**Московский государственный технический**

**университет им. Н. Э. Баумана.**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Основы программирования»

Отчет по лабораторной работе № 3

«Программирование циклических алгоритмов.

Расчет по формулам»

Вариант 8

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-12Б |  | преподаватель каф. ИУ5 |
| Поляков Леонид |  | Правдина А. Н. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

**Постановка задачи**

**Указания по выполнению работы**

Для повторения или завершения выполнения программы используйте цикл do ... while, который должен включать в себя запрос “Продолжить работу? (y/n)” и ввод с клавиатуры соответствующего символа. Это позволит запускать программу с новыми данными, не завершая ее. Используйте такой прием в тех случаях, когда требуется многократно запускать программу с различными исходными данными (например, для отладки или демонстрации работы преподавателю).

***Задача 1***.

* Найдите сумму натуральных чисел, которые делятся на 5 и не делятся на m (m<n). Количество натуральных чисел *n* и значение m введите с клавиатуры.

***Задача 2***.

* Составьте программу для вычисления:



Значение *а* введите с клавиатуры.

Объясните результат: при *а = 5.7 S =* 147 450. Обеспечьте нужную точность представления результата, используя манипулятор ***setprecision*** для потокового вывода.

***Задача 3***.

Составьте программу вычисления значения суммы 

и функции *,* где 0≤ *х* ≤1, с шагом *h=0.2.* Вычисление суммы ряда Тейлора производите с погрешностью, не превышающей 0.000001.

Результат представить в виде таблицы (без рамок), которая содержит четыре строки со значениями *x*, *Y(x), S(x)* *и N,* где *N* - номер последнего слагаемого.

При вычислении значения очередного члена ряда используйте значение предыдущего члена, для чего следует вручную получить соотношение вида

k(x,n) = A.i (x,n) / A.i-1(x,n);

Вычислении членов ряда, начиная со второго (а может и с третьего) следует выполнять по формуле:

Ai = Ai-1\* k;

Для представления результата в виде таблицы используйте манипулятор ***setw.***

Близость значений *S(x)* и *Y(x)* (отличие должно быть меньше 0.000001)во всем диапазоне значений *х* указывает на правильность их вычисления.

