Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №1

«Реализация алгоритмов разветвляющейся и Циклической структуры на Python»

ДИСЦИПЛИНА: «Перспективные языки программирования»

Выполнил: студент гр. ИУК4-31Б _____ (Суриков Н.С.)
(подпись) (Ф.И.О.)

Проверил: _____ (Осипова О. В.)
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2024

Цель: формирование практических навыков процедурного программирования, разработки и отладки программ, овладение методами и средствами разработки и оформления технической документации

Задачи:

1. Изучить условные конструкции.
2. Изучить циклические конструкции.
3. Ознакомиться со структурой массивов.

Вариант 8

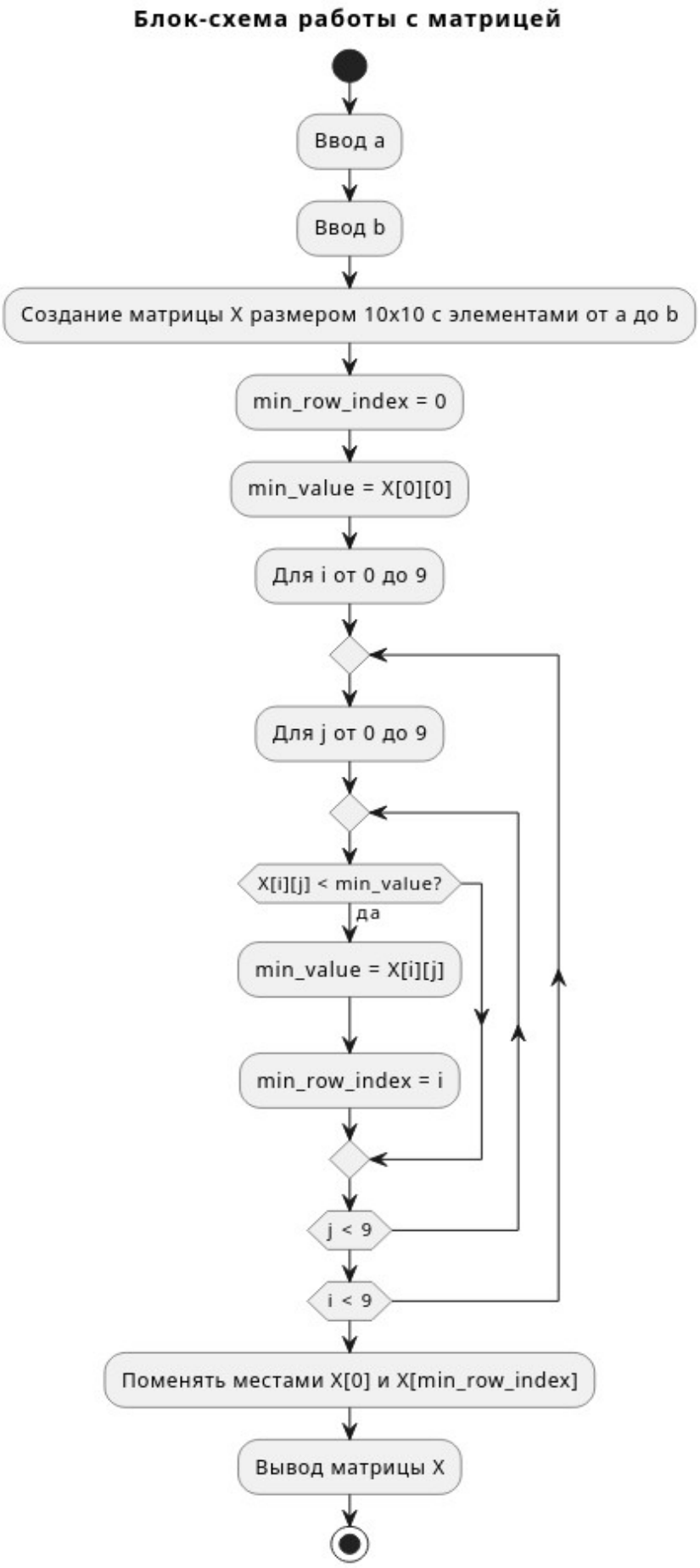
Задача 1:

Даны вещественные числа a и b ($a < b$). Сформировать матрицу X(10, 10), элементами которой являются вещественные случайные числа, равномерно распределенные на отрезке $[a, b]$. Найти в матрице строку с минимальным элементом и поменять ее местами с первой строкой.

