



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. ИУК4-33Б _____ (____ Сароян Н.А.____)
(Подпись) (Ф.И.О.)

Проверил: _____ (____ Осипова О. В.____)
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2024 г.

Целью выполнения лабораторной работы является формирование практических навыков процедурного программирования, разработки и отладки программ, овладение методами и средствами разработки и оформления технической документации.

Основными задачами выполнения лабораторной работы являются:

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

Вариант 5

Задача 1. Напечатать таблицу истинности логической функции

$$(\overline{A \vee B \& C}) \vee A,$$

где $\&$, \vee , \neg — знаки логических операций И, ИЛИ, НЕ.

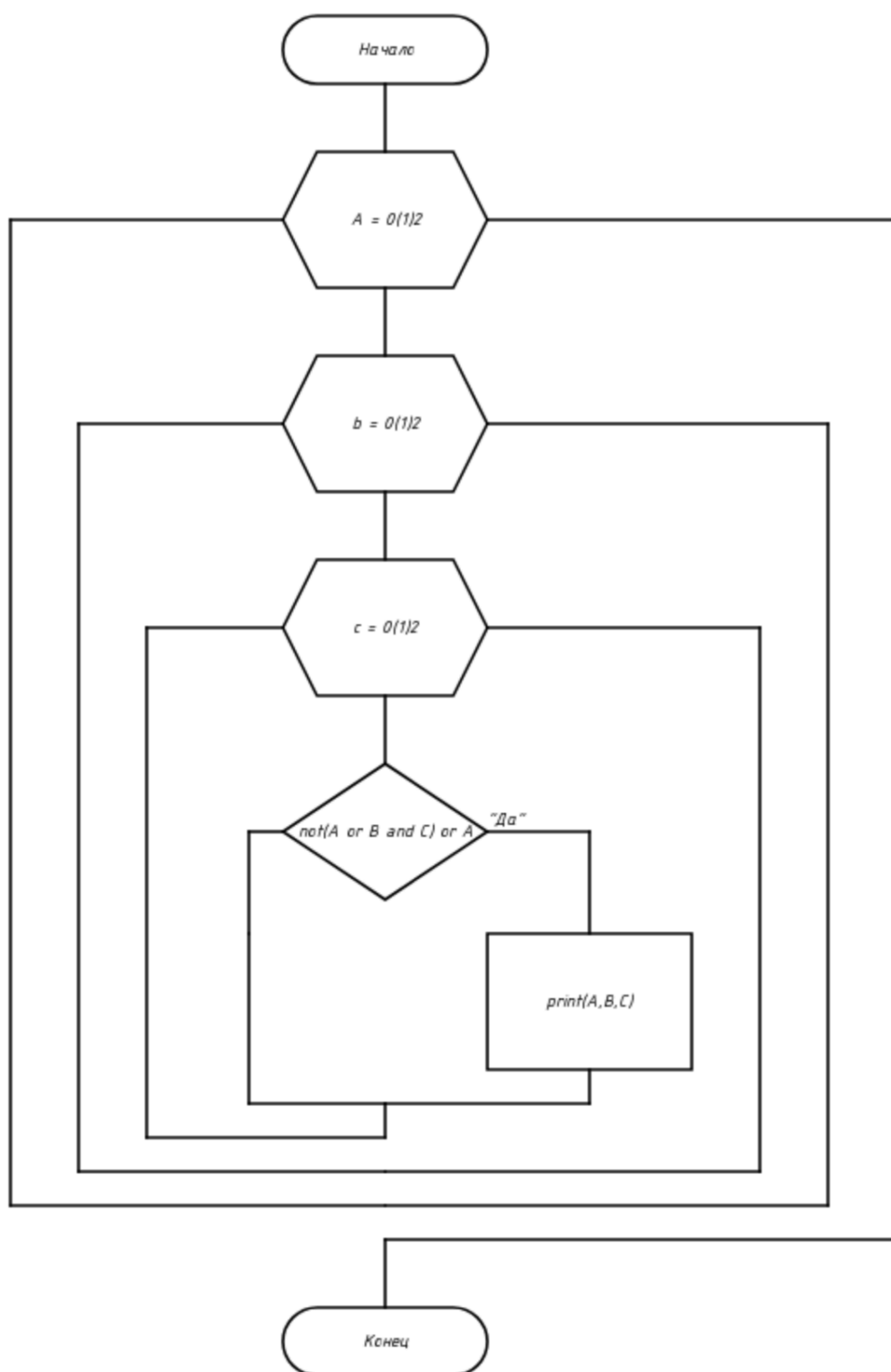
Листинг программы:

```
def task1():
    for A in range(2):
        for B in range(2):
            for C in range(2):
                if not (A or B and C) or A:
                    print(A, B, C, end="\n")
```

Результат выполнения программы:

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

Блок-схема алгоритма:



Задача 2. Дан массив $X(178)$. Вычислить разность между максимальным и минимальным по модулю элементами этого массива.

