

ПОНЯТИЕ АЛГОРИТМА И ЕГО СВОЙСТВА

Понятие алгоритма и его свойства

Алгоритм – упорядоченная совокупность системы правил, определяющая содержание и порядок действий над некоторыми объектами, строгое выполнение которых приводит к решению любой задачи из рассматриваемого класса задач за конечное число шагов.

Слово «*алгоритм*» появилось в IX веке, когда европейцы познакомились с работами выдающегося узбекского математика Муххамеда бен аль-Хорезми («*algorithmi*» – латинская форма записи имени аль-Хорезми). В своей книге «Об индийском счете» он сформулировал правила выполнения арифметических действий над десятичными числами. В дальнейшем понятие «алгоритм» стали использовать для обозначения любой последовательности действий, приводящей к решению поставленной задачи.

При всем разнообразии алгоритмов можно выделить общие для них *свойства: дискретность, массовость, определенность и результативность.*

Дискретность (разрывность) – это свойство алгоритма, характеризующее его структуру: каждый алгоритм состоит из отдельных законченных действий [4, с. 292].

Массовость – применимость алгоритма ко всем задачам рассматриваемого типа, при любых исходных данных [4, с. 292].

Определенность – свойство алгоритма, указывающее на то, что каждый шаг алгоритма должен быть строго определен и не допускать различных толкований; также строго должен быть определен порядок выполнения отдельных шагов [4, с. 292].

Результативность – конечность действий алгоритма решения задач, позволяющая получить желаемый результат при допустимых исходных данных за конечное число шагов.

Способы описания алгоритмов

Разработанный алгоритм можно записать несколькими способами:

- на естественном языке (словесное описание);
- с помощью псевдокода;
- в виде блок-схемы;
- в виде программы.

Словесное описание представляет структуру алгоритма на естественном языке. Примерами такой записи алгоритма являются инструкции по эксплуатации приборов бытовой техники, кулинарные рецепты, правила дорожного движения и т.д.

Словесная форма имеет ряд недостатков:

- строго не формализуема;
- страдает многословностью записей;
- допускает неоднозначность толкования отдельных предписаний.

Эта форма обычно используется на начальных стадиях разработки алгоритма.

Псевдокод – пошагово-словесная запись алгоритма по определенным правилам или соглашениям. В псевдокоде используется общепринятая математическая символика и конструкции. Он занимает промежуточное место между естественным и формальным языками.

Рассмотрим алгоритм умножения двух чисел, записанный с помощью псевдокода.

1. Ввод a, b .
2. $P = a \cdot b$.
3. Вывод P .
4. Конец.

Примером псевдокода является школьный алгоритмический язык (АЯ).

Основные служебные слова этого языка представлены в табл. 1.

Служебные слова школьного алгоритмического языка

алг (алгоритм)	сим (символьный)	цел (целый)	для	вывод	ввод
арг (аргумент)	лит (литерный)	кц (конец цикла)	от	нет	или
рез (результат)	лог (логический)	дано	до	при	пока
нач (начало)	вещ (вещественный)	надо	и	выбор	иначе
кон (конец)	нц (начало цикла)	все	если	то	знач

Пример записи алгоритма на школьном АЯ:

алг Сумма квадратов (**арг цел** n , **рез цел** S)

дано | $n > 0$

надо | $S = 1*1 + 2*2 + 3*3 + \dots + n*n$

нач цел i

ввод n ; $S := 0$

нц для i **от** 1 **до** n

$S := S + i*i$

кц



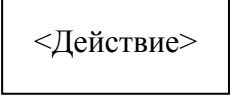
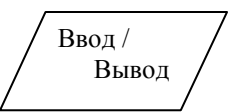




вывод "S = ", S

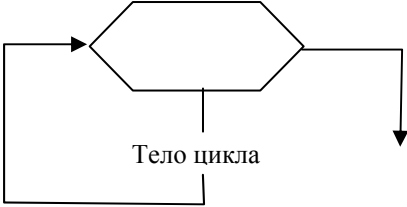




кон

Блок-схема – это наглядное графическое представление алгоритма с помощью геометрических фигур, соединенных линиями-связями, показывающими порядок выполнения инструкций.

Рассмотрим основные фигуры, которые используются для построения блок-схем (табл. 2).

Графические объекты блок-схем

Вид графического объекта	Название блока. Комментарии
	Начало алгоритма
	Конец алгоритма
	Процесс. Внутри блока записывается действие, вычислительная операция или группа
	Ввод/вывод данных с неопределенного носителя. Надпись внутри блока: ввод (вывод) и список вводимых (выводимых) переменных
	Ручной ввод (ввод с клавиатуры)
	Дисплей (вывод на монитор)
	Документ (вывод на печатающее устройство)
	Решение (условный блок). Условие записывается внутри блока. В результате проверки условия осуществляется выбор одного из возможных путей вычислительного процесса

1	2
 <p>Тело цикла</p>	<p>Модификатор. Используется для описания цикла с параметром</p>
 <p>Тело цикла</p>	<p>Границы цикла. Описывает циклические процессы: «цикл с предусловием» и «цикл с постусловием»</p>
	<p>Внутристраничный соединитель</p>
	<p>Межстраничный соединитель</p>
	<p>Предопределенный процесс (подпрограмма)</p>

Программа – описание структуры алгоритма на языке программирования.

ОСНОВНЫЕ АЛГОРИТМИЧЕСКИЕ КОНСТРУКЦИИ

Существует три типа алгоритмических конструкций: *линейная* (последовательная), *разветвляющаяся* и *циклическая*.

Линейная алгоритмическая конструкция

Линейный алгоритм – это такой алгоритм, в котором действия выполняются однократно в заданном порядке (рис. 1).

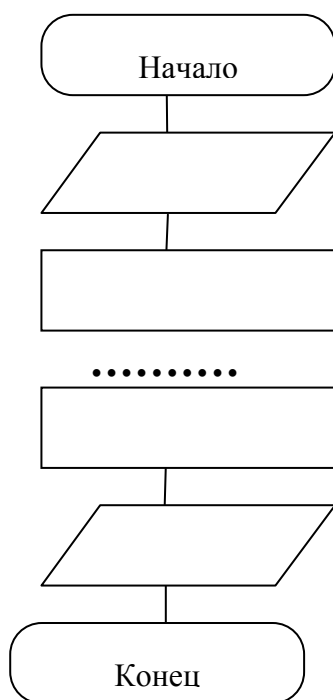


Рис. 1. Размещение блоков в линейном алгоритме

ПРИМЕР 1. Зная радиус основания и высоту цилиндра, вычислите его объем. Разработайте алгоритм решения задачи: составьте псевдокод и блок-схему.

Псевдокод:

1. Ввод R и H.
2. $V = \pi \cdot R^2 \cdot H$.
3. Вывод V.
4. Конец.