Листинг программы 1:

```
import random

a = float(input("a: "))
b = float(input("b: "))

X = [[round(random.uniform(a, b), 3) for _ in range(10)] for _ in range(10)]
min_row_index = 0
min_value = X[0][0]
for i in range(10):
    for j in range(10):
        if X[i][j] < min_value:
            min_value = X[i][j]
            min_row_index = i

X[0], X[min_row_index] = X[min_row_index], X[0]
for row in X:
    print(row)
```

Блок схема программы 1:



Результат работы программы 1:

```
[3.73, 3.1, 3.044, 3.484, 3.113, 3.012, 3.301, 3.235, 3.591, 3.312]
[3.636, 3.139, 3.255, 3.652, 3.591, 3.633, 3.014, 3.793, 3.475, 3.867]
[3.442, 3.077, 3.642, 3.635, 3.15, 3.233, 3.084, 3.55, 3.608, 3.106]
[3.731, 3.371, 3.616, 3.751, 3.84, 3.716, 3.172, 3.388, 3.167, 3.915]
[3.785, 3.952, 3.104, 3.939, 3.396, 3.719, 3.428, 3.123, 3.791, 3.036]
[3.087, 3.737, 3.2, 3.239, 3.952, 3.995, 3.186, 3.65, 3.863, 3.949]
[3.143, 3.203, 3.957, 3.793, 3.709, 3.031, 3.523, 3.725, 3.368, 3.13]
[3.718, 3.329, 3.946, 3.099, 3.016, 3.563, 3.963, 3.498, 3.355, 3.666]
[3.369, 3.899, 3.743, 3.297, 3.229, 3.982, 3.568, 3.126, 3.023, 3.68]
[3.356, 3.752, 3.938, 3.952, 3.529, 3.31, 3.923, 3.023, 3.755, 3.059]
```

Задача 2:

Найти количество трехзначных чисел, кратных 15, но не кратных 30. Распечатать эти числа.

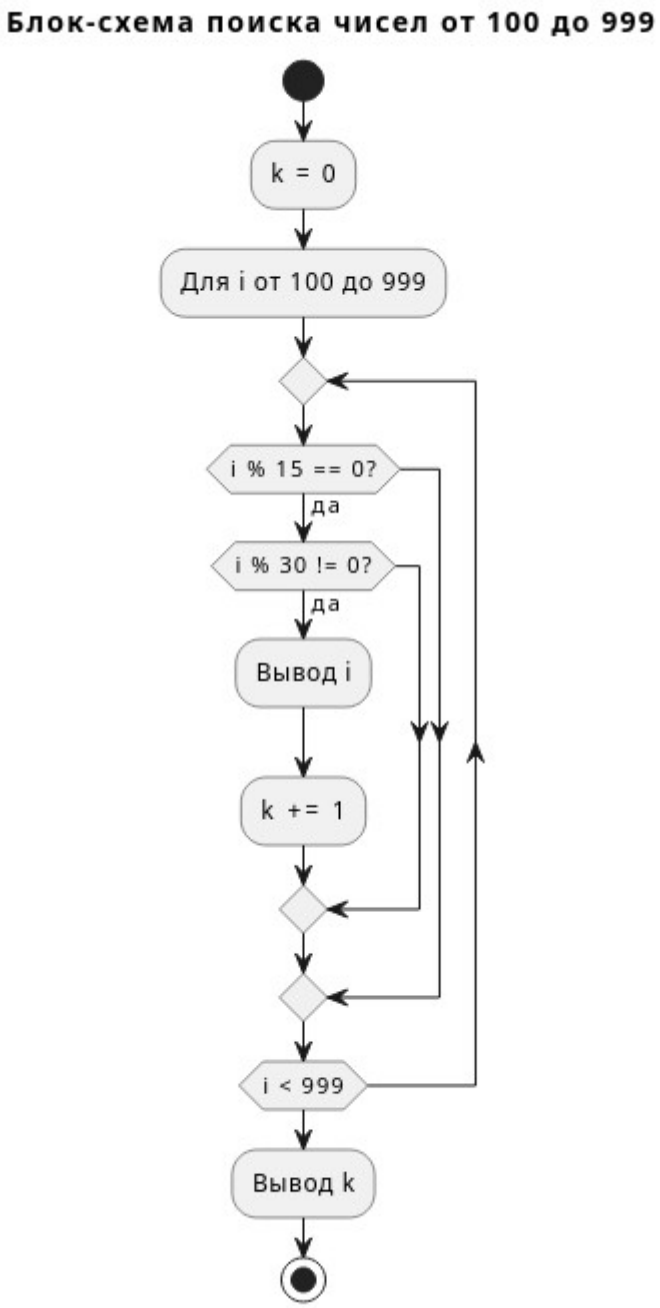
Листинг программы 2:

```
k = 0
for i in range(100, 1000):
    if i % 15 == 0 and i % 30 != 0:
        print(i)
        k += 1
print(k)

