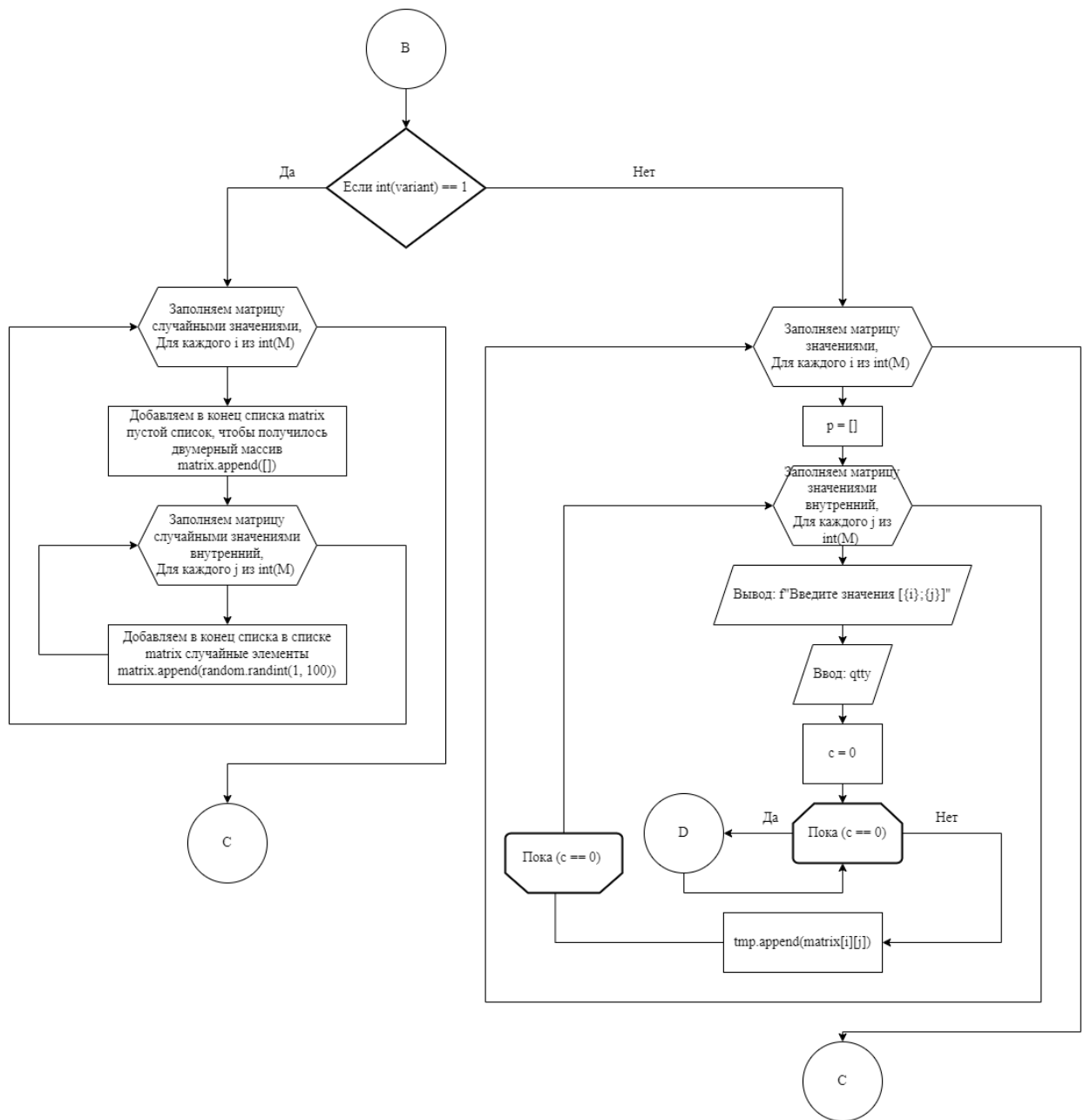


Рисунок 3 – Блок-схема программы, часть 3

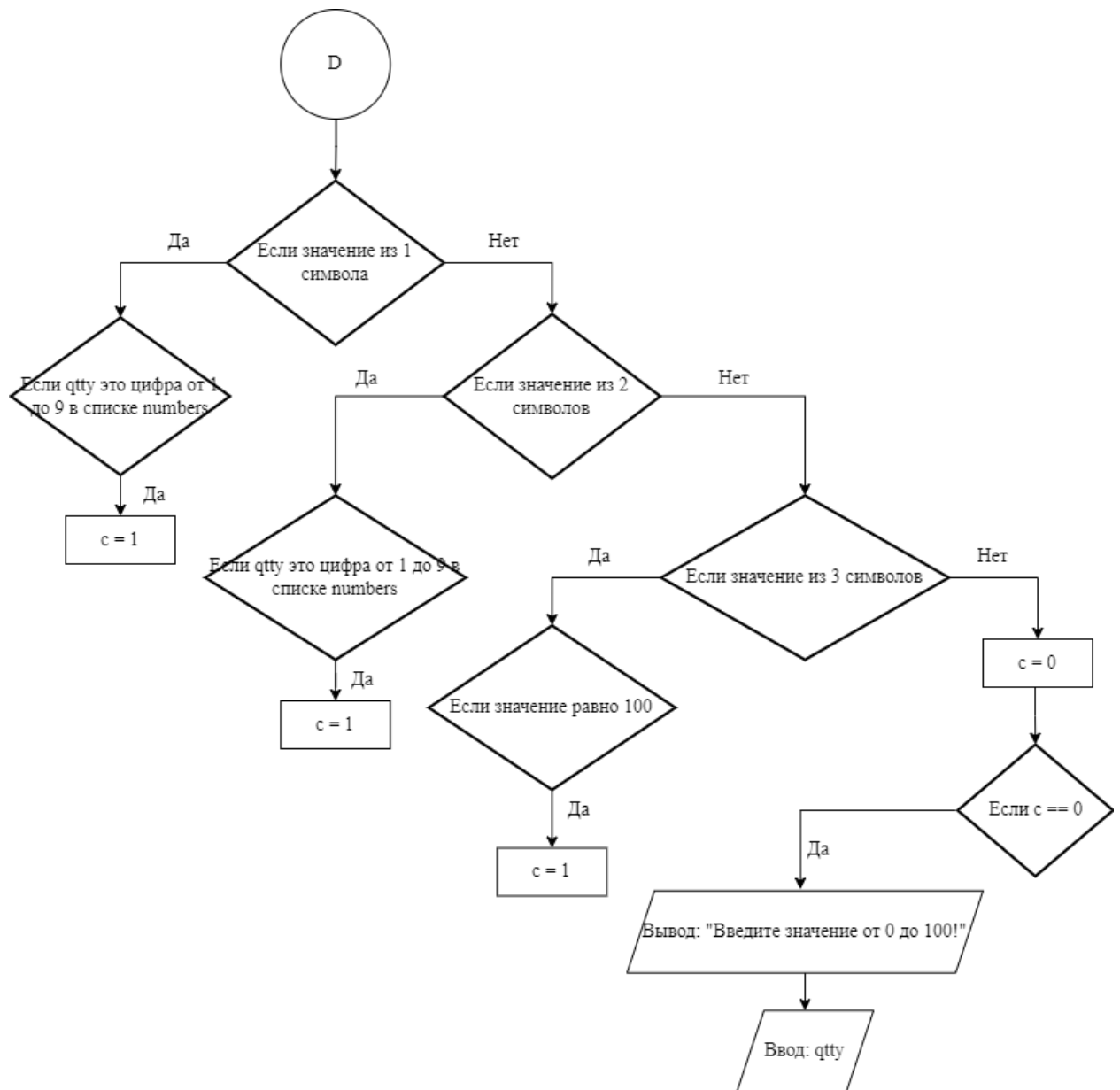


Рисунок 4 – Блок-схема программы, часть 4

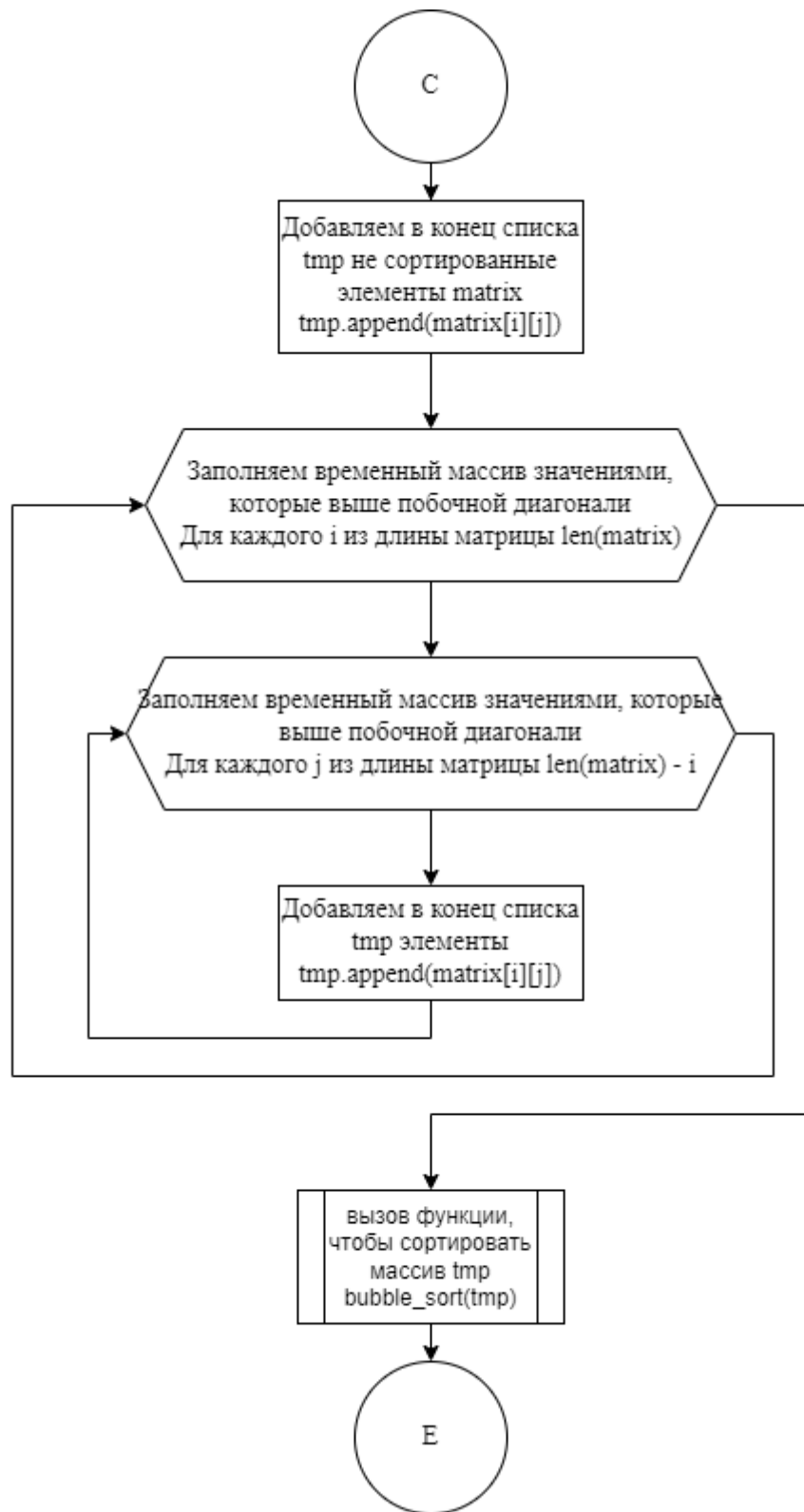


Рисунок 5 – Блок-схема процедуры, часть 5



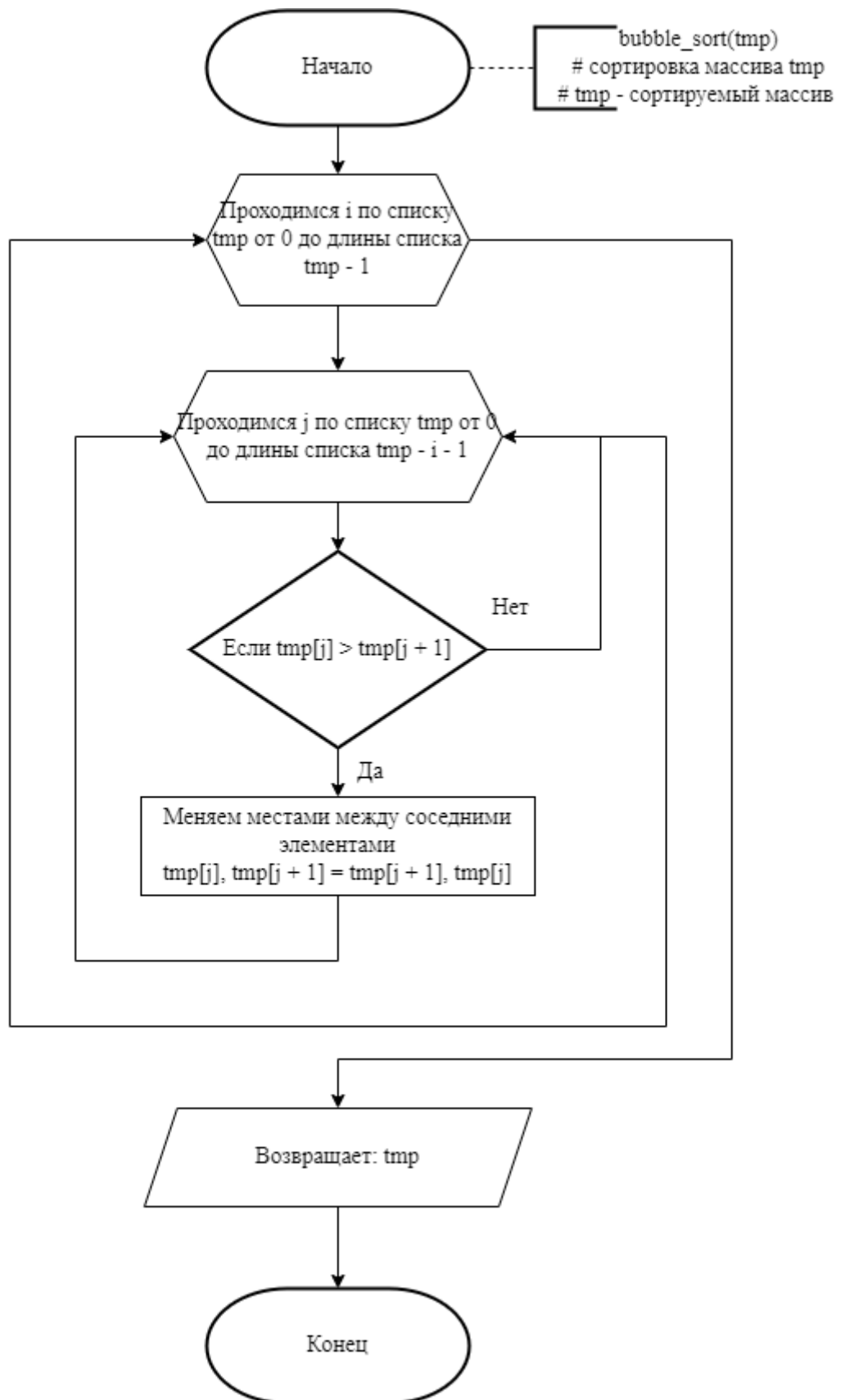


Рисунок 6 – Блок-схема процедуры bubble\_sort

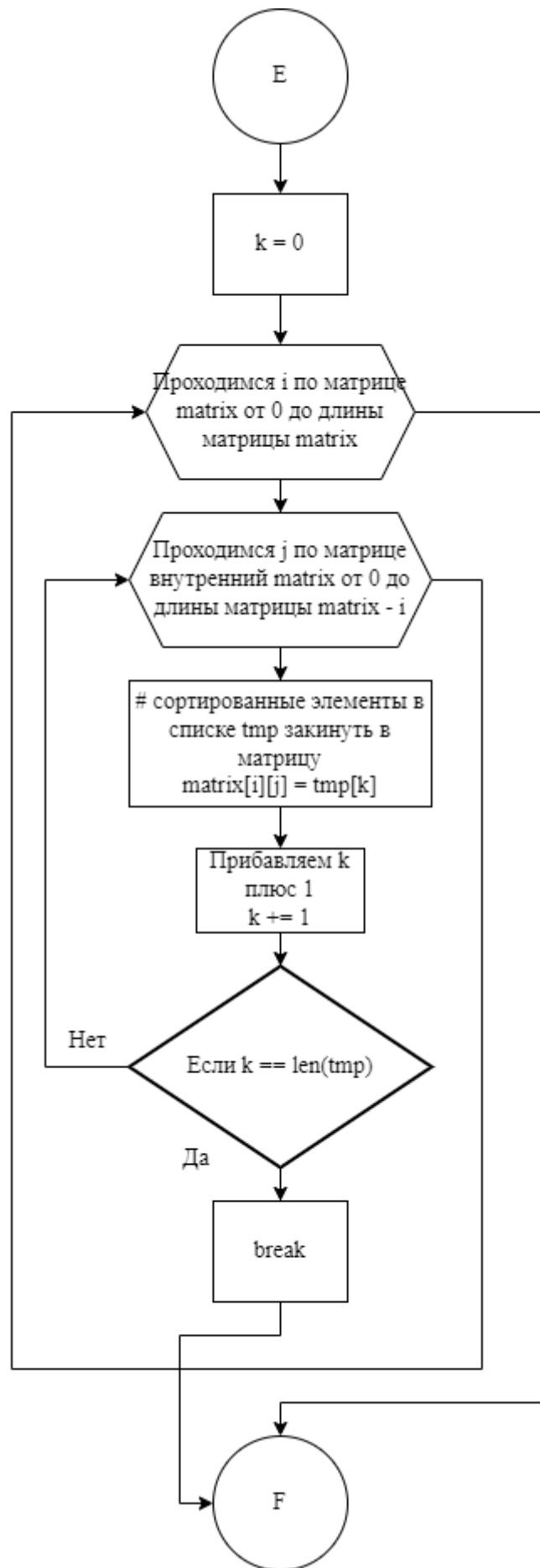


