






Лабораторная работа 1

Основы алгоритмизации. Алгоритмы программ линейной структуры

Блок-схемой называется графическое изображение структуры алгоритма, в котором каждый этап процесса переработки данных представляется в виде геометрических фигур (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых при этом операций. Обозначение и функциональное назначение блоков представлено в таблице А.1.

Таблица А.1 – Обозначение и функциональное назначение блоков, используемых при описании алгоритма с помощью блок-схем

Наименование	Обозначение	Функция
1	2	3
Блок начало-конец (пуск-остановка)		Элемент отображает выход во внешнюю среду и вход из внешней среды (наиболее частое применение – начало и конец программы). Внутри фигуры записывается соответствующее действие.
Блок действия		Выполнение одной или нескольких операций, обработка данных любого вида (изменение значения данных, формы представления, расположения). Внутри фигуры записывают непосредственно сами операции, например, операцию присваивания: $a = 10b + c$.
Логический блок (блок условия)		Отображает решение или функцию переключательного типа с одним входом и двумя или более альтернативными выходами, из которых только один может быть выбран после вычисления условий, определенных внутри этого элемента. Вход в элемент обозначается линией, входящей обычно в верхнюю вершину элемента. Если выходов два или три, то обычно каждый выход обозначается линией, выходящей из оставшихся вершин (боковых и нижней). Если выходов больше трех, то их следует показывать одной линией, выходящей из вершины (чаще нижней) элемента, которая затем разветвляется. Соответствующие результаты вычислений могут записываться рядом с линиями, отображающими эти пути, в программировании это условный оператор, имеющий два выхода: true (+), false(-) или оператор выбора, имеющий множество выходов.

1	2	3
Предопределённый процесс (вызов процедуры функции)		Символ отображает выполнение процесса, состоящего из одной или нескольких операций, который определен в другом месте программы (в подпрограмме, модуле). Внутри символа записывается название процесса и передаваемые в него данные. В программировании это вызов процедуры или функции, с указанием входных и выходных параметров.
Данные (ввод-вывод)		Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод). Символ не определяет носителя данных (для указания типа носителя данных используются специфические символы).
Границы счетного цикла (цикла с параметром)		Символ состоит из двух частей – соответственно, начало и конец цикла – операции, выполняемые внутри цикла, размещаются между ними. Условия цикла и приращения записываются внутри символа начала или конца цикла – в зависимости от типа организации цикла. Формат надписи в начале цикла: <переменная цикла>=<начальное значение>; <конечное значение>; <шаг>. Пример: i=1,n,1. Формат надписи в конце цикла: <переменная цикла>. Пример: i. Для изображения на блок-схеме цикла с пред- или пост-условием вместо указанного символа используют символ условия, а одну из линий выхода замыкают выше в блок-схеме (перед операциями цикла).
Соединитель		Символ отображает вход в часть схемы и выход из другой части этой схемы. Используется для обрыва линии и продолжения её в другом месте (для избежания излишних пересечений или слишком длинных линий, а также, если схема состоит из нескольких страниц). Соответствующие соединительные символы должны иметь одинаковое (при том уникальное) обозначение.
Комментарий		Используется для более подробного описания шага, процесса или группы процессов. Описание помещается со стороны квадратной скобки и охватывается ей по всей высоте. Пунктирная линия идет к описываемому элементу, либо группе элементов (при этом группа выделяется замкнутой пунктирной линией). Также символ комментария следует использовать в тех случаях, когда объём текста, помещаемого внутри некоего символа (например, символ процесса, символ данных и др.), превышает размер самого этого символа.

Вычислительные процессы, используемые для решения различного рода задач на ЭВМ, в общем виде могут быть разделены на три большие группы: **линейные, разветвляющиеся и циклические.**

Линейным принято называть вычислительный процесс, в котором этапы вычислений выполняются в линейной последовательности и каждый этап выполняется только один раз. На схеме блоки размещаются сверху вниз в порядке их выполнения (рисунок 1). Для таких процессов характерно, что направление вычислений не зависит от исходных данных или промежуточных результатов.

Линейные процессы имеют место, например, при вычислении арифметических выражений.

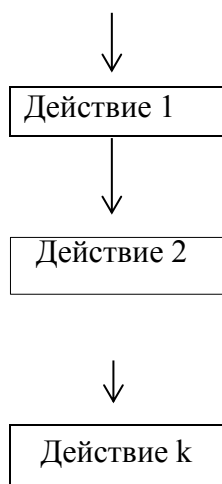


Рисунок 1 – Блок-схема линейного процесса

Разветвляющийся вычислительный процесс реализуется по одному из нескольких заранее предусмотренных направлений в зависимости от выполнения некоторого условия (логического выражения). Каждое направление вычислений называется ветвью. В любом конкретном случае процесс реализуется только по одной ветви, а выполнение остальных исключается. Ветвящийся процесс, включающий в себя две ветви, называется **простым**, более двух ветвей – **сложным**.

Сложный ветвящийся процесс можно представить с помощью простых ветвящихся процессов. Изображения полного и неполного вариантов ветвлений в виде блок-схем представлены на рисунках 2, 3.