```

Результат работы программы 2:

105 135 165 195 225 255 285 315 345 375 405 435 465 495 525 555	585 615 645 675 705 735 765 795 825 855 885 915 945 975 30

Блок схема программы 2:



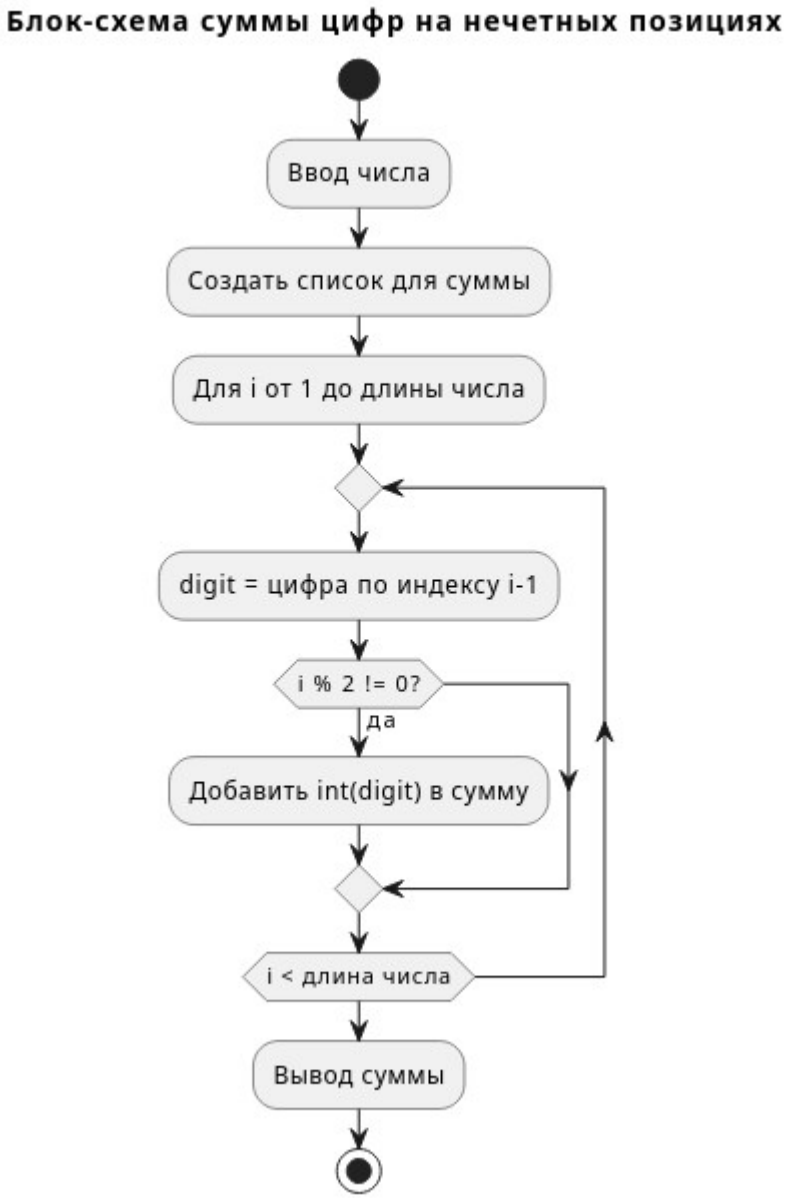
Задача 3:

Дано натуральное число N. Подсчитать сумму цифр этого числа, находящихся на нечетных позициях (нумерация позиций идет слева направо).