***Задача 4***.

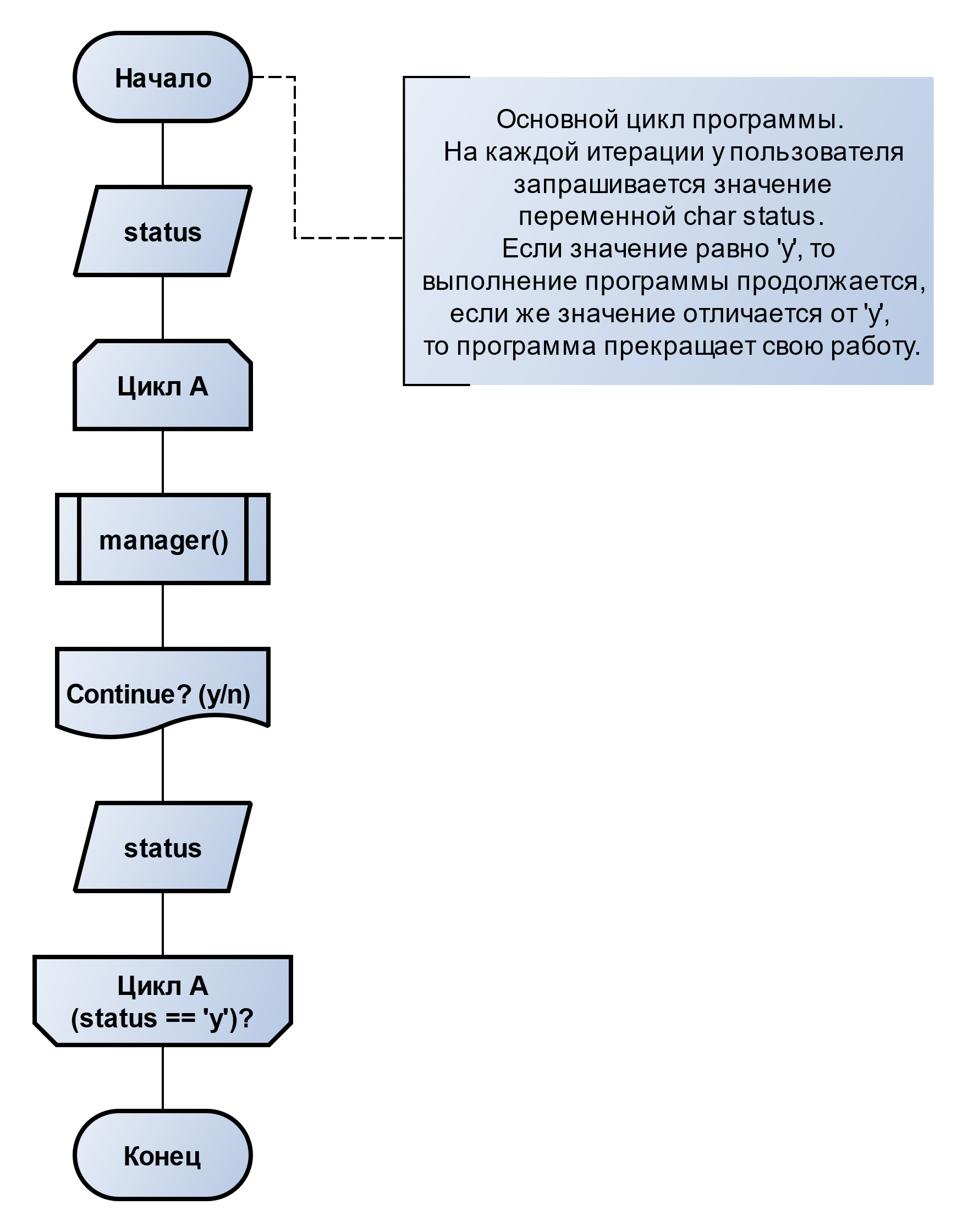