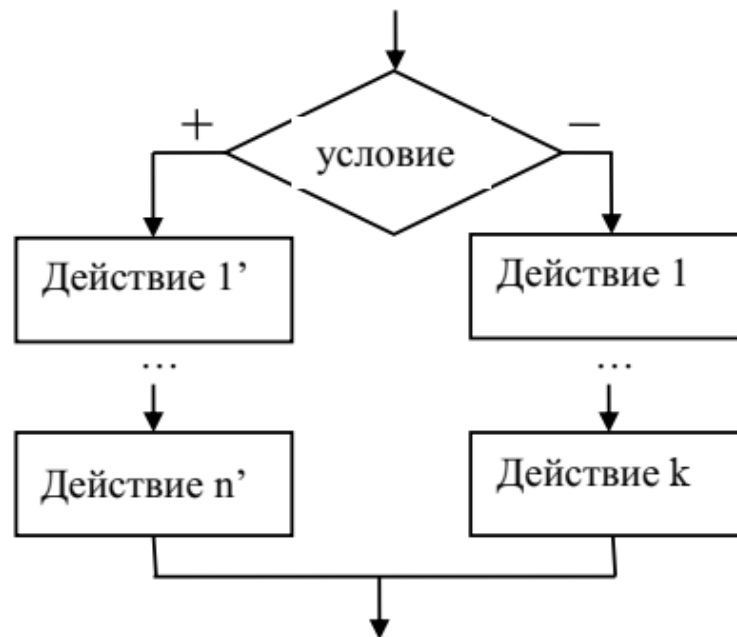


Рисунок 2 – Блок-схема полного варианта ветвления «если-то-иначе»

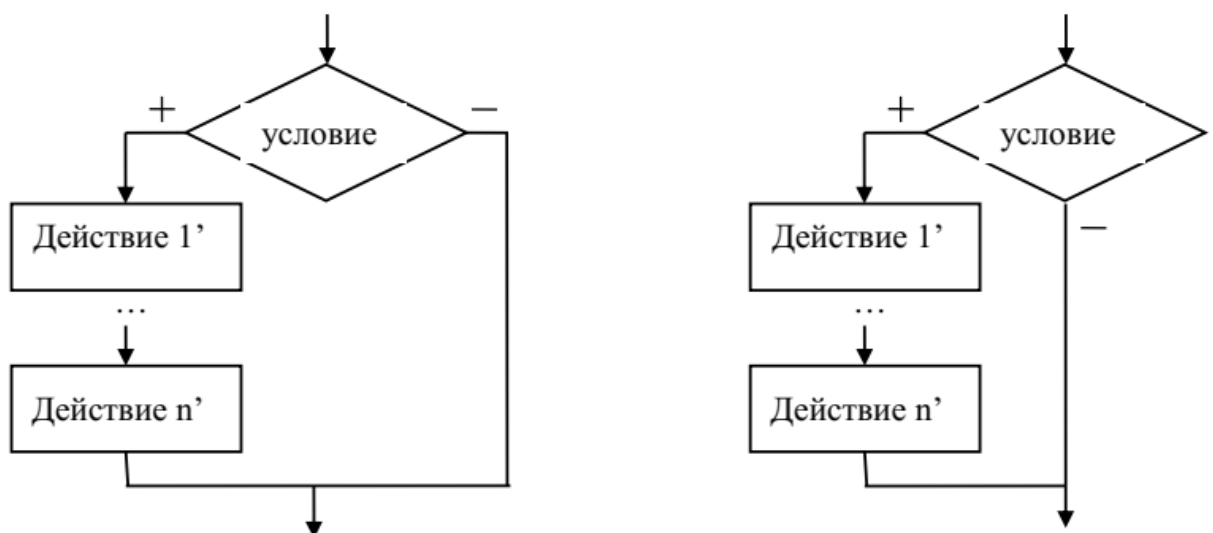


Рисунок 3 – Блок-схемы неполного варианта ветвления «если-то»

Циклический вычислительный процесс включает участки, на которых вычисления выполняются многократно по одним и тем же математическим формулам, но при разных значениях исходных данных. Такой многократно повторяющийся участок вычислений называется циклом. Для организации цикла необходимо предусмотреть:

- задание начального значения параметра цикла – переменной, которая будет изменяться при его повторении;
- изменение значения этой переменной перед каждым новым повторением цикла;
- проверку условия окончания цикла по значению его параметра и порядок перехода к началу цикла, если он не окончен.

Схема классификации циклов представлена на рисунке 4.