Рисунок 7 – Блок-схема программы, часть 7

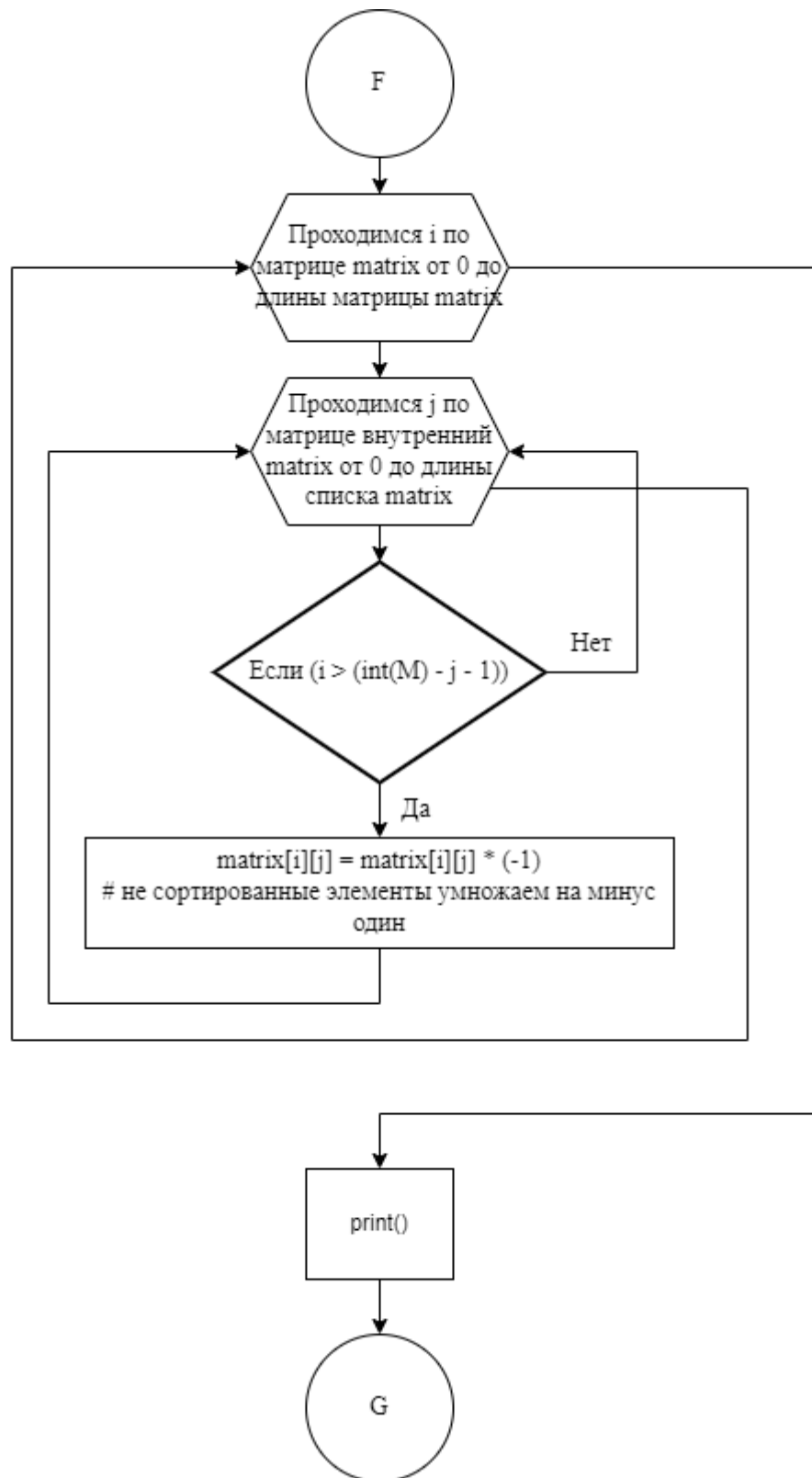


Рисунок 8 – Блок-схема программы, часть 8

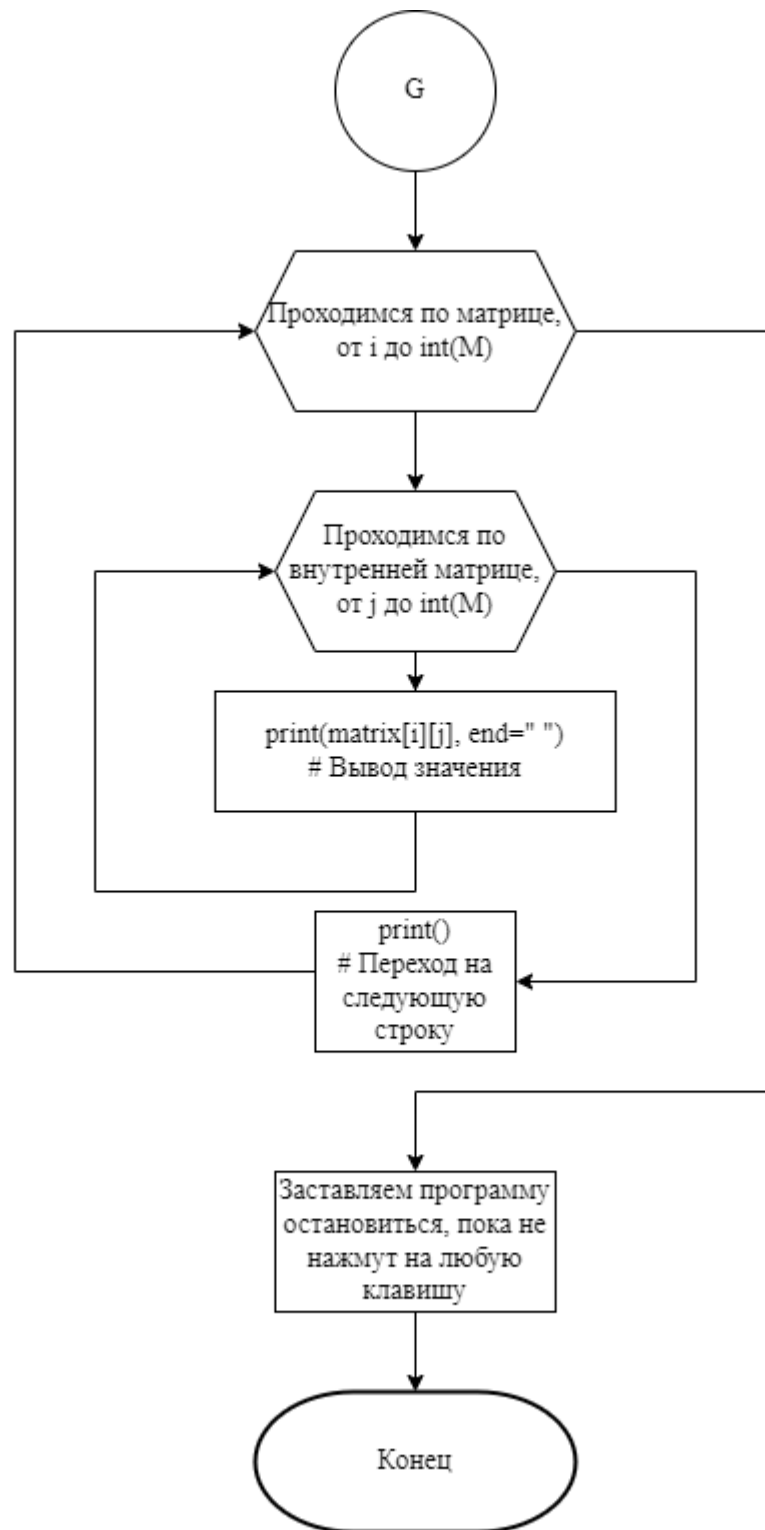


Рисунок 9 – Блок-схема программы, часть 9

На основе блок-схемы был реализован код на языке Python 3 в среде разработки PyCharm.

## 2.2 Код программы

Код программы на языке программирования Python представлен листингом 1.