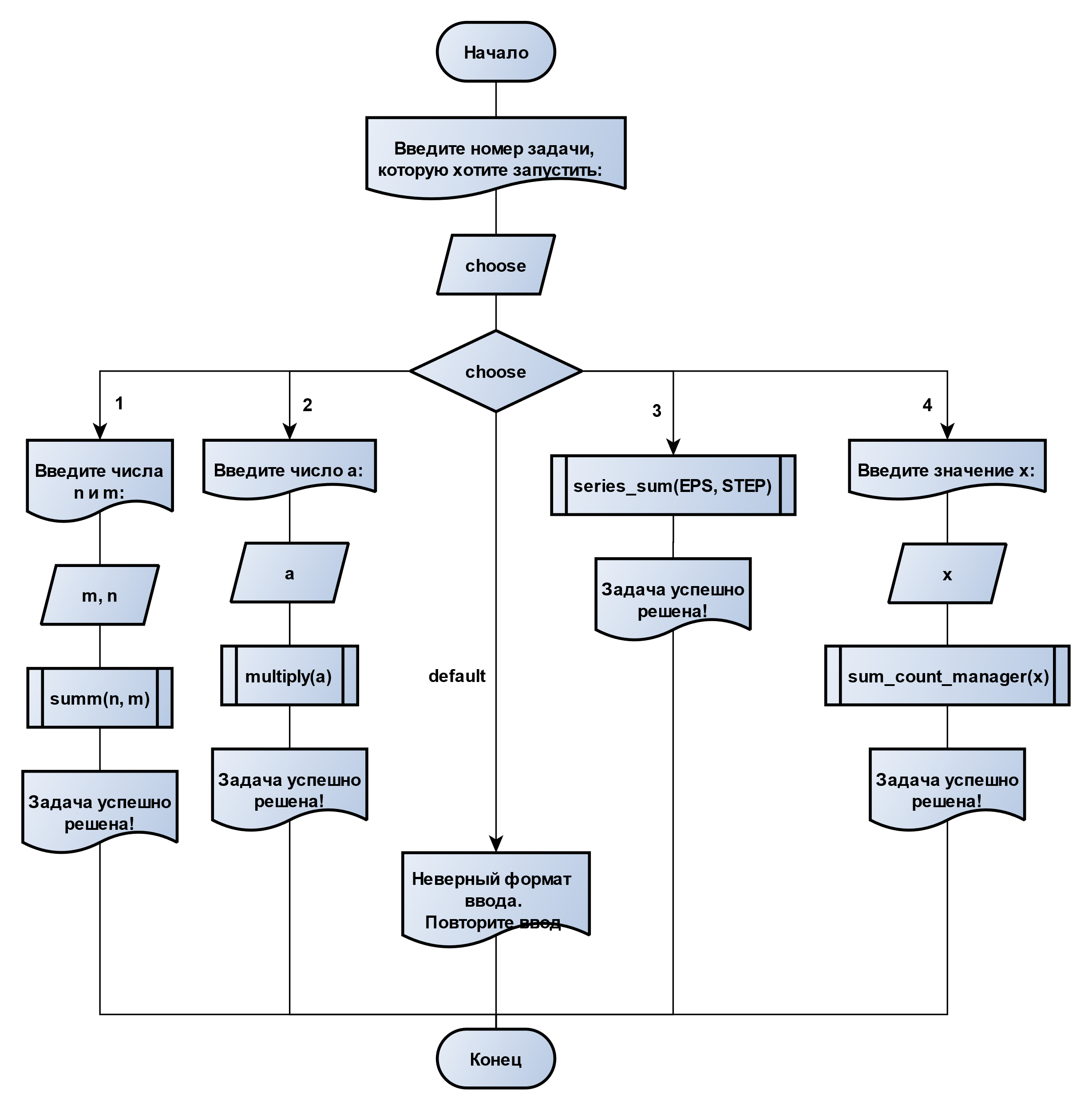
Напишите программу для вычисления *у* по формуле: .