Листинг программы 3:

```
1  print(sum([int(digit) for i, digit in enumerate(input(), start=1) if i % 2 != 0]))
2
```

Блок схема программы 3:



Результат работы программы 3:

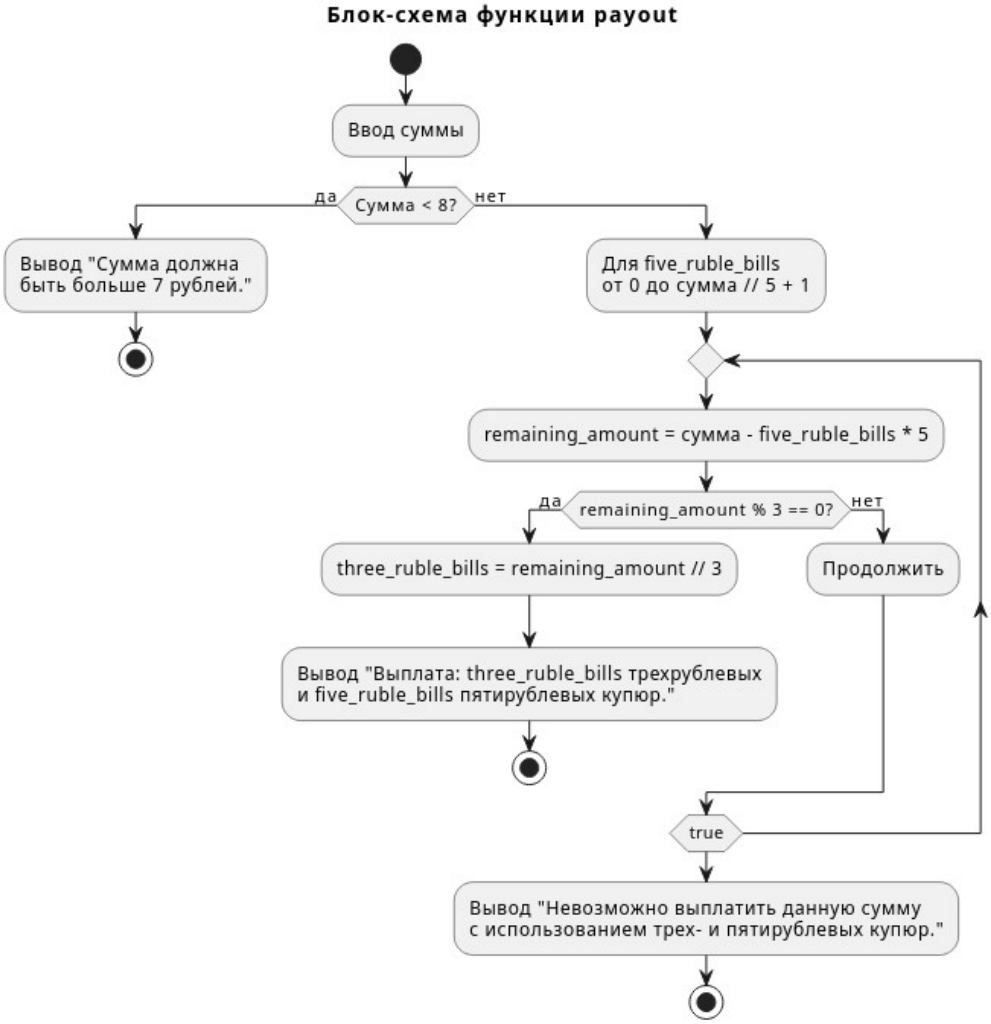
```
3456 → 8
```

Задача 4:
В кассе имеются только трех- и пятирублевые купюры (это было в далеком 1980 г.). Составить программу, которая “выплачивала “бы такими купюрами любую сумму более 7 рублей.

Листинг программы 4:

```python
def payout(amount):
    if amount < 8:
        return "Сумма должна быть больше 7 рублей."

    for five_ruble_bills in range(0, amount // 5 + 1):
        remaining_amount = amount - five_ruble_bills * 5
        if remaining_amount % 3 == 0:
            three_ruble_bills = remaining_amount // 3
            return f"Выплата: {three_ruble_bills} трехрублевых и {five_ruble_bills} пятирублевых купюр."

    return (
        "Невозможно выплатить данную сумму с использованием трех- и пятирублевых купюр."
    )


amount = int(input("Введите сумму для выплаты (более 7 рублей): "))
print(payout(amount))

```

Блок схема программы 4:



Результат работы программы 4:

```
Введите сумму для выплаты (более 7 рублей): 45
Выплата: 15 трехрублевых и 0 пятирублевых купюр.
```

Задача 5:

Элементами массива IM(N) являются числа 0 и 1. Отсортировать этот массив таким образом, чтобы все нули находились в начале, а единицы - в конце массива. Дополнительный массив не заводить.