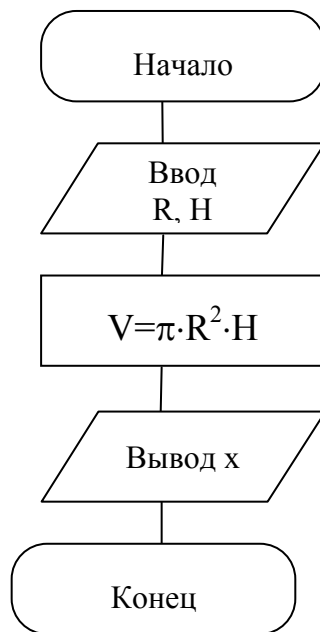


Рис. 2. Блок-схема к примеру 1

Разветвляющаяся алгоритмическая конструкция

Разветвляющийся алгоритм – такой алгоритм, в котором выполняется либо одна, либо другая последовательность действий, в зависимости от условия.

Различают две формы разветвляющейся алгоритмической структуры: *полное ветвление* (ЕСЛИ – ТО – ИНАЧЕ) и *неполное ветвление* (ЕСЛИ – ТО).

Полное ветвление (рис. 3) позволяет организовать в алгоритме две ветви (ТО или ИНАЧЕ). Неполное ветвление (рис. 4) предполагает наличие действий только на одной ветви (ТО), вторая ветвь отсутствует.

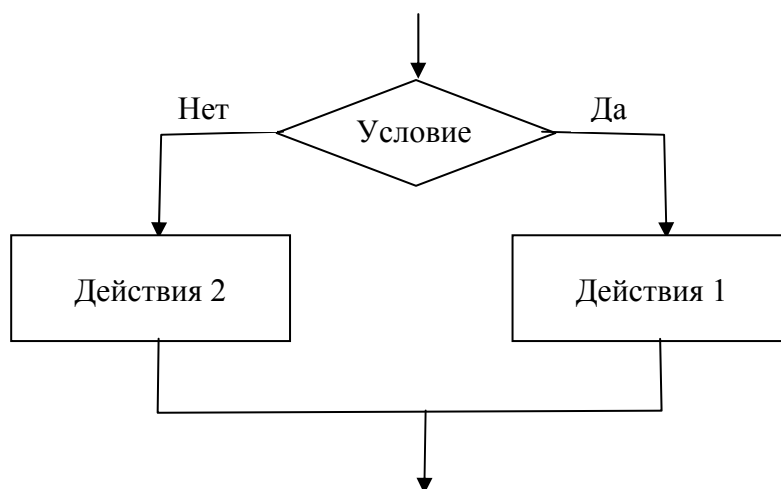


Рис. 3. Полное ветвление

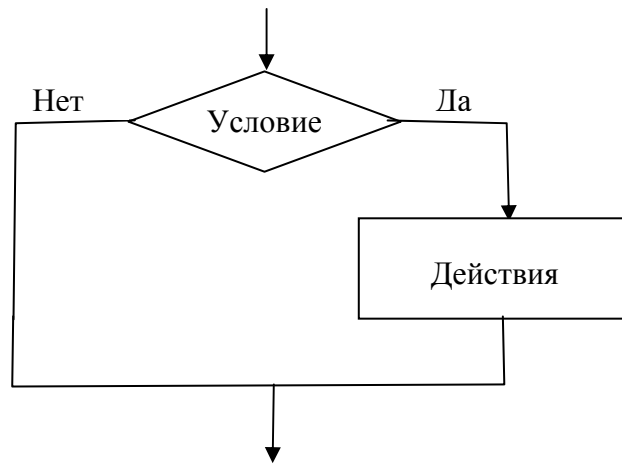


Рис. 4 Неполное ветвление

ПРИМЕР 2. Составить псевдокод и блок-схему к алгоритму вычисления значения функции

$$Y(X) = \begin{cases} -1, & x \leq 0 \\ x^3, & x > 0. \end{cases}$$

Псевдокод:

1. Ввод X.
2. ЕСЛИ $X > 0$, ТО $Y(X) = X^3$,
ИНАЧЕ $Y(X) = -1$.
3. Вывод $Y(X)$.
4. Конец.

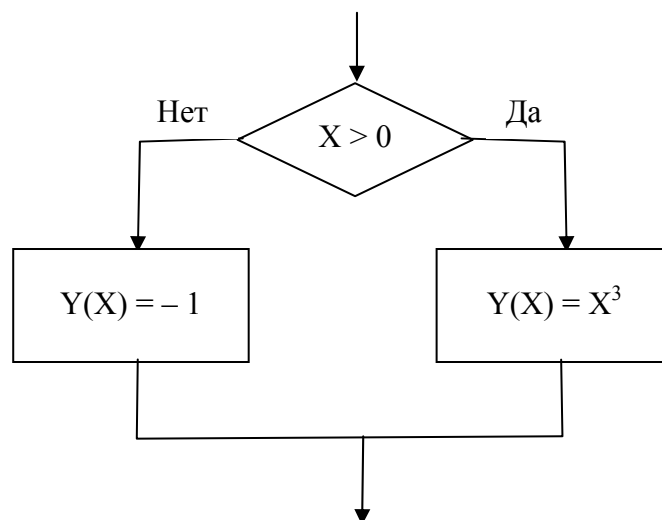


Рис. 5. Фрагмент блок-схемы к примеру 2

ПРИМЕР 3. Известны коэффициенты a , b , c квадратного уравнения.

Вычислите корни квадратного уравнения. Составьте псевдокод и блок-схему.

Псевдокод:

1. Ввод a , b , c .
2. $d = b^2 - 4 \cdot a \cdot c$.
3. ЕСЛИ $d < 0$, ТО «Корней нет», перейти к п.5. ИНАЧЕ

3.1. $X_1 = \frac{-b + \sqrt{d}}{2 \cdot a}$,

$X_2 = \frac{-b - \sqrt{d}}{2 \cdot a}$.

- 3.2. Вывод X_1 и X_2 .

4. Конец.