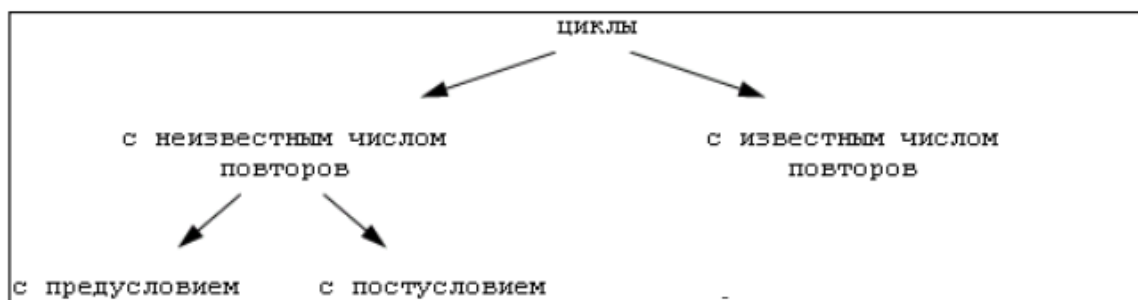


Рисунок 4 – Схема классификации циклических процессов

Цикл называется **детерминированным** (цикл с параметром), если число повторений тела цикла заранее известно или определено. Цикл называется **итерационным** (с пред- и постусловием), если число повторений тела цикла заранее неизвестно и зависит от значений параметров (некоторых переменных), участвующих в вычислениях.

Выполнение цикла «пока» начинается с проверки условия, поэтому такую разновидность циклов называют **циклы с предусловием**. Переход к выполнению действия осуществляется только в том случае, если условие выполняется, в противном случае происходит выход из цикла. Можно сказать, что условие цикла «пока» – это условие входа в цикл. В частном случае может оказаться, что действие не выполнялось ни разу. Условие цикла необходимо подобрать так, чтобы действия, выполняемые в цикле, привели к нарушению его истинности, иначе произойдет заикливание.

Зацикливание – это бесконечное повторение выполняемых действий. Блок-схема цикла с предусловием (цикла «пока») представлена на рисунке 5.

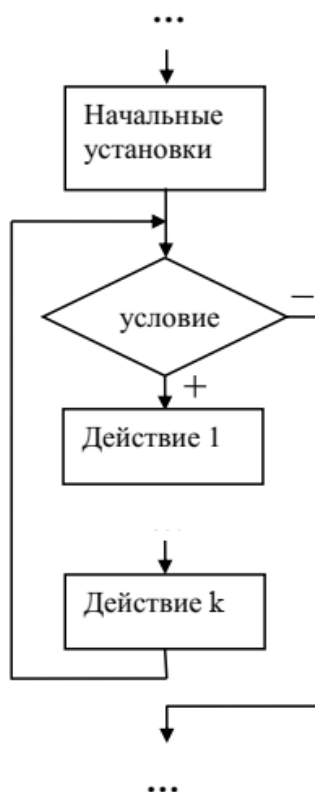


Рисунок 5 – Блок-схема цикла с предусловием (цикл «пока»)

Исполнение цикла «до» начинается с выполнения действия. Таким образом, тело цикла будет реализовано хотя бы один раз. После этого происходит проверка условия. Поэтому цикл «до» называют **циклом с постусловием**. Если условие не выполняется, то происходит возврат к выполнению действий. Если условие истинно, то осуществляется выход из цикла. Таким образом, условие цикла «до» – это условие выхода. Для предотвращения заикливания необходимо предусмотреть действия, приводящие к истинности условия. Блок-схема цикла с постусловием (цикла «до») представлена на рисунке 6.

Цикл с параметром (цикл со счетчиком или арифметический цикл) – это цикл с заранее известным числом повторов.

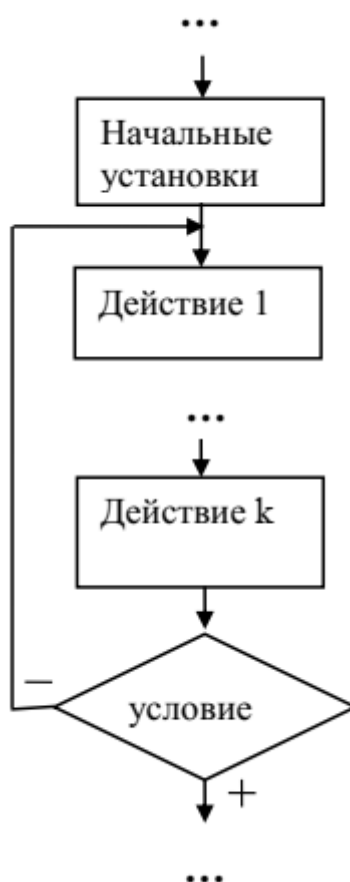


Рисунок 6 – Блок-схема цикла с постусловием (цикл «до»)

Блок-схема цикла с параметром представлено на рисунке 7.



Рисунок 7 – Блок-схема цикла с параметром

В блоке модификации указывается закон изменения переменной параметра: m – начальное значение параметра цикла; h – шаг; n – последнее значение параметра цикла.

Для создания циклов с параметром необходимо соблюдать следующие правила:

- 1) параметр цикла, его начальное, конечное значения и шаг должны быть одного типа;
- 2) запрещено изменять в теле цикла начальное, текущее и конечное значения для параметра;
- 3) запрещено входить в цикл, минуя блок модификации;
- 4) если начальное значение больше конечного, то шаг – число отрицательное;
- 5) после выхода из цикла значение переменной параметра не определено и не может использоваться в дальнейших вычислениях;
- 6) из цикла можно выйти не закончив его, тогда переменная параметр сохраняет свое последнее значение.