Натуральное значение *n* введите с клавиатуры. Значения *x* также введите с клавиатуры.

Обеспечьте возможность, не завершая программу, вычислить *y* для нескольких значений *n* и выведите на экран значения промежуточных (частичных) сумм при количестве слагаемых 3, 5 и 10.

**Разработка алгоритма**

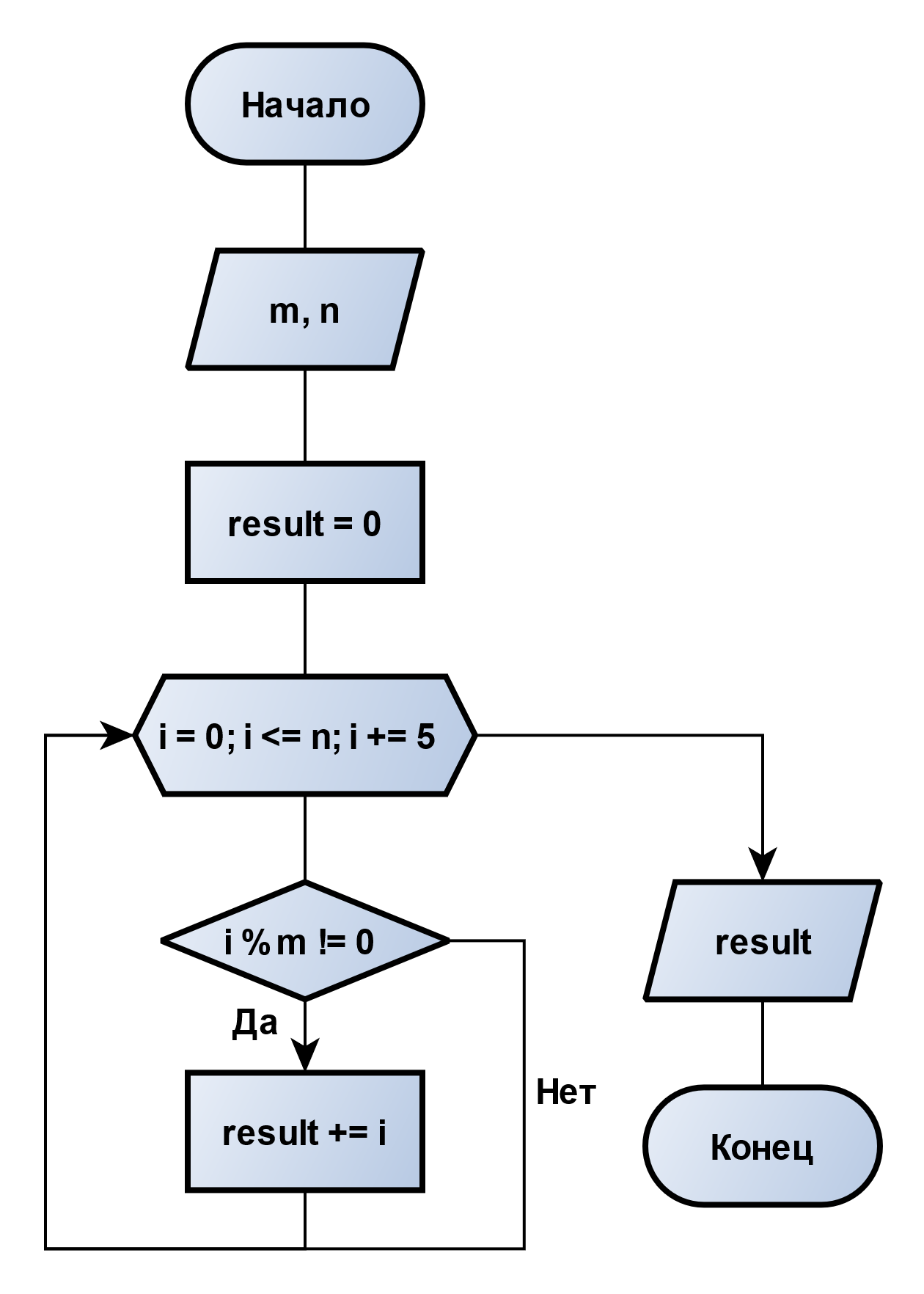
Программа представляет собой вечный цикл, на каждой итерации которого программа спрашивает пользователя: продолжить выполнение задач (‘y’) или завершить свою работу (‘n’). В случае положительного ответа пользователю на выбор предлагается решить 4 задачи, описанные в ТЗ. Пользователь вводит порядковый номер задачи, после чего запускается соответствующая этой задаче функция, после выполнения которой пользователь опять вправе решить: завершить выполнение программы или продолжить.

**Основной цикл программыФункция manager()**

**Функция** int summ(int n, int m)

Находит сумму натуральных чисел, которые делятся на 5 и не делятся на m (m<n)