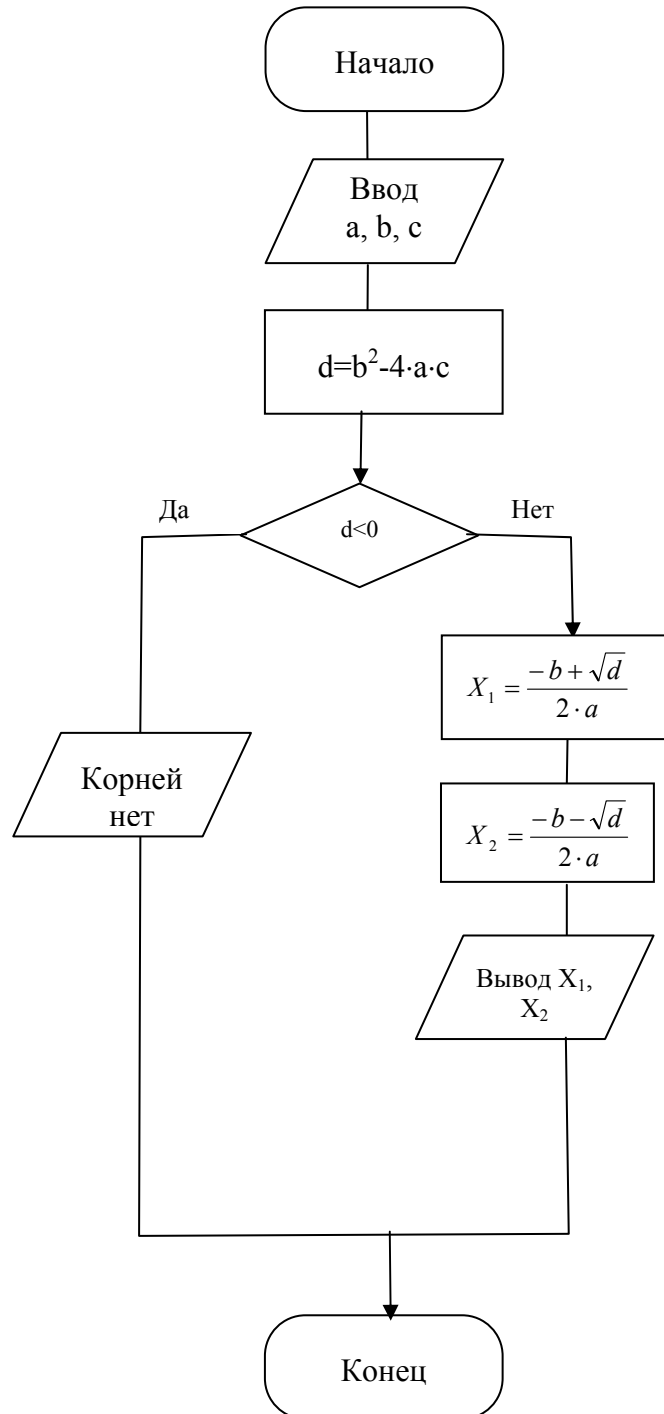


Рис. 6. Блок-схема алгоритма решения квадратного уравнения

Алгоритмическая структура «выбор»

Для выбора из нескольких альтернативных действий используется алгоритмическая структура «выбор». Перед выполнением команды «выбор» вычисляется значение некоторого выражения X , а затем начинается проверка условий $Y_1(X)$, $Y_2(X) \dots Y_n(X)$. Проверка продолжается до тех пор, пока не встретится условие, принимающее значение ИСТИНА при данном X (рис. 7).

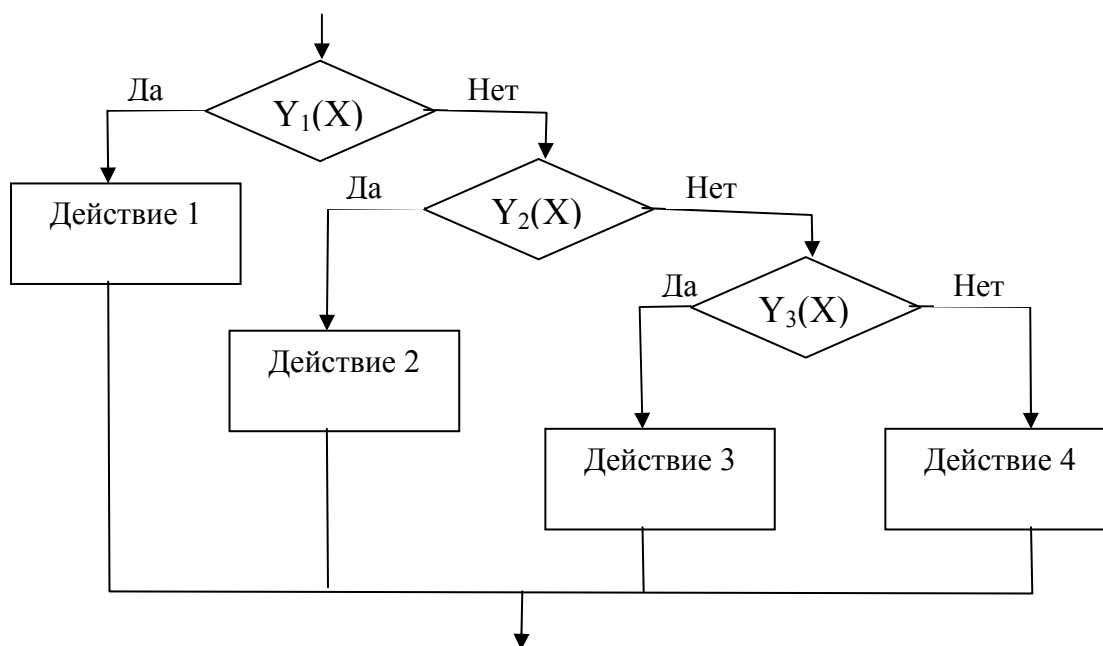


Рис. 7. Алгоритмическая структура «выбор»

Структуру «выбор» можно также изобразить в форме, представленной на рис. 8.

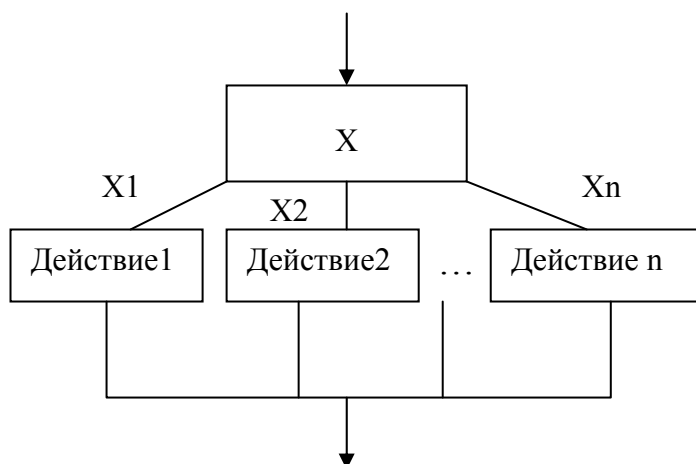


Рис. 8. Вариант оформления алгоритмической структуры «выбор»

ПРИМЕР 4. Составить блок-схему к программе, которая запрашивает у пользователя номер дня недели и выводит одно из сообщений: «рабочий день», «суббота» или «воскресенье».

Псевдокод:

1. Ввод X.
2. Выбор X :
 - 1.. 5: вывод «рабочий день»;
 - 6: вывод «суббота»;
 - 7: вывод «воскресенье»;ИНАЧЕ вывод «Ошибка! Введите число от 1 до 7».
3. Конец.