Листинг 1 – Код программы

```
import random # подключение библиотеки для реализации случайных чисел
```

```
import keyboard # подключение модуля для блокировки клавиш
```

```
import os # подключаем модуль для того, чтобы в конце программа могла ждать, пока не будет нажата любая клавиша
```

```
def bubble_sort(tmp):  
    for i in range(len(tmp) - 1):  
        for j in range(len(tmp) - i - 1):  
            if tmp[j] > tmp[j + 1]:  
                tmp[j], tmp[j + 1] = tmp[j + 1], tmp[j]  
    return tmp
```

```
keyboard.block_key("ctrl")
```

```
keyboard.block_key("alt")
```

```
numbers = ["1", "2", "3", "4", "5", "6", "7", "8", "9"]
```

```
print("Какой диапазон для размера матрицы?")
```

```
M = input() # размер матрицы M*M от 2 до 5
```

```
while (M not in ["2", "3", "4", "5"]):
```

```
    print("Размер матрицы должен быть от 2 до 5!")
```

```
    M = input()
```

```
matrix = [] # объявление матрицы
```

```
tmp = [] # временный массив
```

```
print("Введите способ ввода матрицы (1 - случайные, 2 - пользователь  
вводит с клавиатуры)")
```

```

variant = input() # пользователь выбирает вариант создания матрицы
while (variant not in ["1", "2"]):
    print("1 или 2!")
    variant = input()
if (int(variant) == 1):
    for i in range(int(M)):
        matrix.append([])
        for j in range(int(M)):
            matrix[i].append(random.randint(1, 100))

    print()
    print("Вывод изначальной матрицы")
    for i in range(int(M)): # вывод изначальной матрицы
        for j in range(int(M)):
            print(matrix[i][j], end=" ")
        print()

    for i in range(len(matrix)): # закидываю элементы выше побочной
диагонали
        for j in range(len(matrix) - i):
            tmp.append(matrix[i][j])

    bubble_sort(tmp) # сортируем элементы

    k = 0
    for i in range(len(matrix)): # сортированные элементы закидываю в
матрицу
        for j in range(len(matrix) - i):
            matrix[i][j] = tmp[k]
            k += 1
        if (k == len(tmp)):