Реальные данные, которые обрабатывает программа, – это целые и вещественные числа, символы и логические величины. Эти простые типы данных называют базовыми. Все данные, обрабатываемые компьютером, хранятся в ячейках памяти компьютера, каждая из которых имеет свой адрес. Для того чтобы не следить за тем, по какому адресу будут записаны те или иные данные, в языках программирования используется понятие переменной, позволяющее отвлечься от адреса ячейки памяти и обращаться к ней с помощью имени (идентификатора).

Переменная – это именованный объект (ячейка памяти), который может изменять своё значение. Имя переменной указывает на значение, а способ ее хранения и адрес остаются скрытыми от программиста. Кроме имени и значения, переменная имеет тип,

определяющий, какая информация находится в памяти. Тип переменной задает используемый способ записи. Объем памяти для каждого типа определяется таким образом, чтобы в него можно было поместить любое значение из допустимого диапазона значений данного типа. Например, тип «байт» может принимать значения от 0 до 255, что в двоичном коде ($255 = 11111111$) соответствует ячейке памяти длиной в 8 бит (или 1 байт). В описанных выше алгоритмах все данные хранятся в виде переменных.

Например, инструкция «Ввод двух чисел a , b » означает введение пользователем значений двух переменных, а инструкция « $k = k + 1$ » означает увеличение значения переменной k на единицу. Переменные, которые присутствуют в программе на протяжении всего времени ее работы, называют **статическими**. Переменные, создающиеся и уничтожающиеся на разных этапах выполнения программы, называют **динамическими**. Данные, не меняющиеся на протяжении работы программ, называются константами или постоянными. Константы, как и переменные, имеют тип. Константы можно указывать явно (например, в инструкции « $k = k + 1$ » есть константа (единица)) или для удобства обозначать идентификаторами ($\pi = 3,1415926536$). Значение π нельзя изменить, так как это константа, а не переменная.

Для повышения производительности и качества работы программы необходимо иметь данные, максимально приближенные к реальным аналогам. Тип данных, позволяющий хранить вместе под одним именем несколько переменных, называется **структурированным**. Каждый язык программирования имеет свои структурированные типы. Рассмотрим структуру, объединяющую элементы одного типа данных, – **массив**. Массивом называется упорядоченная совокупность однотипных величин, имеющих общее имя, элементы которой адресуются (различаются) порядковыми номерами (индексами). В качестве иллюстрации можно представить шкаф, содержащий множество пронумерованных ящиков (совокупность – «Ящик № 1», «Ящик № 2», «Ящик № 3» и т.д.; «Ящик» – общее имя всех ее элементов). Доступ к содержимому конкретного ящика (элементу массива) осуществляется после выбора ящика по его номеру (индексу). Элементы массива в памяти компьютера хранятся по соседству, одиночные элементы простого типа такого расположения данных в информации в ячейки памяти и необходимый объем памяти для ее хранения памяти не предполагают. Массивы различаются количеством индексов, определяющих их элементы. **Одномерный массив** предполагает наличие у каждого элемента только одного индекса. Примерами одномерных массивов служат арифметическая и геометрическая последовательности, определяющие конечные ряды чисел. Количество элементов массива называют **размерностью**. При определении одномерного массива его размерность записывается в круглых скобках, рядом с его именем. Например, если сказано: «задан массив $A(10)$ », это означает, что даны элементы: a_1, a_2, \dots, a_{10} . Рассмотрим алгоритмы обработки элементов одномерных массивов. Ввод элементов одномерного массива осуществляется поэлементно, в порядке, необходимом для решения конкретной задачи. Обычно, когда требуется ввести весь

массив, порядок ввода элементов не важен, и элементы вводятся в порядке возрастания их индексов.

В математике двумерный массив (таблица чисел) называется **матрицей**. Каждый ее элемент имеет два индекса (aij) , первый индекс i определяет номер строки, в которой находится элемент (координата по горизонтали), а второй j – номер столбца (координата по вертикали). Двумерный массив характеризуется двумя размерностями N и M , определяющими число строк и столбцов соответственно.

Ввод элементов двумерного массива осуществляется построчно, в свою очередь, ввод каждой строки производится поэлементно, тем самым определяется циклическая конструкция, реализующая вложение циклов. Внешний цикл определяет номер вводимой строки (i), внутренний – номер элемента по столбцу (j).

Примеры разработки схем алгоритмов

Задача 1 (пример линейного процесса) Разработать блок-схему алгоритма вычисления площади и периметра треугольника, зная длины сторон треугольника.

Рассмотрим теоретическое обоснование решения задачи. Пусть a, b, c – длины сторон треугольника. Необходимо найти S – площадь треугольника, P – периметр.

Для нахождения площади треугольника можно воспользоваться формулой Герона:

$$S = \sqrt{r(r-a)(r-b)(r-c)}, \quad \text{где } r = \frac{P}{2}, \quad P = a + b + c$$

- периметр треугольника.

Псевдокод алгоритма и блок-схема алгоритма решения задачи представлены на рисунке 2.1.