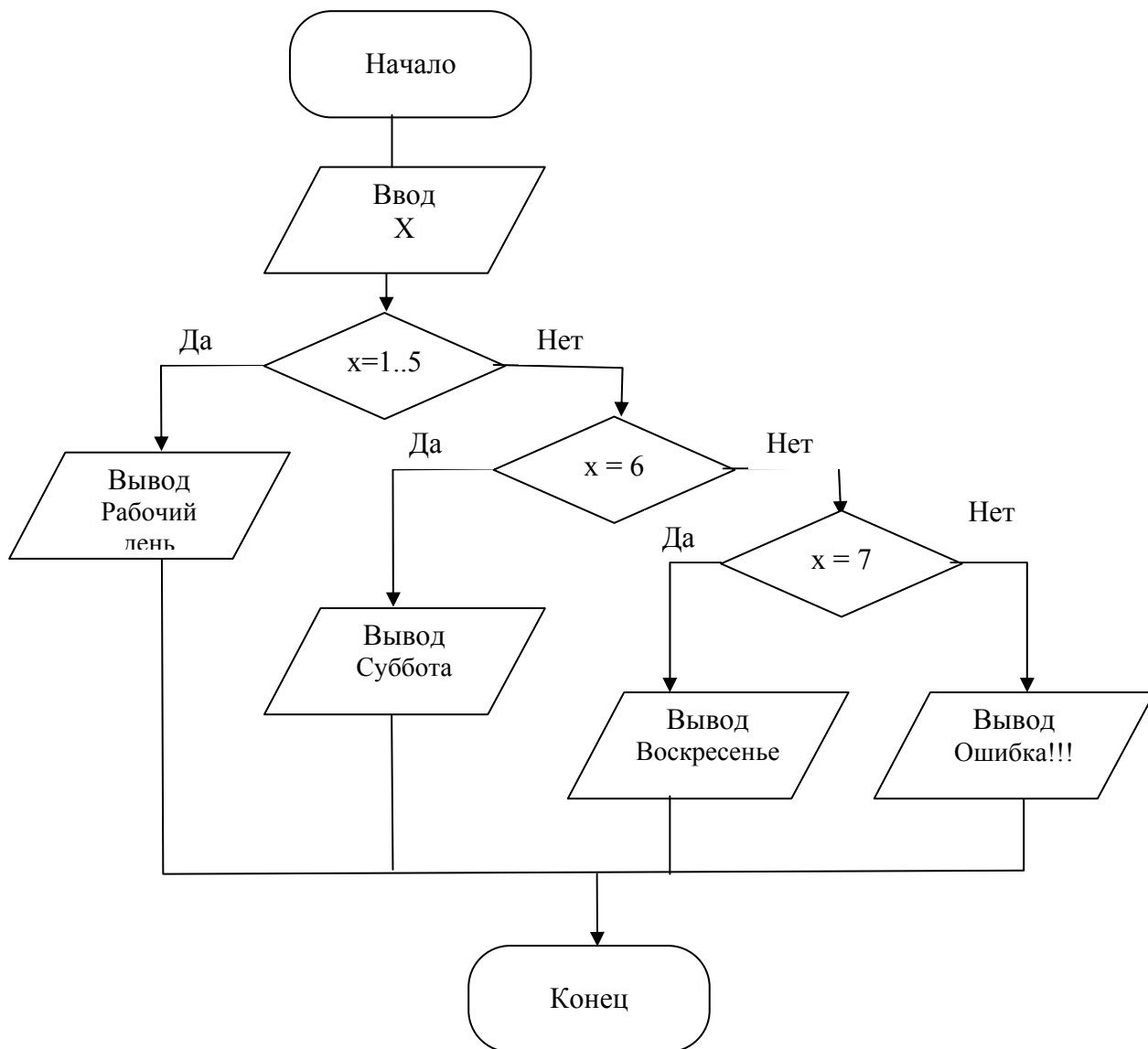


Рис. 9. Блок-схема к примеру 4 (Вариант 1)

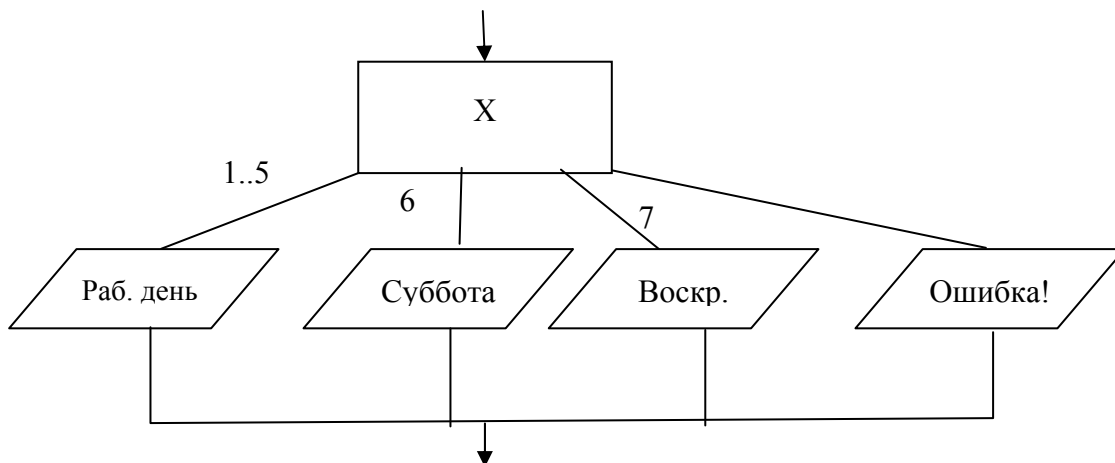


Рис. 10. Часть блок-схемы к примеру 4 (Вариант 2)

Циклическая алгоритмическая конструкция

Циклической называют алгоритмическую конструкцию, в которой действие выполняется указанное число раз, или - пока не выполнится условие.

Группа повторяющихся действий цикла называется *телом цикла*.

Существует три типа циклов: *цикл с параметром (арифметический)*, *цикл с предусловием* и *цикл с постусловием*.

Цикл с параметром

В цикле с параметром число повторений цикла однозначно определено и задается с помощью начального, конечного значений параметра и шагом его изменения.

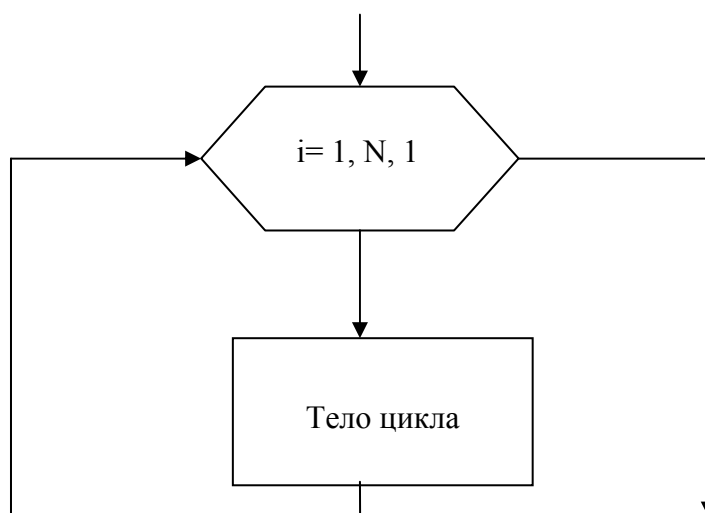


Рис. 11. Цикл с параметром

ПРИМЕР 5. Составьте блок-схему к следующей задаче: даны целые числа K и N ($N > 0$). Вывести N раз число K .