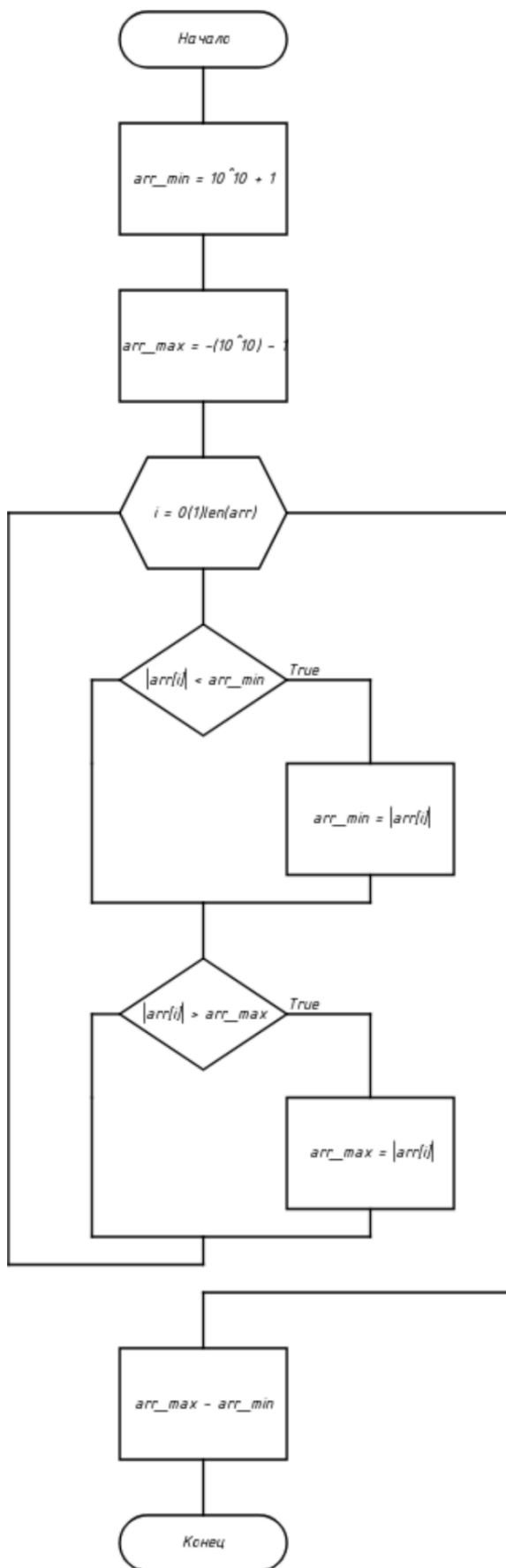
Листинг программы:

```
def task2(arr: list[int]):  
    arr_min = 10 ** 10 + 1  
    arr_max = -(10 ** 10) - 1  
    for i in range(0, len(arr)):  
        if abs(arr[i]) < arr_min:  
            arr_min = abs(arr[i])  
        if abs(arr[i]) > arr_max:  
            arr_max = abs(arr[i])  
    return arr_max - arr_min
```

Результат выполнения программы:

977317568

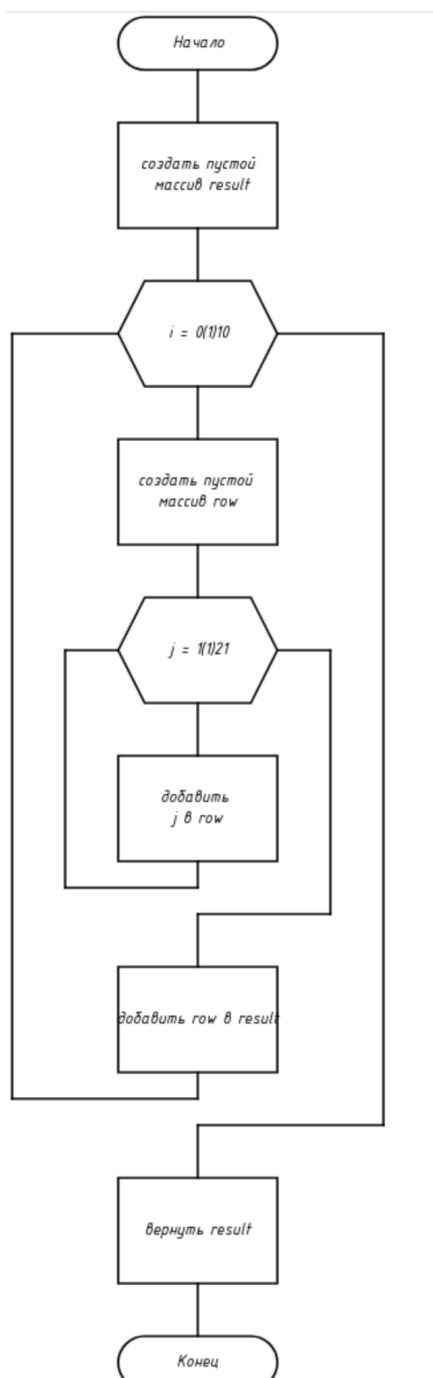
Блок-схема алгоритма:



Задача 3. Сформировать прямоугольную матрицу $A(10, 20)$ следующего вида:

$$\begin{pmatrix} 1 & 2 & . & . & . & 20 \\ 1 & 2 & . & . & . & 20 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 1 & 2 & . & . & . & 20 \end{pmatrix}$$

Блок-схема алгоритма:



Листинг программы:

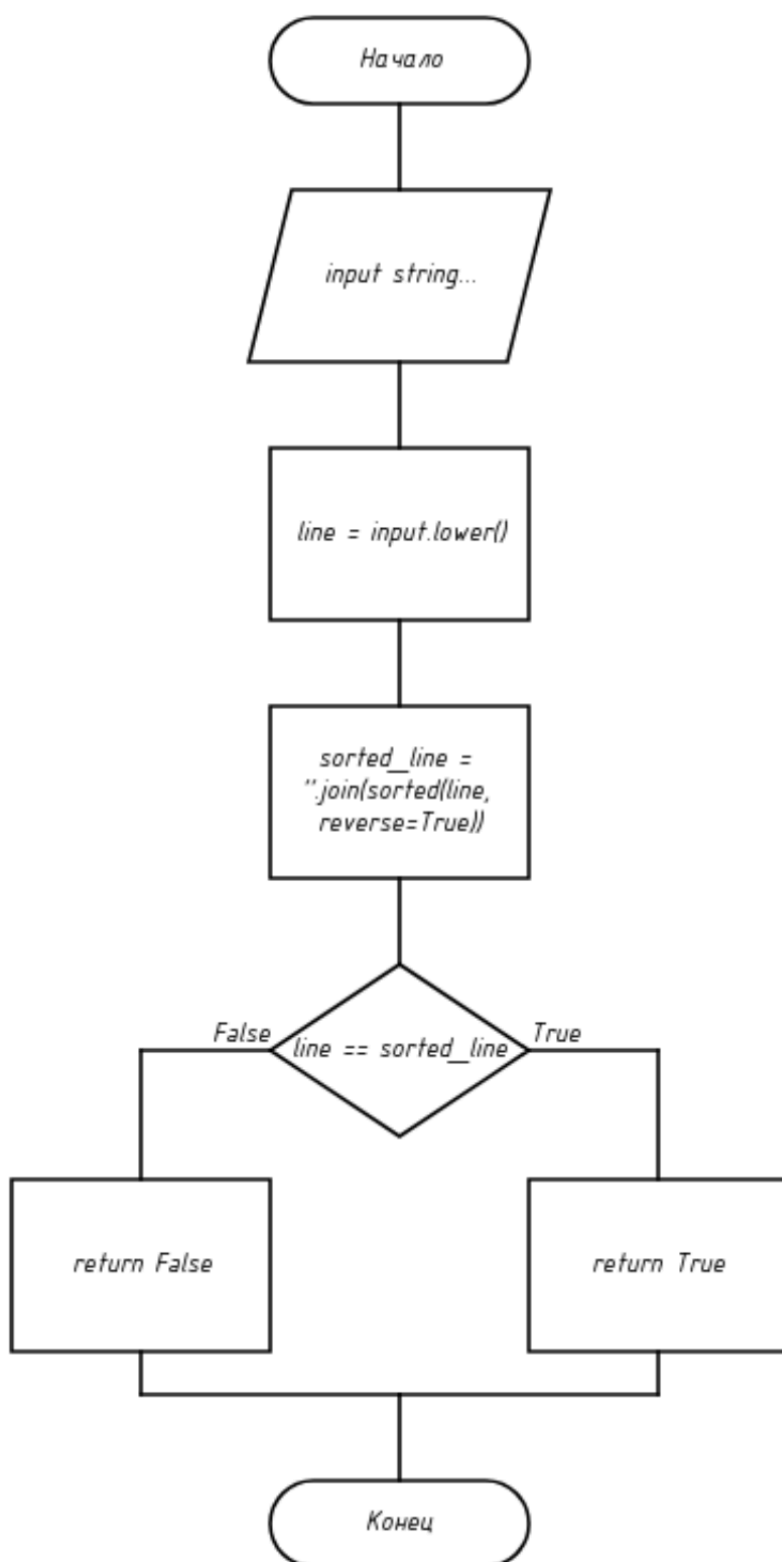
```
def task3():  
    return [[j for j in range(1, 21)] for i in range(10)]
```