Входные данные: a, b, c .

Выходные данные: S, P .

Шаг 1. Ввод a, b, c

Шаг 2. $P = a + b + c$

Шаг 3. $r = \frac{P}{2}$

Шаг 4. $S = \sqrt{r(r-a)(r-b)(r-c)}$

Шаг 5. Вывод S, P

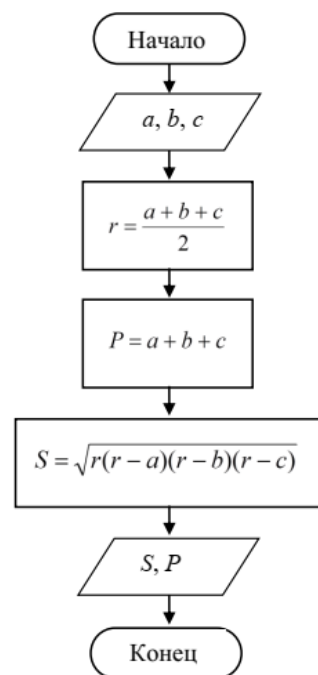


Рисунок 2.1 –Блок-схема алгоритма решения задачи 1

Задача 2 (пример разветвляющегося процесса) Разработать блок-схему алгоритма нахождения наименьшего из трех чисел. Пусть a , b , c – три числа, вводимые пользователем. Рассмотрим алгоритм решения задачи. Входные данные: a , b , c .

Выходные данные: m , где $m = \min\{a, b, c\}$.

Ниже приведен псевдокод алгоритма нахождения наименьшего из трех чисел.

Шаг 1. Ввод a , b , c .

Шаг 2. Если $a < b$, тогда

Шаг 2.1 Если $a < c$, тогда $m = a$ иначе $m = c$ иначе

Шаг 2.2 Если $b < c$, тогда $m = b$ иначе $m = c$

Шаг 3. Вывод m .

Блок-схемы алгоритмов решения задачи представлены на рисунках 2.2, 2.3.

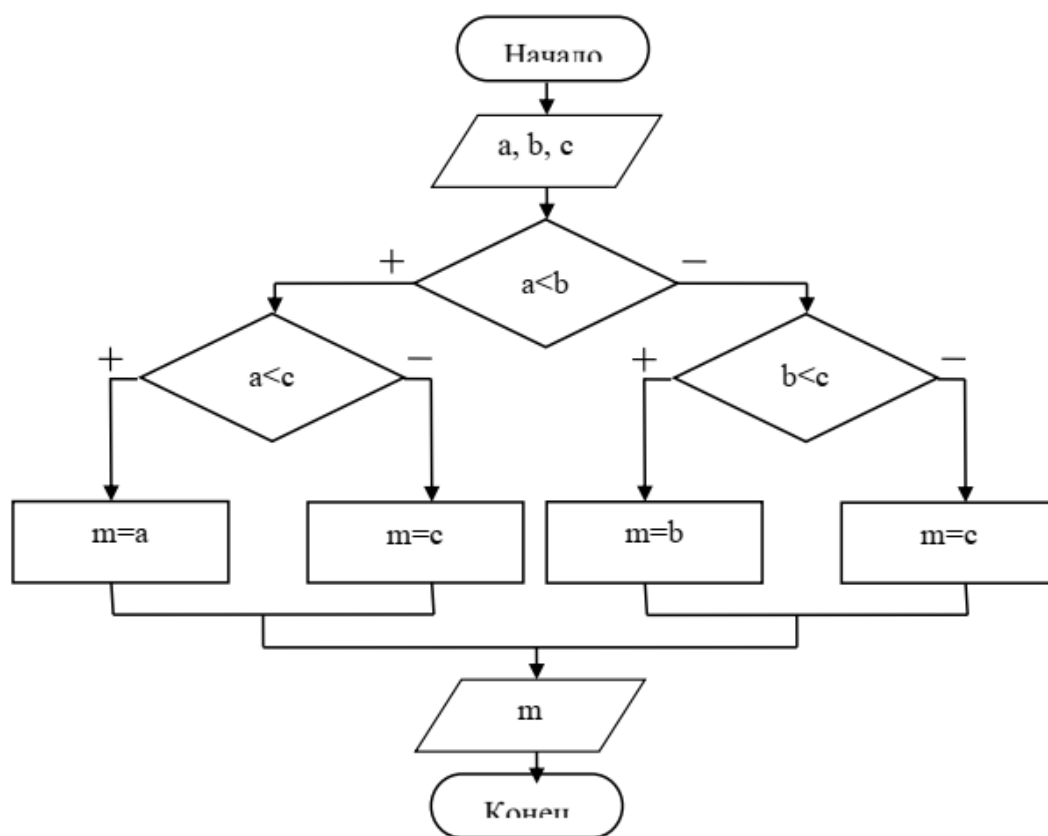


Рисунок 2.2 – Блок-схема алгоритма решения задачи 2 (алгоритм а)