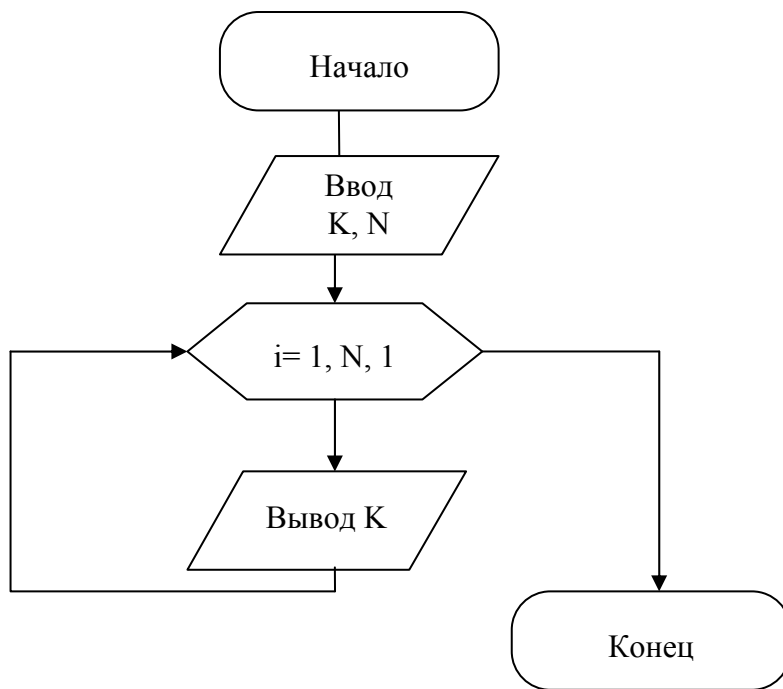


Рис. 12. Блок-схема к примеру 5

Цикл с предусловием

Действия внутри этого цикла повторяются, пока выполняется условие в блоке ветвления, причем сначала проверяется условие, а затем выполняется действие.

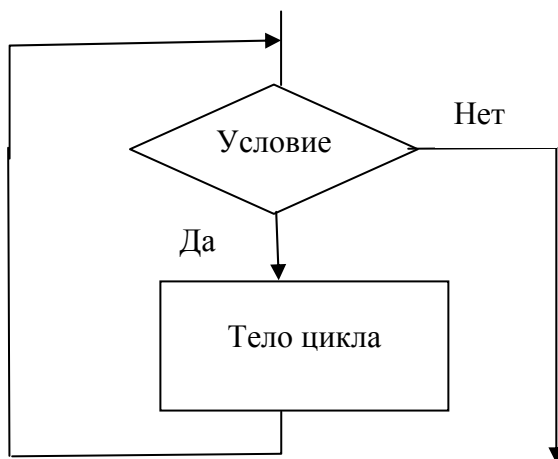


Рис. 13. Блок-схема цикла с предусловием, представленная с помощью условного блока



Рис. 14. Границы цикла

ПРИМЕР 6. Найти значение всех $Y = X^2 + 1$ при X , изменяющемся от 1 до 10 с шагом 0,5.

Псевдокод:

1. $X = 1$.
2. Пока $X > 10$ делать
 - 2.1. $Y = X^2 + 1$.
 - 2.2. Вывод X, Y .
 - 2.3. $X = X + 0,5$.
3. Конец.

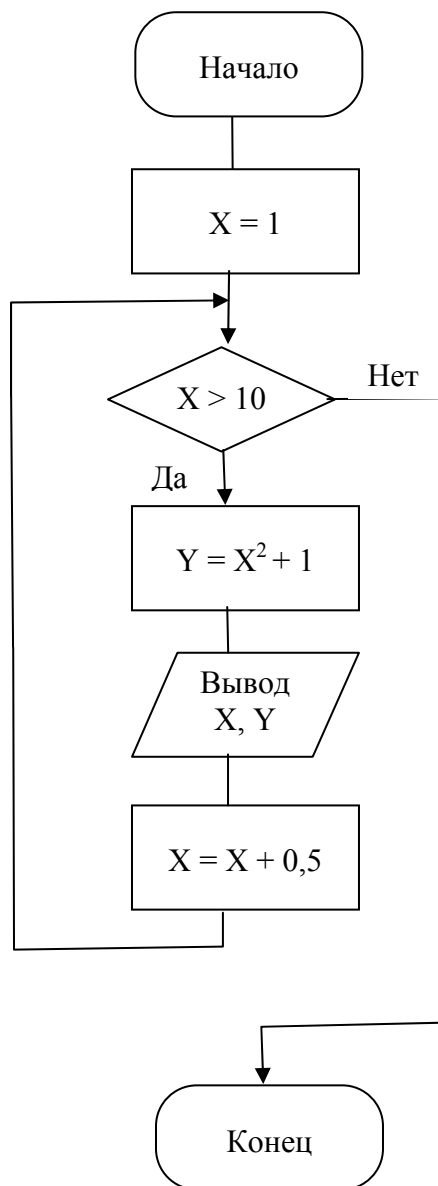


Рис. 15. Блок-схема к примеру 6

Цикл с постусловием

Тело цикла с постусловием всегда будет выполнено хотя бы один раз. Оно будет выполняться до тех пор, пока значение условного выражения ЛОЖНО. Как только условное выражение принимает значение ИСТИНА, цикл завершается.

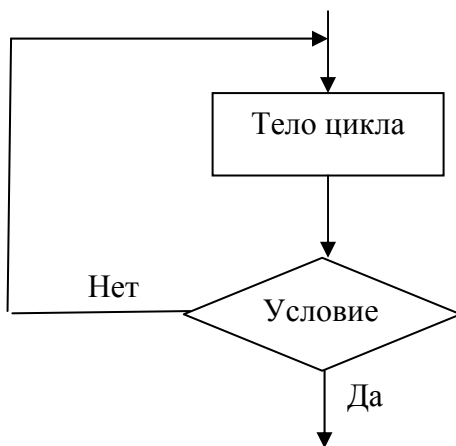


Рис. 16. Блок-схема цикла с постусловием, представленная с помощью условного блока

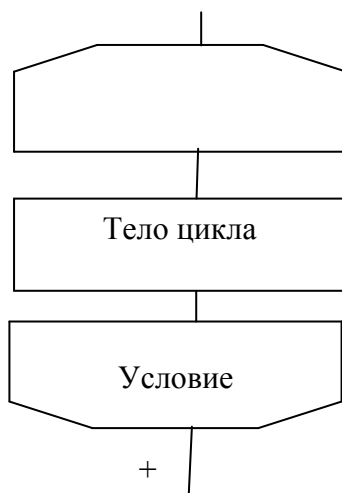


Рис. 17. Второй способ представления цикла с постусловием

ПРИМЕР 7. Составить блок-схему к программе, которая запрашивает у пользователя положительные числа, считает их сумму и количество. Как только введено отрицательное число или ноль, программа завершается.

В данной задаче используются стандартные циклические алгоритмы: вычисление суммы и подсчет количества элементов, удовлетворяющих некоторому признаку. Сформулируем правила разработки этих алгоритмов.

Правило суммирования:

1. Необходимо задать начальное значение суммы $S = 0$.
2. В теле циклической конструкции выполнить команду:
 $S = S + \langle \text{слагаемое} \rangle$.

Правило счетчика:

1. Начальное значение счетчика $K = 0$.
2. В теле цикла выполнить команду $K = K + 1$.

Псевдокод:

1. $S = 0, K = 0$.
2. ДЕЛАТЬ
 - 2.1. Ввод x .
 - 2.2. $S = S + x$.
 - 2.3. $K = K + 1$.
 - 2.4. ПОКА ложно $x \leq 0$.
3. $S = S - x, K = K - 1$.
4. ВЫВОД S, K
5. Конец.