```

```
break
```

```
else:
```

```
    for i in range(int(M)): # наполнение матрицы пользовательскими  
        числами
```

```
        p = []
```

```
        for j in range(int(M)):
```

```
            print(f"Введите значения [{i};{j}]"
```

```
            qty = input()
```

```
            c = 0
```

```
            while (c == 0):
```

```
                if (len(qty) == 1):
```

```
                    if (qty[0] in numbers):
```

```
                        c = 1
```

```
                elif (len(qty) == 2):
```

```
                    if (qty[0] in numbers and qty[1] in numbers):
```

```
                        c = 1
```

```
                elif (len(qty) == 3):
```

```
                    if (int(qty) == 100):
```

```
                        c = 1
```

```
            else:
```

```
                c = 0
```

```
            if (c == 0):
```

```
                print("Введите значение от 0 до 100!")
```

```
                qty = input()
```

```
            p.append(int(qty))
```

```
        matrix.append(p)
```

```
print()
```

```
print("Вывод изначальной матрицы")
```

```
for i in range(int(M)): # вывод изначальной матрицы
```

```

        for j in range(int(M)):
            print(matrix[i][j], end=" ")
        print()

    for i in range(len(matrix)): # закидываю элементы выше побочной
диагонали
        for j in range(len(matrix) - i):
            tmp.append(matrix[i][j])

    bubble_sort(tmp) # сортируем элементы

    k = 0
    for i in range(len(matrix)): # сортированные элементы закидываю в
матрицу
        for j in range(len(matrix) - i):
            matrix[i][j] = tmp[k]
            k += 1
            if (k == len(tmp)):
                break

    for i in range(int(M)): # оставшиеся не сортированные элементы
умножаю на минус один
        for j in range(int(M)):
            if (i > (int(M) - j - 1)):
                matrix[i][j] = matrix[i][j] * (-1)

```



```
print()
print("Вывод конечной матрицы")
for i in range(int(M)): # вывод конечной матрицы
    for j in range(int(M)):
        print(matrix[i][j], end=" ")
    print()
os.system("pause")
```

## 2.3 Примеры тестирования

Было проведено тестирование программы, которое показало, что работает корректно. Результаты тестирования представлены рисунками 10–11.

```
Какой диапазон для размера матрицы?  
4  
Введите способ ввода матрицы (1 - случайные, 2 - пользователь вводит с клавиатуры)  
1  
  
Вывод изначальной матрицы  
11 25 54 40  
95 19 19 8  
62 89 3 57  
35 51 42 60  
  
Вывод конечной матрицы  
11 19 19 25  
35 40 54 -8  
62 89 -3 -57  
95 -51 -42 -60  
  
Process finished with exit code 0
```

Рисунок 10 – Тестирование со случайными числами

```
Какой диапазон для размера матрицы?
4
Введите способ ввода матрицы (1 - случайные, 2 - пользователь вводит с клавиатуры)
2
Введите значения [0;0]
2
Введите значения [0;1]
123
Введите значение от 0 до 100!
21
Введите значения [0;2]
74
Введите значения [0;3]
65
Введите значения [1;0]
53
Введите значения [1;1]
2
Введите значения [1;2]
43
Введите значения [1;3]
6
Введите значения [2;0]
34
Введите значения [2;1]
35
Введите значения [2;2]
23
Введите значения [2;3]
6
Введите значения [3;0]
88
Введите значения [3;1]
9
Введите значения [3;2]
99
Введите значения [3;3]
75

Вывод изначальной матрицы
2 21 74 65
53 2 43 6
34 35 23 6
88 9 99 75

Вывод конечной матрицы
2 2 21 34
35 43 53 -6
65 74 -23 -6
88 -9 -99 -75
```

Рисунок 11 – Тестирование с числами от клавиатуры

### **3 ВЫВОДЫ**

Разработана блок-схема алгоритма и написана программа обработки данных в соответствии с выбранным и согласованным с преподавателем вариантом. При этом требуется проконтролированы типы и диапазоны вводимых данных, а также предусмотрена обработка других исключительных ситуаций (если они есть), например, ситуацию выхода за границу диапазона. Блок-схема изображена по ГОСТу. При обнаружении ошибки ввода или ошибки вычислений программа уведомляет пользователя о причине ошибки. Если ошибка произошла на этапе ввода данных, то программа просит пользователя повторить ввод.

#### **4 ИНФОРМАЦИОННЫЙ ИСТОЧНИК**

1. Информатика: Методические указания по выполнению практических работ / С.С. Смирнов, Д.А. Карпов — М., МИРЭА — Российский технологический университет, 2020. — 102 с.