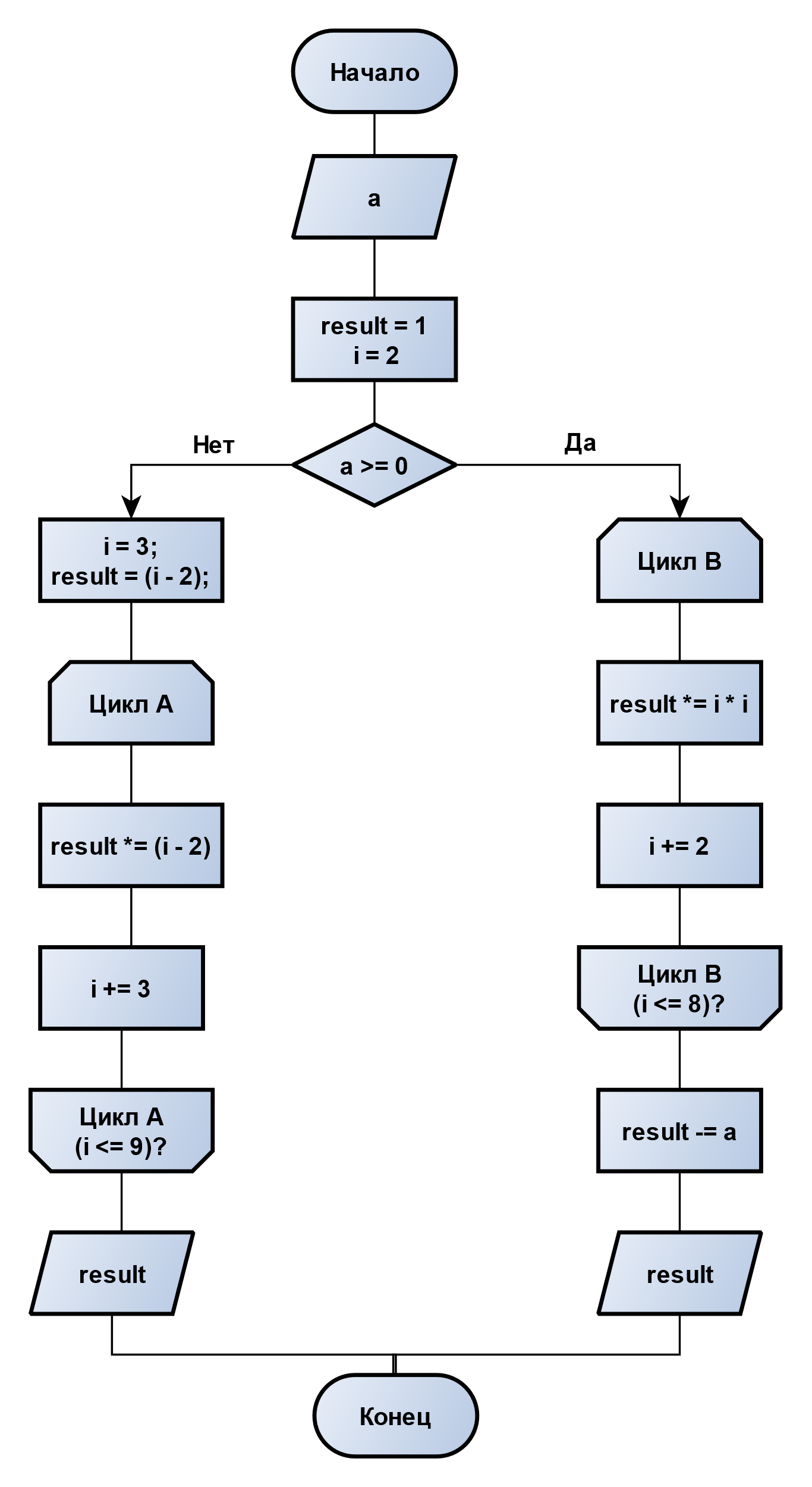
Переменные m, n вводятся с клавиатуры.