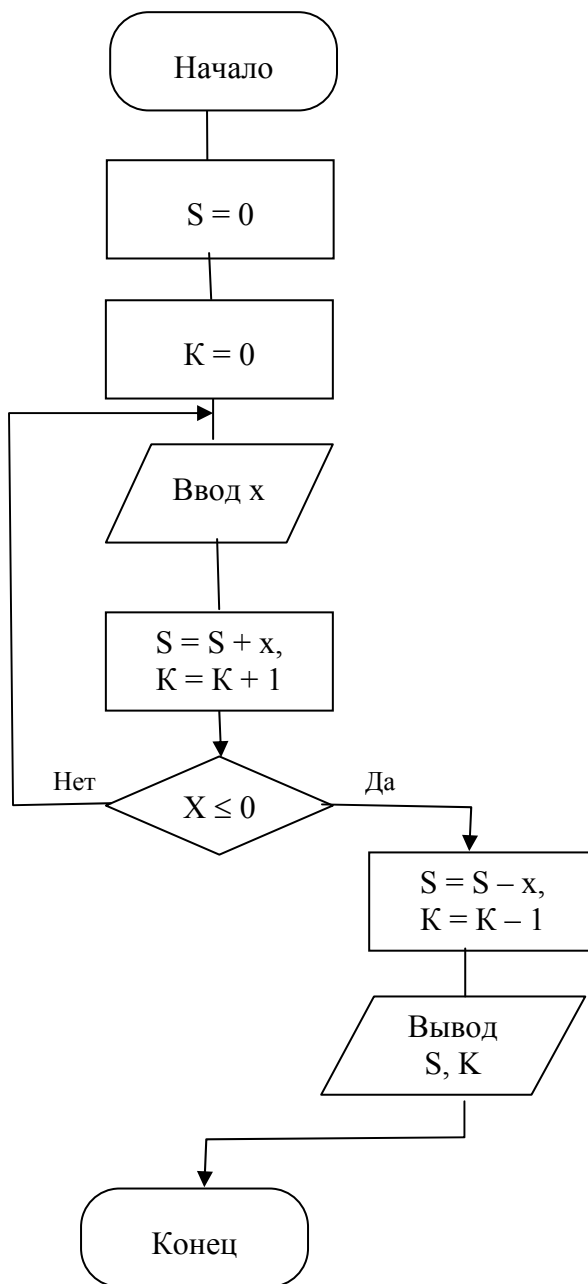


Рис. 18. Блок-схема к примеру 8

ОБРАБОТКА МАССИВОВ

Массив – это упорядоченная совокупность однотипных величин, имеющих общее имя, элементы которой различаются порядковыми номерами (индексами).

Одномерный массив

Одномерный массив – это массив, элементы которого имеют только один индекс.

Количество элементов массива называют его *размерностью*. Размерность массива записывается рядом с именем в круглых скобках, например $A(10)$.

Примерами одномерных массивов являются арифметические и геометрические прогрессии.

Ввод элементов массива осуществляется поэлементно, обычно в порядке возрастания индексов. Алгоритм ввода элементов одномерного массива представлен на блок-схеме (рис. 17).

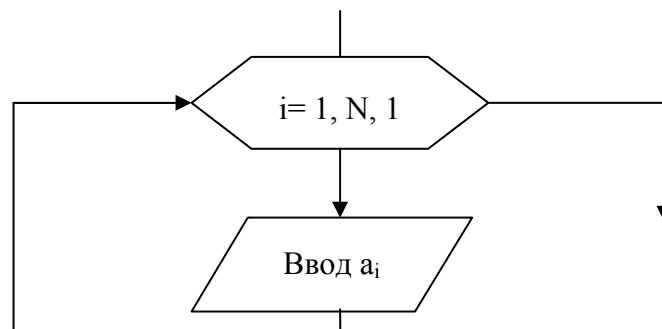


Рис. 19. Ввод элементов одномерного массива

При обработке элементов массива доступ к ним осуществляется также в цикле, как и ввод.

ПРИМЕР 8. Вычислить произведение элементов одномерного массива $A(10)$.

При решении данной задачи необходимо задать начальное значение для произведения $P = 1$, а в теле цикла выполнить команду $P = P \cdot a_i$.

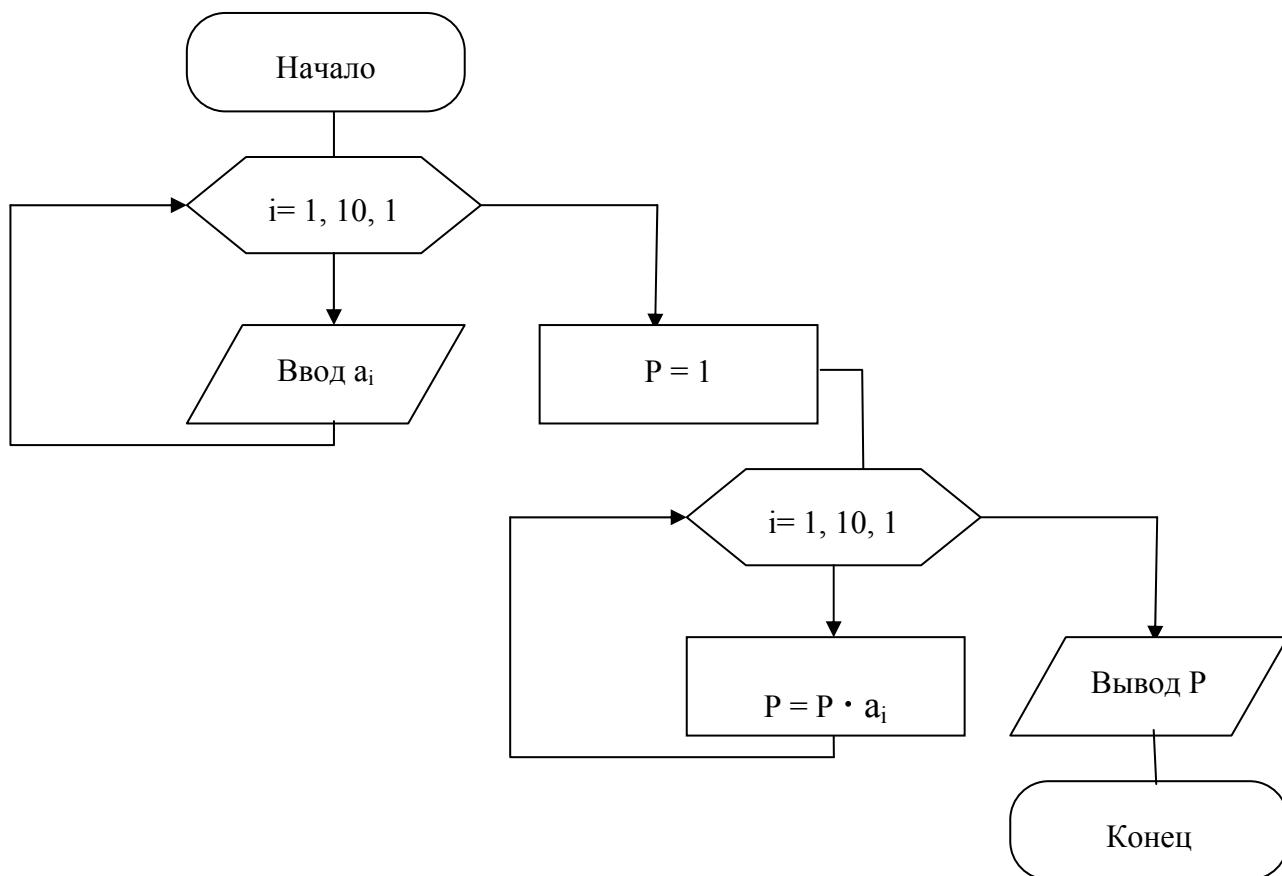


Рис. 20. Блок-схема к примеру 8

Двумерный массив

Двумерный массив – массив, каждый элемент которого имеет два индекса: номер строки i и номер столбца j (рис. 21).