Результат выполнения программы:

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

Задача 4. Ввести строку и определить, располагаются ли буквы в ней в порядке, обратном алфавитному.

Блок-схема алгоритма:



Листинг программы:

```
def task4() -> bool:  
    line = input("input string...").lower()  
    return line == ''.join(sorted(line, reverse=True))
```

Результат выполнения программы:

```
input string...edcba  
True
```

Задача 5. По заданному вещественному x вычислить значение $\sqrt[3]{x}$ по следующей итерационной формуле:

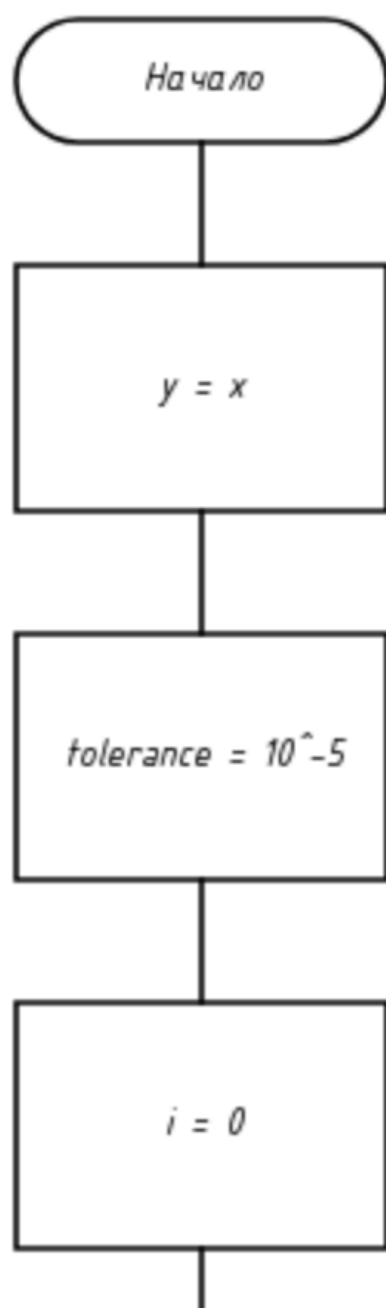
$$y_{i+1} = \frac{1}{3} \left(2y_i + \frac{x}{y_i^2} \right).$$