**Функция** double multiply(double a)

Принимает на вход параметр a. Считает произведение по формуле 

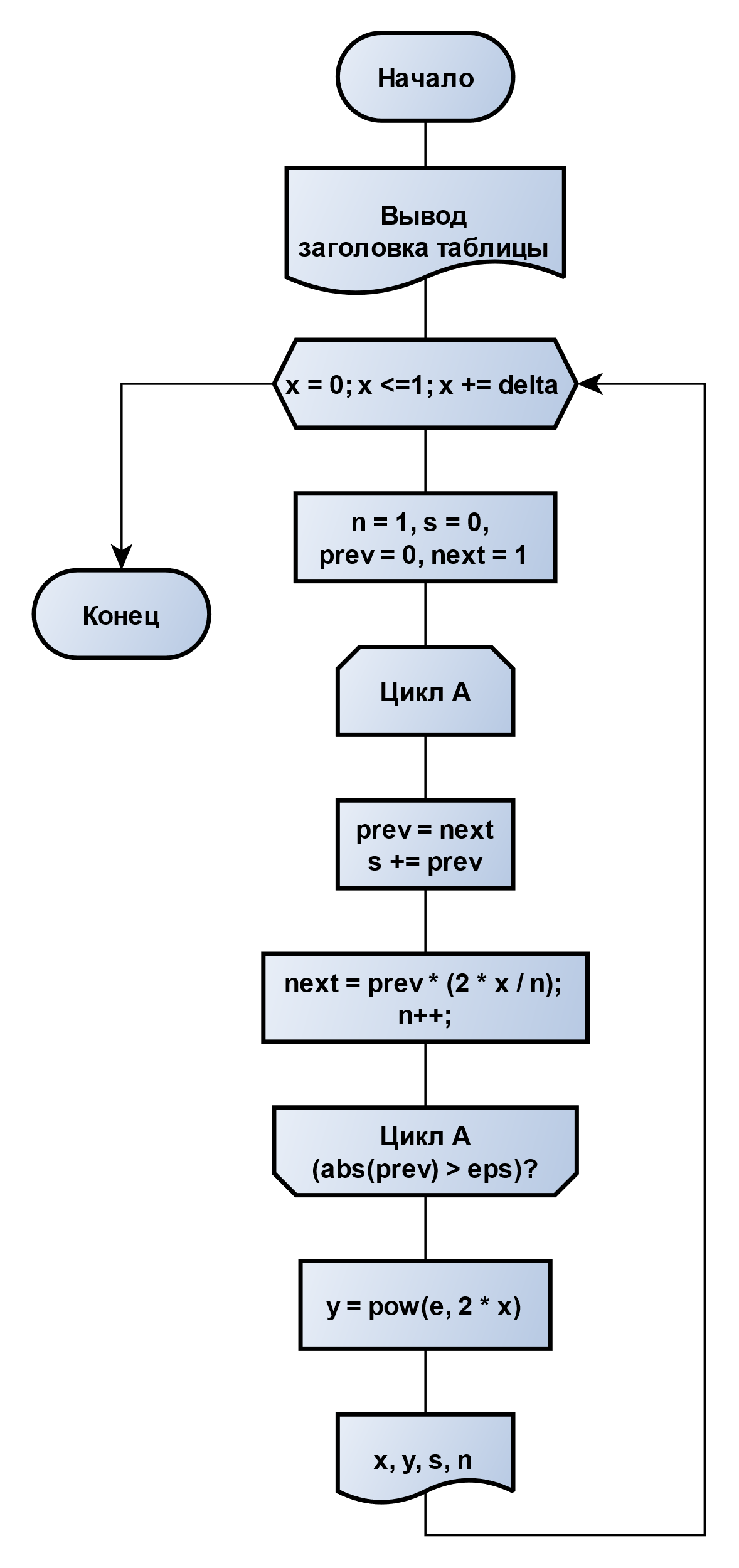
Переменная i соответствует переменной i в формуле, переменная result хранит в себе результат вычислений