В математике двумерный массив называется матрицей.

Также как и в одномерных массивах, ввод элементов матрицы осуществляется поэлементно. Но для работы с элементами матрицы необходимо использовать два цикла. Обработка двумерного массива заключается в том, что вначале поочередно рассматриваются все элементы первой строки, затем второй и т. д. Алгоритмическая конструкция матрицы реализует вложенный цикл. Внешний цикл определяет номер вводимой строки, а внутренний – номер элемента по столбцу.

Алгоритм ввода матрицы представлен на рис. 22.

Обработка элементов матрицы осуществляется в цикле, как и ввод.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & \dots & a_{1M} \\ a_{21} & a_{22} & a_{23} & \dots & \dots & a_{2M} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & a_{ij} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & a_{N3} & \dots & \dots & a_{NM} \end{pmatrix}$$

Рис. 21. Изображение матрицы в математике

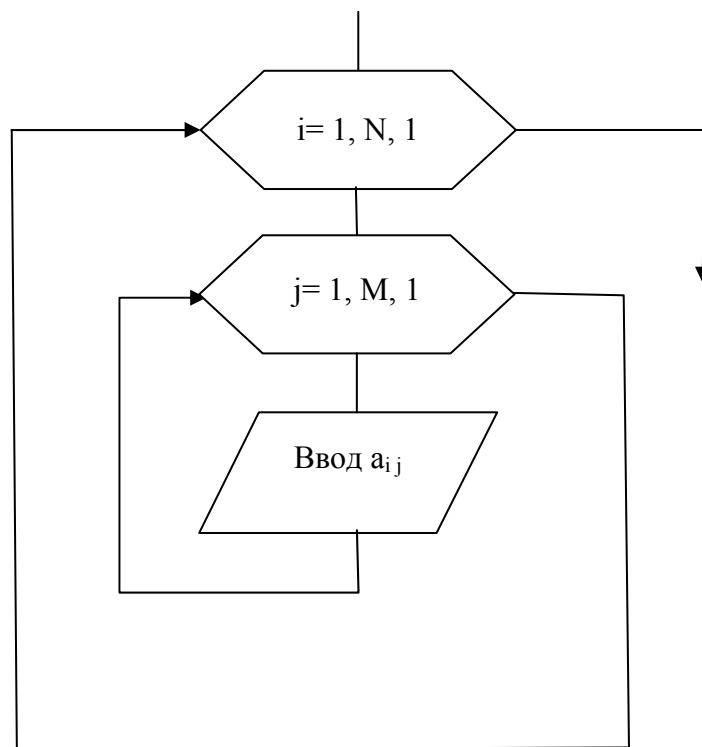


Рис. 22. Ввод элементов матрицы

ПРИМЕР 9. Разработать алгоритм к задаче: в двумерном массиве A (10x10) заменить все 1 на 0. Вывести новый массив.

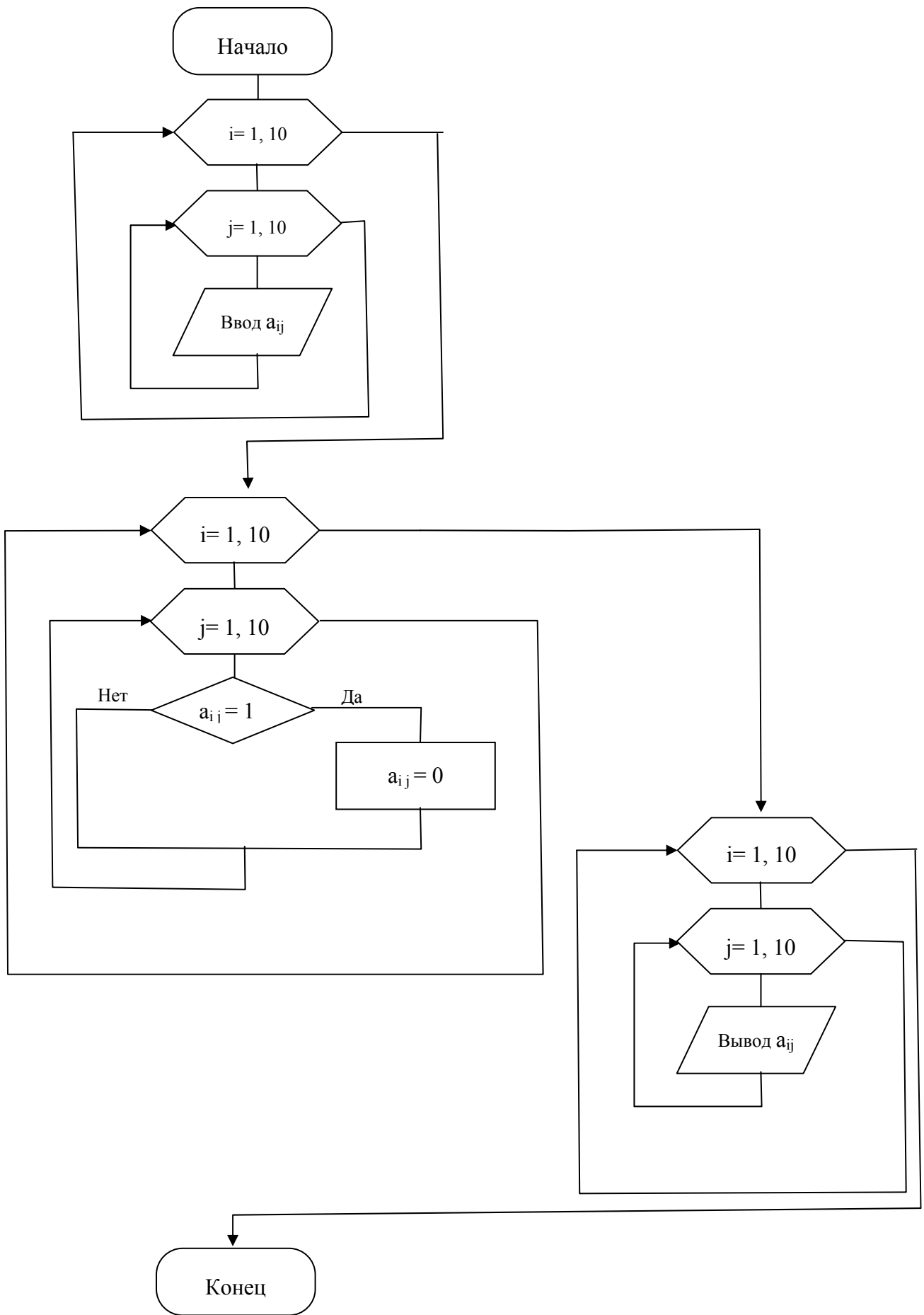
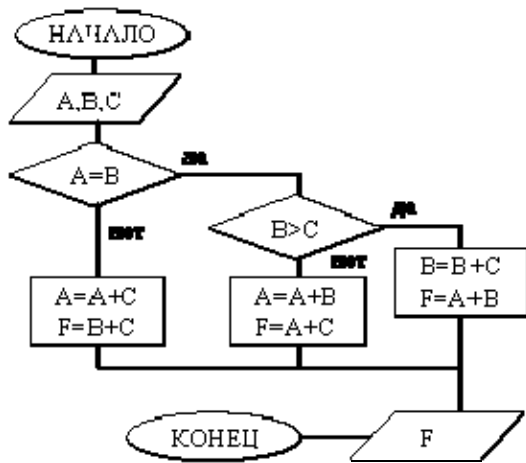