Шаг 1. Ввод a, b, c

Шаг 2. $m=a$

Шаг 3. Если $m > b$, тогда $m=b$

Шаг 4. Если $m > c$, тогда $m=c$

Шаг 5. Вывод m

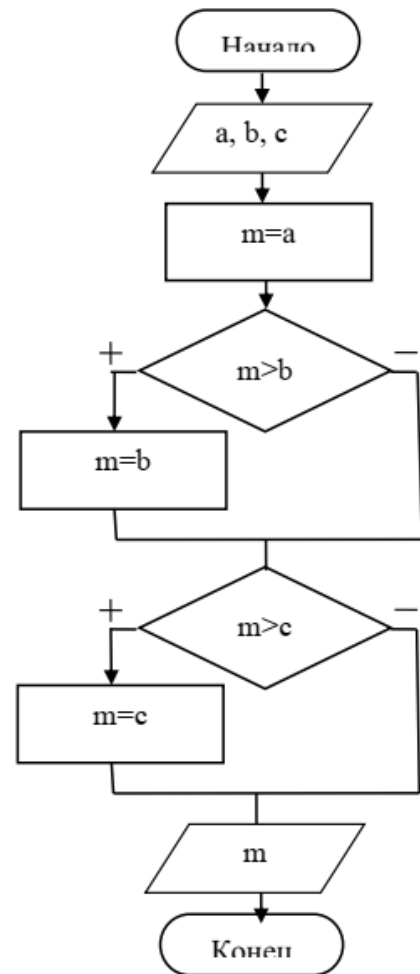


Рисунок 2.2 –Блок-схема алгоритма решения задачи 2 (алгоритм б)

Задача 3 (пример циклического процесса)

Разработать блок-схему алгоритма вычисления факториала натурального числа n . Рассмотрим теоретическое обоснование решения задачи. Факториал натурального числа вычисляется по формуле: $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$. Для решения задачи будет использоваться цикл со счетчиком i . Сформулируем правило произведения:

- начальное значение произведения $S=1$;
- в теле некоторой циклической конструкции выполнить команду $S = S \cdot \langle \text{множитель} \rangle$.

Псевдокод алгоритма и блок-схема алгоритма решения задачи с использованием цикла с параметром представлены на рисунке 2.4.

Шаг 1. Ввод n

Шаг 2. $S = 1$

Шаг 3. Цикл для $i = 1$ до n шаг 1

Шаг 3.1 $S = S \cdot i$

Конец цикла

Шаг 4. Вывод S

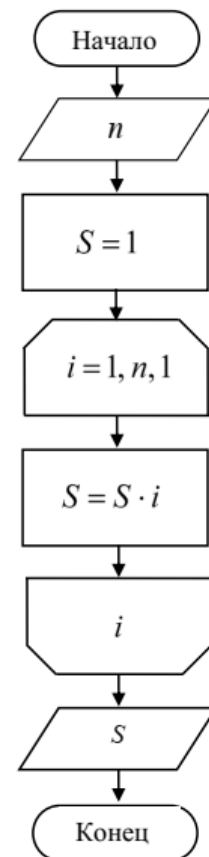


Рисунок 2.4 – Блок-схема алгоритма решения задачи 3 с использованием цикла с параметром

Псевдокод алгоритма и блок-схема алгоритма решения задачи 3 с использованием цикла «пока» представлены на рисунке 2.5.

Шаг 1. Ввод n
 Шаг 2. $S = 1; i = 1$
 Шаг 3. Цикл «пока» $i \leq n$
 3.1 $S = S \cdot i$
 3.2 $i = i + 1$
 Конец цикла «пока»
 Шаг 4. Вывод S

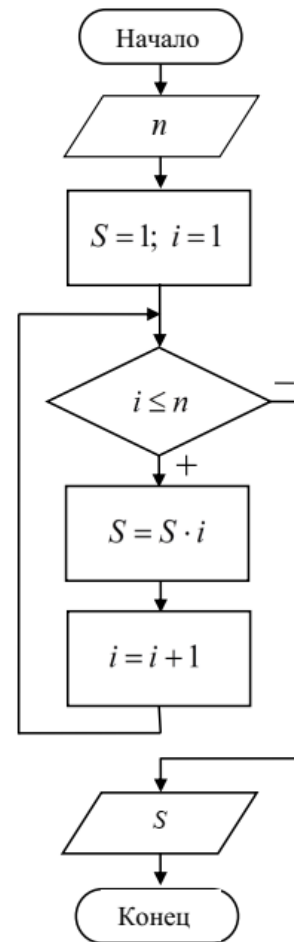


Рисунок 2.5 – Блок-схема алгоритма решения задачи 3 с использованием цикла «пока»

Псевдокод алгоритма и блок-схема алгоритма решения задачи с использованием цикла «до» представлены на рисунке 2.6.