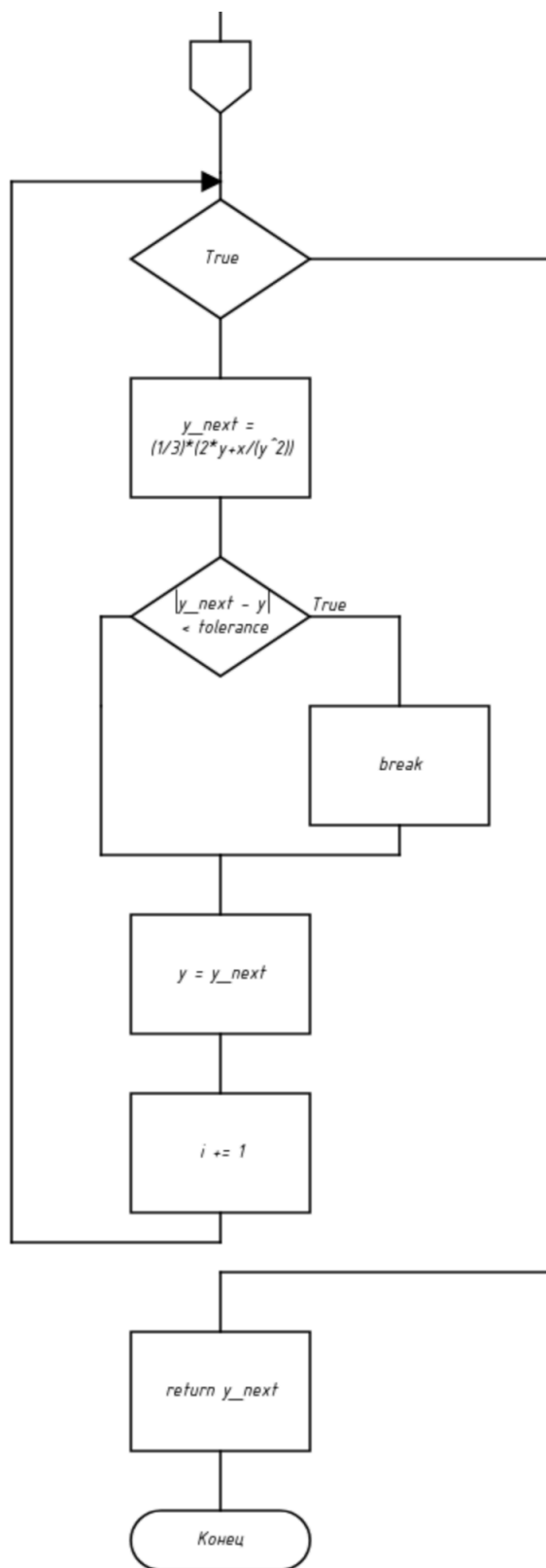
Начальное приближение: $y_0 = x$.

Итерации прекратить при

$$|y_{i+1} - y_i| < 10^{-5}$$

Блок-схема алгоритма:





Листинг программы:

```
def task5(x: int):  
    y = x  
    tolerance = 10 ** -5  
    i = 0  
    while True:  
        y_next = (1 / 3) * (2 * y + x / (y ** 2))  
        if abs(y_next - y) < tolerance:  
            break  
        y = y_next  
        i += 1  
    return y_next
```

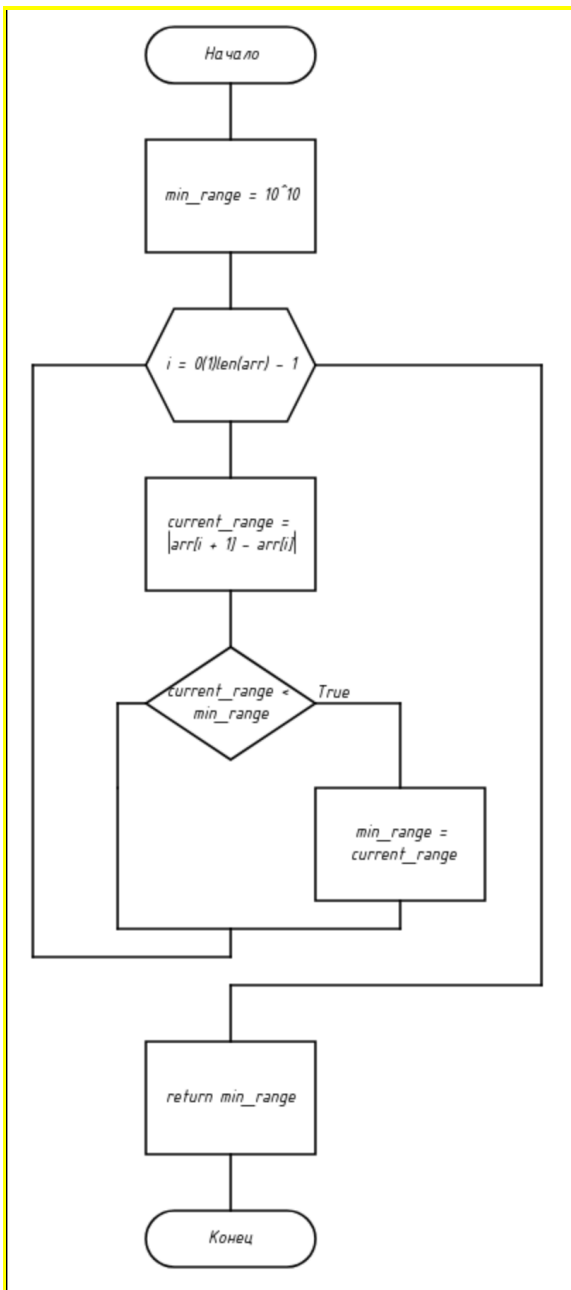
Результат выполнения программы:

3.0

Задача 6. Дан массив $A(N)$. Найти пару соседних элементов, наиболее близко расположенных друг к другу. Мера близости:

$$R = |A[i + 1] - A[i]|$$

Блок-схема алгоритма:



Листинг программы:

```

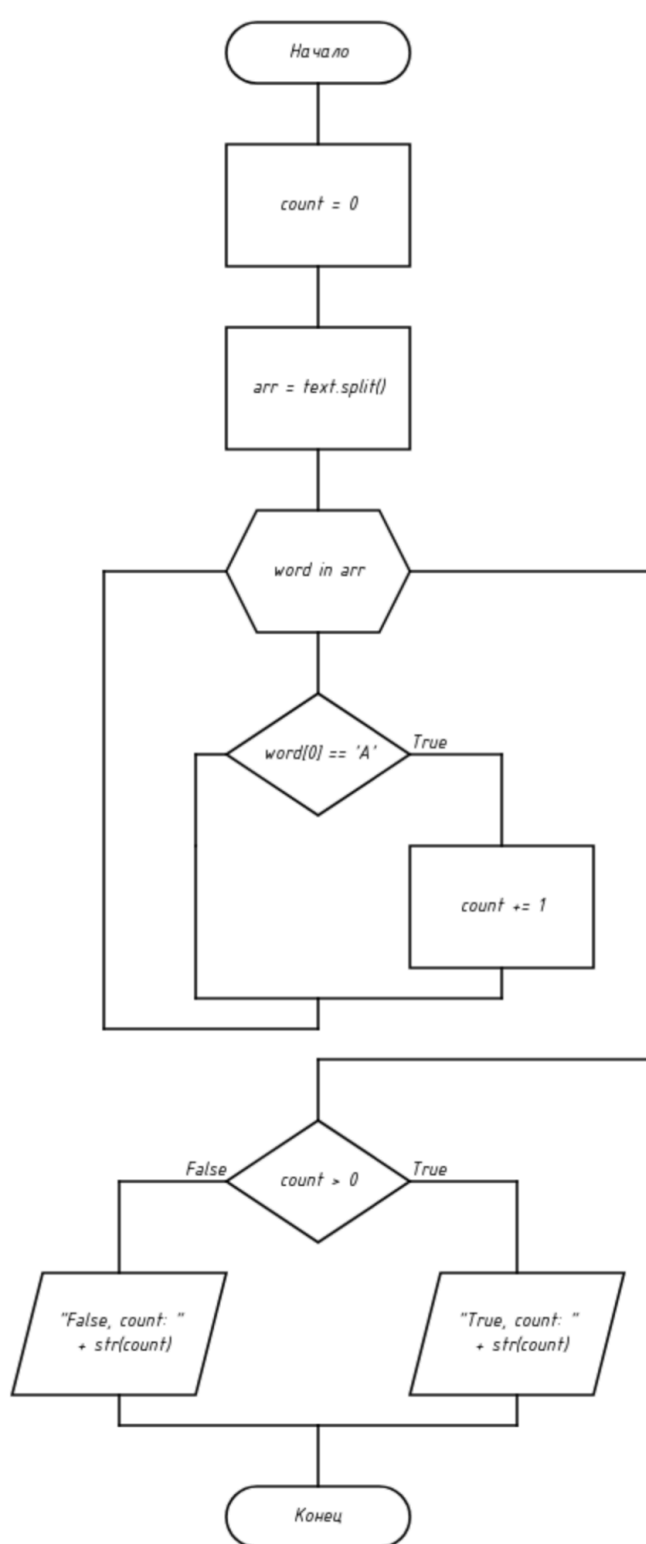
def task6(arr: list[int]):
    min_range = 10 ** 10
    for i in range(0, len(arr) - 1):
        current_range = abs(arr[i + 1] - arr[i])
        if current_range < min_range:
            min_range = current_range
    return min_range
  
```

Результат выполнения программы:

19

Задача 7. Выяснить, есть ли во введенном тексте слова, начинающиеся с буквы А, и сколько таких слов.