Листинг программы 5:

```
1  bin_array = [0, 1, 0, 1, 1, 0, 0, 1, 0, 1]
2  bin_array.sort()
3  print(*bin_array)
5
```

Блок схема программы 5:

Результат работы программы 5:

```
0 0 0 0 0 1 1 1 1 1
```

Задача 6:

Во введенной строке подсчитать наибольшее количество одинаковых букв, идущих подряд.

Листинг программы 6:

```
import itertools


print(max(len(
    list(g)) for k, g in itertools.groupby(input("Введите строку: ")
                                           )))

```

Блок схема программы 6:

Результат работы программы 6:

```
Введите строку: qqqqwwqqeee
4
```

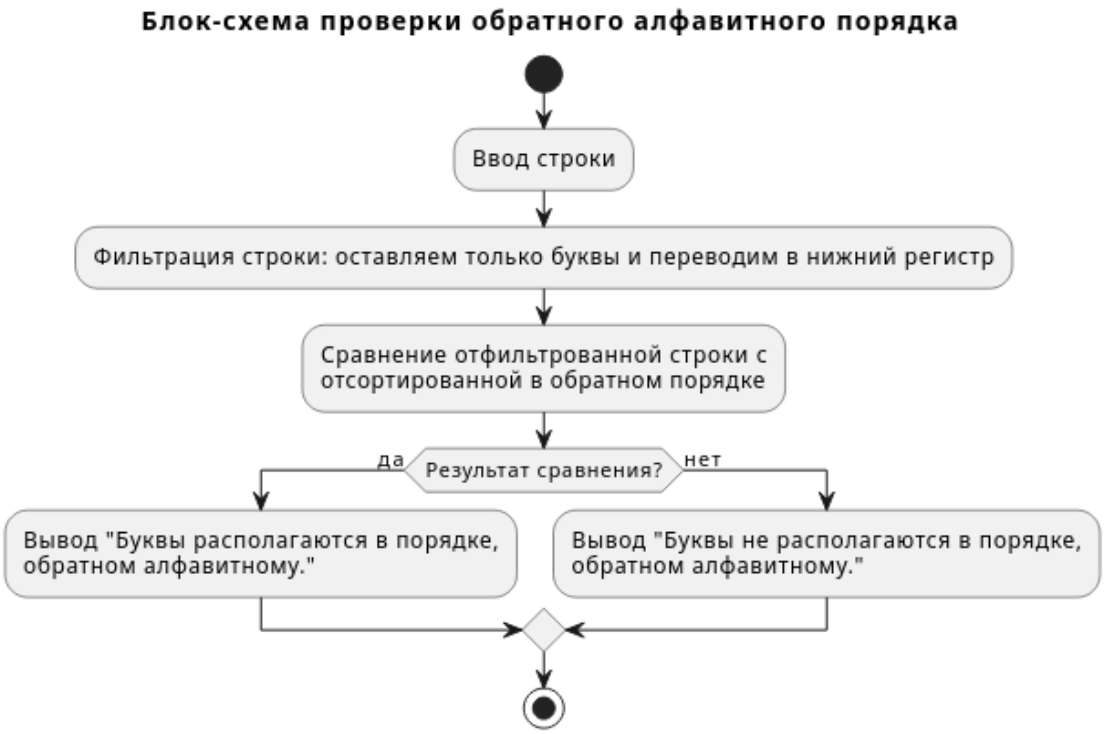
Задача 7:
Ввести строку и определить, располагаются ли буквы в ней в порядке, обратном алфавитному

Листинг программы 7:

```
def is_reverse_alphabetical(s):
    filtered_str = "".join(filter(str.isalpha, s.lower()))

    return filtered_str == "".join(sorted(filtered_str, reverse=True))


input_string = input("Введите строку: ")
```

```
if is_reverse_alphabetical(input_string):
    print("Буквы располагаются в порядке, обратном алфавитному.")
else:
    print("Буквы не располагаются в порядке, обратном алфавитному.")
```

Блок схема программы 7:



Результат работы программы 7:

```
Введите строку: абв
Буквы не располагаются в порядке, обратном алфавитному.

Введите строку: бав
Буквы не располагаются в порядке, обратном алфавитному.

Введите строку: вба
Буквы располагаются в порядке, обратном алфавитному.
```

Задача 8:

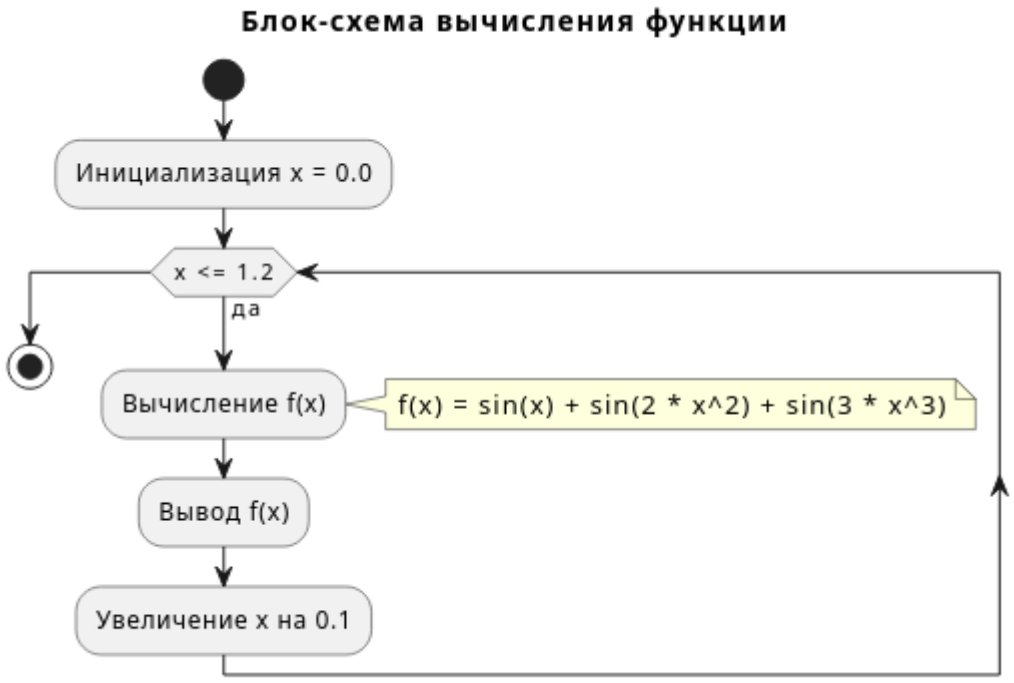
Вычислить значения функции $f(x)=\sin x+\sin 2x2+\sin 3x3$ *для значений аргумента x от 0.0 до 1.2 с шагом 0.1.*

Листинг программы 8:

```
import math


def f(x) -> float:
```

```
5       return math.sin(x) + math.sin(2 * x**2) + math.sin(3 * x**3)
6
7
8   x = 0.0
9   while x <= 1.2:
10      print(f"f({x:.1f}) = {f(x):.4f}")
11      x += 0.1
12
```

Блок схема программы 8:



Результат работы программы 8:

```
f(0.0) = 0.0000
f(0.1) = 0.1228
f(0.2) = 0.3026
f(0.3) = 0.5555
f(0.4) = 0.8948
f(0.5) = 1.3251
f(0.6) = 1.8276
f(0.7) = 2.3315
f(0.8) = 2.6748
f(0.9) = 2.5982
f(1.0) = 1.8919
f(1.1) = 0.7996
f(1.2) = 0.2998
```

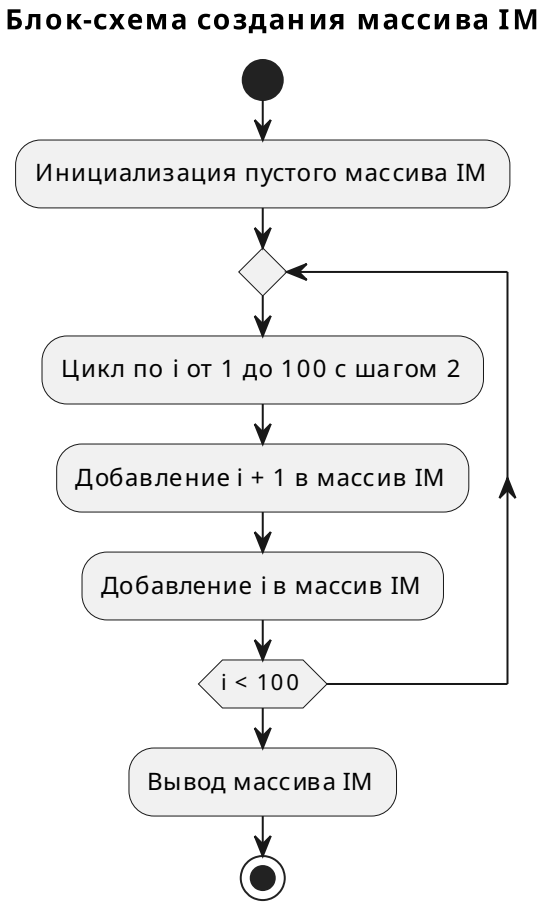
Задача 9:

Сформировать массив IM(100), элементами которого являются числа 2, 1, 4, 3, 6, 5, ... , 100, 99.

Листинг программы 9:

```
IM = []

for i in range(1, 101, 2):
    IM.append(i + 1)
    IM.append(i)

print("Массив IM(100):", *IM)

```

Блок схема программы 9:



Результат работы программы 9:

```
Массив IM(100): 2 1 4 3 6 5 8 7 10 9 12 11 14 13 16 15 18 17 20 19 22 21 24 23 26
25 28 27 30 29 32 31 34 33 36 35 38 37 40 39 42 41 44 43 46 45 48 47 50 49 52 51
54 53 56 55 58 57 60 59 62 61 64 63 66 65 68 67 70 69 72 71 74 73 76 75 78 77 80
79 82 81 84 83 86 85 88 87 90 89 92 91 94 93 96 95 98 97 100 99
```

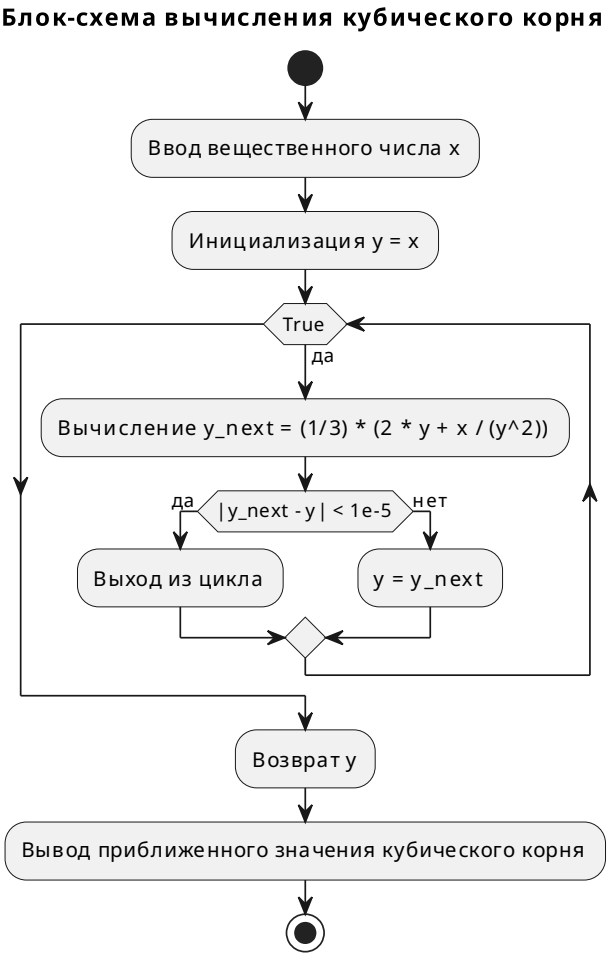
Задача 10:

По заданному вещественному x вычислить значение $\sqrt{x}$ 3 по следующей итерационной формуле: yi+1 =13(2yi + xyi2). Начальное приближение: y0 =x. Итерации прекратить при |yi+1 −yi|<10^−5

Листинг программы 10:

```python
def cube_root(x):
    y = x

    while True:
        y_next = (1 / 3) * (2 * y + x / (y**2))
        if abs(y_next - y) < 1e-5:
            break
        y = y_next
    return y


x = float(input("Введите вещественное число x: "))
result = cube_root(x)
print(f"Приближенное значение кубического корня из {x} равно {result:.6f}")

```

Блок схема программы 10:



Результат работы программы 10:

```
Введите вещественное число x: 3.2
Приближенное значение кубического корня из 3.2 равно 1.473613
```

Вывод: в ходе работы были сформированы практические навыки процедурного программирования, разработки и отладки программ, были освоены методы и средства разработки и оформления технической документации.