Шаг 1. Ввод n
 Шаг 2. $S = 1; i = 1$
 Шаг 3. Начало цикла «до»
 Шаг 3.1 $S = S \cdot i$
 Шаг 3.2 $i = i + 1$
 Конец цикла «до» ($i > n$)
 Шаг 4. Вывод S

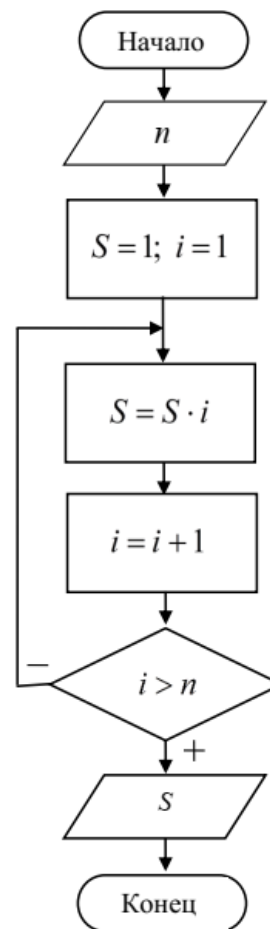


Рисунок 2.6 – Блок-схема алгоритма решения задачи 3 с использованием цикла «до»

Задача 4 Разработать алгоритм вычисления среднего арифметического значения элементов числового массива $A(10)$.

Псевдокод и блок-схема алгоритма решения задачи представлены на рисунке 2.7.

Входные данные: $A(10)$.

Выходные данные: D – среднее арифметическое значение.

Шаг 1. Цикл для $i = 0$ до 9 шаг 1

1.1 Ввести $A[i]$

Конец цикла

Шаг 2. $D = 0$

Шаг 3. Цикл для $i = 0$ до 9 шаг 1

3.1 $D = D + A[i]$

Конец цикла

Шаг 4. $D = D/10$

Шаг 5. Вывод D

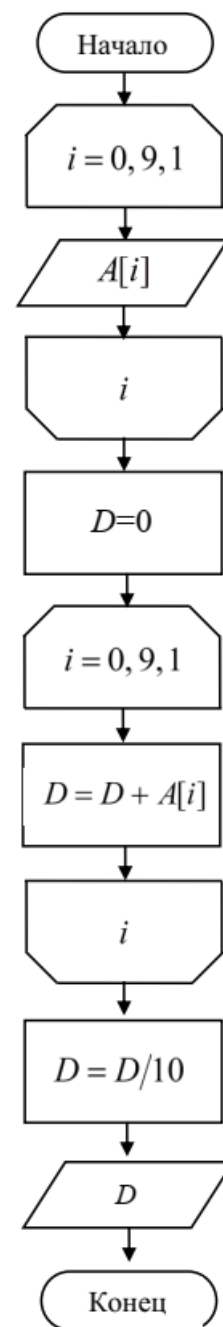


Рисунок 2.7 –Блок-схема алгоритма решения задачи 4

Задание к лабораторной работе, требования к оформлению и защите отчета по лабораторной работе

Задание к лабораторной работе: разработать блок-схемы алгоритмов решения задач (задания в соответствии с вариантом).

Выполнение лабораторной работы состоит из следующих этапов: 1) ознакомление с формулировкой задания к лабораторной работе и порядком её выполнения; 2) разработка алгоритмов; 3) оформление результатов; 4) подготовка отчета по лабораторной работе; 5) защита отчета по лабораторной работе.

Отчет по лабораторной работе оформляется в соответствии с требованиями. Отчет должен содержать титульный лист; задание к лабораторной работе, включая перечень задач заданного варианта; краткие теоретические сведения по теме лабораторной работы; результаты выполнения лабораторной работы – блок-схемы алгоритмов задач. Схемы алгоритмов могут быть оформлены с использованием редактора изображений MS Word, MS Visio, графических редакторов, к примеру, редактор Gimp, либо допускается оформлять от руки, на листах формата А4, аккуратно карандашом и/или черной ручкой, блоки изображать по линейке, текст в блоках печатным шрифтом. Для защиты отчета по лабораторной работе необходимо подготовиться к ответу на вопросы, приведенные ниже.

1) Приведите основные блоки, используемые для изображения блок-схем алгоритмов, опишите их назначение. 2) Приведите основные алгоритмические конструкции.

3) Объясните последовательность этапов решения задач вашего варианта.

Контрольные вопросы:

- 1) Что такое блок-схема?
- 2) Какие бывают блоки и для чего они предназначены?
- 3) На какие группы можно разделить вычислительные процессы? Охарактеризуйте каждый из них.
- 4) Какие бывают циклы? Чем они отличаются?
- 5) Что такое заикливание?
- 6) Что такое переменная? Какими они бывают?
- 7) Что такое массив? Какие они бывают?

Задания к лабораторной работе

Нарисовать блок-схемы алгоритмов решения приведенных задач. Ниже приведены варианты заданий.

Вариант 1

- 1) Даны три целых числа. Возвести в квадрат отрицательные числа и в третью степень — положительные (число 0 не изменять).
- 2) Даны два целых числа А и В ($A < B$). Вывести все целые числа, расположенные между данными числами (включая сами эти числа), в порядке их возрастания, а также количество N этих чисел.
- 3) Даны целые числа b_1, b_2, \dots, b_{10} . Выяснить, верно ли, что сумма тех из них, которые больше 20, превышает 100.
- 4) Дано ребро куба a . Вычислить площадь одной грани куба, его объем и площадь полной поверхности.
- 5) Вычислить выражение

$$\frac{b + \sqrt{b^2 + 4ac}}{2a} - a^3c + b^{-2};$$

Вариант 2

1) Значения переменных X, Y, Z поменять местами так, чтобы они оказались упорядоченными по возрастанию.