**Функция** void series\_sum(double eps, double delta = constants::STEP)

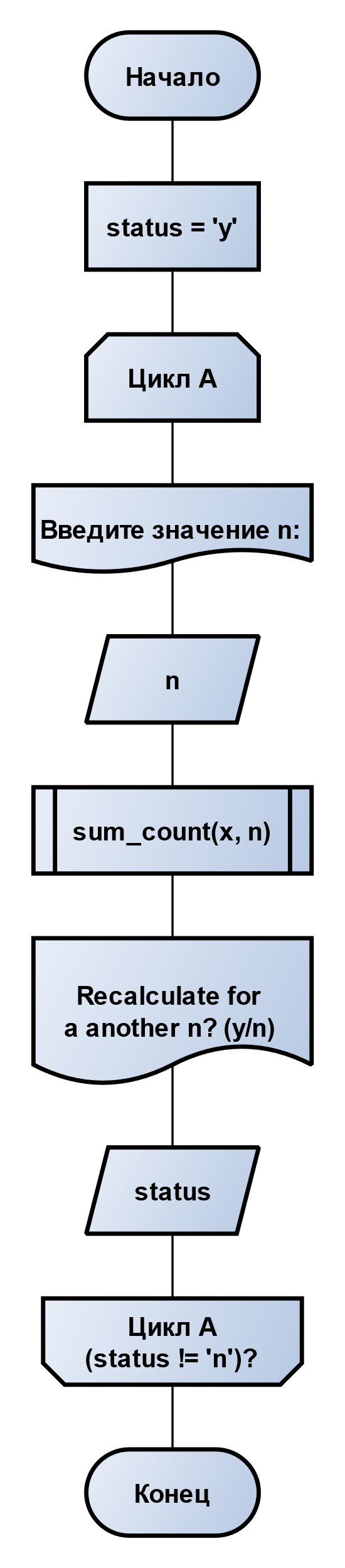
Функция, вычисляющая значение функции e^2x для разных x от 0 до 1 с шагом delta.

Первый способ заключается в итеративном подсчёте, основанном на последовательном вычислении элементов ряда тейлора.

Второй способ считает значение напрямую, используя математическую константу "e" из заголовочного файла <numbers>. Выводит результат в виде таблицы.

Параметр eps отвечает за точность вычислений, параметр delta обозначает величину изменения переменной x на каждой итерации и по умолчанию равен 0.2. Переменная n хранит в себе номер текущего слагаемого ряда; переменная s хранит в себе текущую сумму ряда; переменные prev и next хранят в себе на каждой итерации предыдущее и новое значение слагаемого в ряду соответственно; в переменную y помещается результат вычисления функции e^2x, при помощи встроенной функции для вычисления степени pow()