Рис. 23. Блок-схема к примеру 9

Задания для самостоятельного выполнения

1. Вычисленное по блок-схеме значение переменной F для входных данных 1, 1, 3 равно ...



- a) 7;
- b) 3;
- c) 4;
- d) 5;
- e) 6.

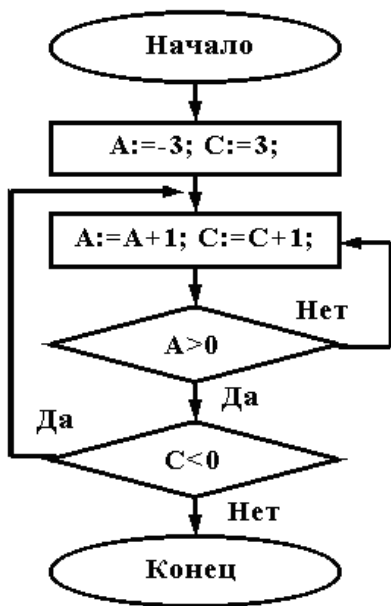
2. Установите порядок выполнения операций так, чтобы при начальных значениях $A = 1$, $B = -2$, $C = 0$ результирующим стало значение $C = 8$. Ответ запишите в виде последовательности букв (например, a b c d).

- a) $B = B * A$;
- b) $C = C * A$;
- c) $A = A + 1$;
- d) $C = C - B$.

3. Установите порядок выполнения операций так, чтобы при начальных значениях $A = 1$, $B = -2$, $C = 0$ результирующим стало значение $C = 8$. Ответ запишите в виде последовательности букв (например, a b c d)

- a) $B = B * A$;
- b) $C = C * A$;
- c) $A = A + 1$;
- d) $C = C - B$.

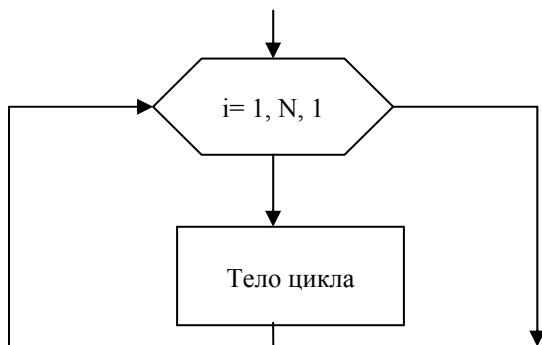
4. Выходные значения переменных A и C после выполнения алгоритма равны ...



- a) 1, 7;
- b) 0, -4;
- c) 1, 3;
- d) 0, -5.

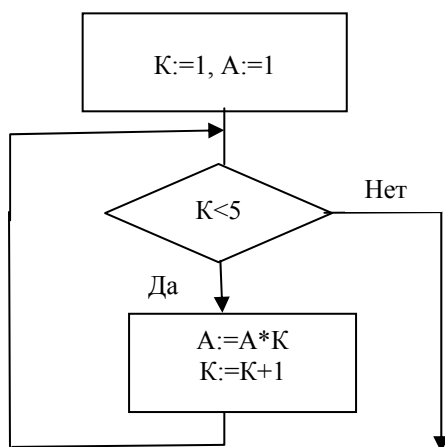
5. На рисунке представлен фрагмент алгоритма, имеющего структуру –

...



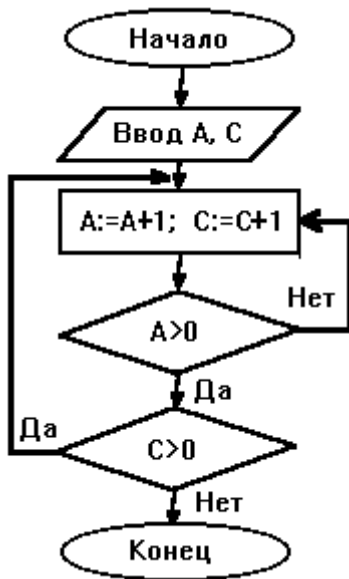
- a) неполное ветвление;
- b) цикл с постусловием;
- c) цикл с предусловием;
- d) цикл с параметром.

6. Представленный фрагмент блок-схемы алгоритма вычисляет ...



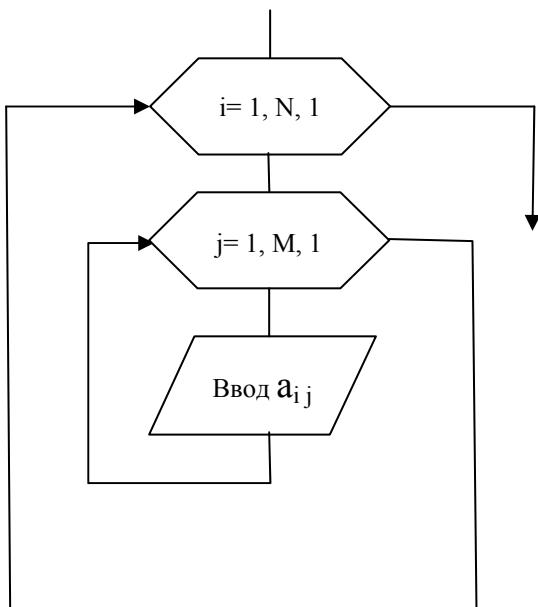
- a) $1*2*3*4$;
- b) A^5 ;
- c) $1*2*3*4*5$;
- d) A^4 .

7. Алгоритм закончит работу при начальных значениях переменных равных ...



- a) $A=-2; C=-3;$
- b) $A=-3; C=-2;$
- c) $A=-3, C=-3;$
- d) $A=-2; C=-1;$
- e) $A=-4, C=-3.$

8. Представленный фрагмент блок-схемы описывает ...



- a) ввод элементов одномерного массива;
- b) ввод элементов двумерного массива;
- c) цикл с предусловием;
- d) цикл с постусловием.