Блок-схема алгоритма:



Листинг программы:

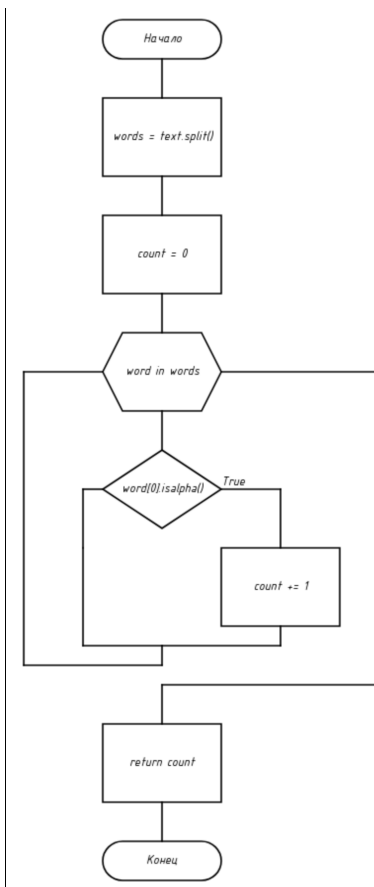
```
def task7(text: str):
    count = 0
    arr = text.split()
    for word in arr:
        if word[0] == 'A':
            count += 1
    if count > 0:
        print("True, count: " + str(count))
    else:
        print("False, count: " + str(count))
```

Результат программы:

```
True, count: 3
```

Задача 8. Во введенном тексте подсчитать количество слов, считая словом последовательность букв и цифр, начинающуюся с буквы (слова разделены пробелами).

Блок-схема алгоритма:



Листинг программы:

```
def task8(text: str):  
    words = text.split()  
    return sum(1 for word in words if word[0].isalpha())
```

Результат выполнения программы:



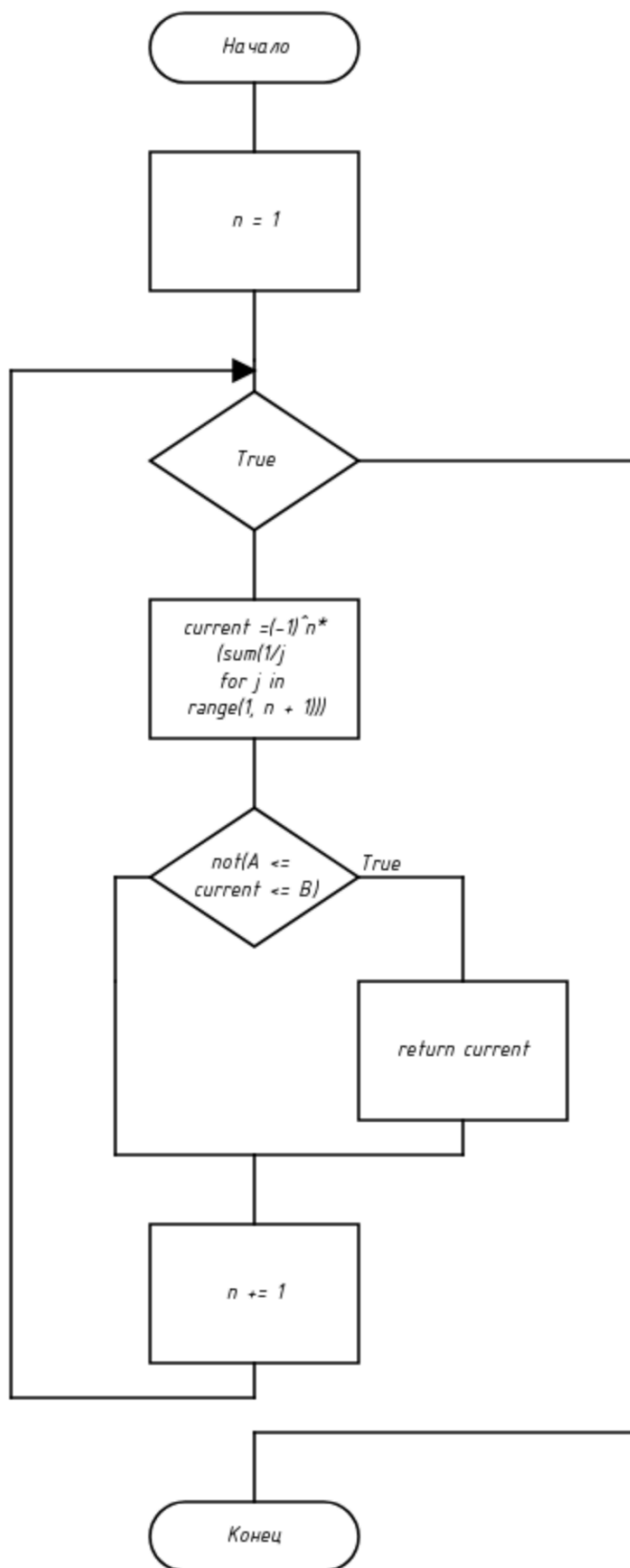
11

Задача 9. Даны вещественные числа A и B ($A < B$). Найти первый член последовательности

$$a_n = (-1)^n \left(1 + \frac{1}{2} + \dots + \frac{1}{n} \right), n = 1, 2, \dots,$$

который не принадлежит $[A, B]$.

Блок-схема алгоритма:



Листинг программы:

```
def task9(A, B: int):
    n = 1
    while True:
```

```

current = (-1) ** n * (sum(1 / j for j in range(1, n + 1)))
if not (A <= current <= B):
    return current
n += 1

```

Результат выполнения программы:

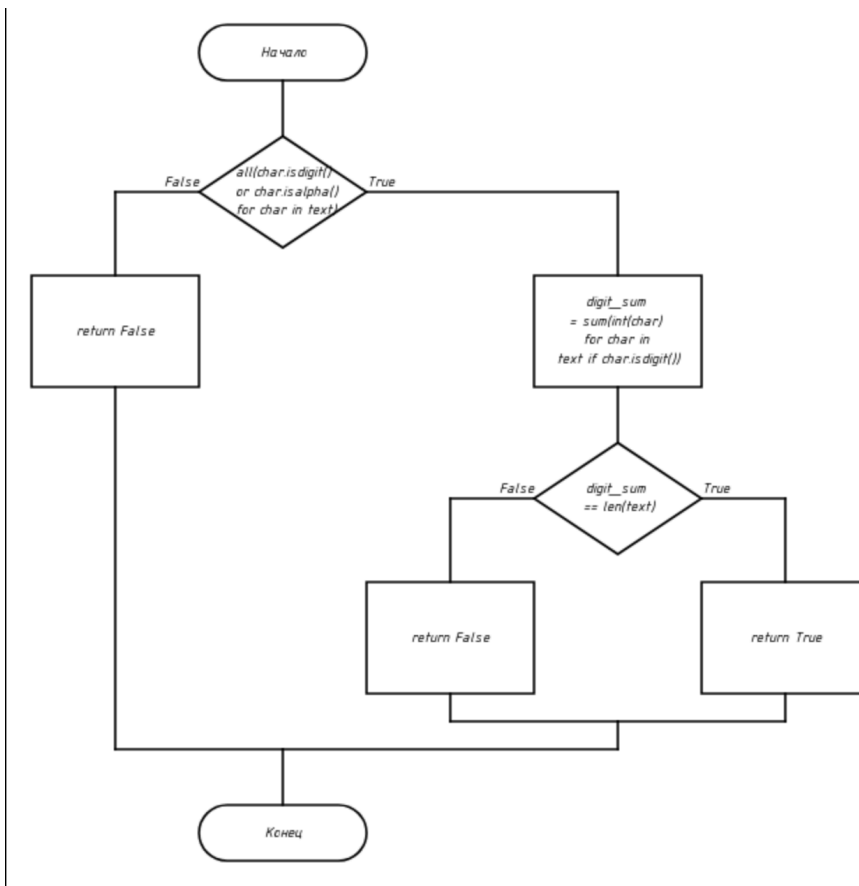
```

input: 1, 100
-1.0

```

Задача 10. Ввести текст, состоящий только из цифр и букв. Выяснить, верно ли, что сумма числовых значений цифр, находящихся в тексте, равна длине текста.

Блок-схема алгоритма:



Листинг программы:

```
def task10(text: str):  
    if all(char.isdigit() or char.isalpha() for char in text):  
        return sum(int(char) for char in text if char.isdigit()) == len(text)  
    return False
```

Результат выполнения программы:

```
input: a1b2c3d4e5aaaaa  
True
```

Выводы:

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