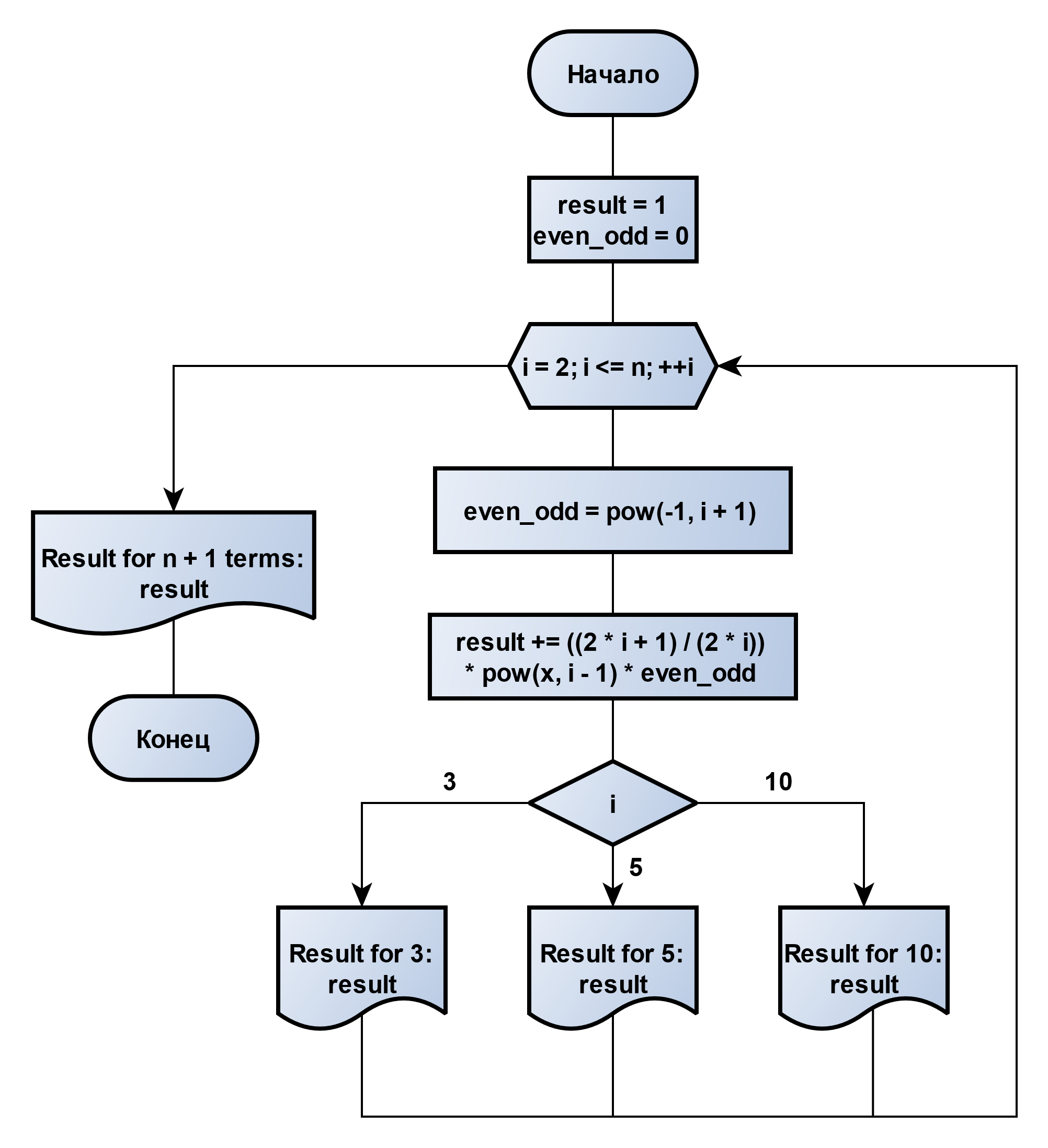
**Функция** void sum\_count\_manager(double x, double n)

Данная функция позволяет считать выражение в задаче номер 4 для разных значений переменной n. Функция запрашивает значение n у пользователя с клавиатуры, а затем вызывает функцию sum\_count(x, n), которая считает сумму ряда из задачи номер 4. После каждого успешного выполнения задачи номер 4, функция спрашивает у пользователя: желает ли он продолжить и посчитать выражение для другого значения n, или же он желает закончить. Ответ пользователя (y/n) хранится в переменной status.

**Функция** void sum\_count(double x, double n)

Данная функция считает выражение 

Параметры x и n задаются пользователем с клавиатуры. Переменная result хранит в себе сумму на текущей итерации цикла, переменная even\_odd отвечает за знак каждого слагаемого в зависимости от того, чётный или нечётный у него порядковый номер. Также функция выводит промежуточные результаты для промежуточных (частичных) сумм при количестве слагаемых 3, 5 и 10.

****

**Текст Программы**

**constants.h**

**Изображение выглядит как текст, линия, Шрифт, число

Автоматически созданное описание**

**functions.h**

**Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, число

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, число

Автоматически созданное описание**

**main.cppИзображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Анализ результатовИзображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Результат работы программы соответствует ожиданиям и полностью удовлетворяет требованиям ТЗ**

**Использованные источники**

* **Веб-сайт кафедры ИУ5, курс «основы программирования»**

[**https://cpp1.wiki.iu5edu.ru**](https://cpp1.wiki.iu5edu.ru)

* **Веб-сайт кафедры ИУ5 «автоматизированные системы обработки информации и управления»**

**https://e-learning.bmstu.ru/iu5/mod/folder/view.php?id=277**