2) Даны два целых числа A и B ($A < B$). Вывести все целые числа, расположенные между данными числами (не включая сами эти числа), в порядке их убывания, а также количество N этих чисел.

3) Даны целые числа b_1, b_2, \dots, b_{10} . Выяснить, верно ли, что сумма тех из них, которые меньше 50, есть чётное число.

4) Дано положительное число a. Вычислить: а) площадь равностороннего треугольника со стороной a; б) площадь квадрата со стороной a; в) площадь круга, радиус которого равен a.

5) Для заданных значений вычислить результат:

$$\frac{\sin x + \cos y}{\cos x - \sin y} \operatorname{tg} xy$$

Вариант 3

1) Даны две переменные целого типа: A и B. Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения.

2) Дано вещественное число A и целое число N (> 0). Вывести все целые степени числа A от 1 до N.

3) Даны натуральное число n и целые числа a_1, a_2, \dots, a_n . Выяснить, верно ли, что сумма тех чисел, a_i , которые меньше 20.5, не превышает 50.

4) Пусть идет k-ая секунда суток. Определить, сколько целых часов h и целых минут m прошло к этому моменту.

$$\sqrt{\sin^2(x) + \cos^2(y^3)};$$

5) Для заданных значений вычислить результат:

Вариант 4

1) Даны две переменные целого типа: A и B. Если их значения не равны, то присвоить каждой переменной максимальное из этих значений, а если равны, то присвоить переменным нулевые значения.

2) Дано целое число N (> 0). Вывести произведение $1 \cdot 2 \cdot \dots \cdot N$. Чтобы избежать целочисленного переполнения, вычислять это произведение с помощью вещественной переменной и выводить его как вещественное число.

3) Даны натуральное число n и целые числа a_1, a_2, \dots, a_n . Выяснить, верно ли, что сумма тех чисел a_i , которые не превышают 10, кратна трем.

4) Даны координаты вершин треугольника. Вычислить его площадь.

Для заданных a и b вычислить:

$$\begin{cases} z = x^3 + y^3 + 10, \\ y = \sin(b) - \cos(a + x), \\ x = \sqrt{|a+1|}. \end{cases}$$

5)

Вариант 5

1) Даны четыре целых числа, одно из которых отлично от трех других, равных между собой. Вывести порядковый номер этого числа.

2) Дано целое число $N (> 0)$. Вывести сумму $2 + 1/(2!) + 1/(3!) + \dots + 1/(N!)$

3) Даны натуральное число n и вещественные числа x_1, x_2, \dots, x_n . Выяснить, верно ли, что сумма тех вещественных чисел, которые больше 20.5, меньше p .

4) Присвоить целой переменной L сумму последних цифр целой части положительного вещественного числа X (так, если $X=143,57$, то $L=4+3=7$).

$$y = \frac{\cos(x_1 + 5) + x_2}{3x_1 + 6x_2} * |\sin x_1|^2$$

5) Вычислить

Вариант 6

1) Значениями переменных A, B, C являются цифры. Присвоить целой переменной L число, составленное из целых цифр (например $A=1, B=3, C=5$, то $L=135$).

2) Дано трехзначное число L . Трех переменным присвоить значения каждой цифры числа L (например, $L=135$, тогда $A=1, B=3, C=5$).

3) Целой переменной X присвоить значение суммы цифр заданного трехзначного числа.

4) Найти произведение P двух первых цифр из дробной части положительного вещественного числа X (так если $X=31,956$, то $P=9*5=45$).

$$\operatorname{ctg}(x) - \sin\left(\sqrt{x^2 + 1}\right);$$

5) Вычислить

Вариант 7

1) Присвоить двум переменным A и B крайнюю левую и крайнюю правую цифры трехзначного числа K , переменной C присвоить сумму цифр числа K (например, если $K=135$, тогда $A=1, B=5, C=1+3+5=9$).

2) Определить число, полученное выписыванием в обратном порядке цифр заданного трехзначного целого числа.

3) Даны катеты a и b прямоугольного треугольника. Вычислить длину гипотенузы и площадь треугольника.

4) Целой переменной X присвоить значение произведения цифр заданного четырехзначного числа.

5) Вычислить
$$\frac{3 + e^{y-1}}{1 + x^2 |y - \operatorname{tg} x|};$$

Вариант 8

1) Дано целое четырехзначное число. Найти сумму его цифр.

2) Даны длины ребер a , b , c прямоугольного параллелепипеда. Найти его объем V и площадь поверхности S .

3) Найти расстояние между двумя точками с заданными координатами (x_1, y_1) и (x_2, y_2) .

4) Поменять местами значения целых переменных x и y .

$$y = \frac{\log_3 |x - 10|}{2x - 7} - \cos(3x + 5)$$

5) Вычислить