

Алгоритмы
Основные алгоритмические конструкции
Сборник задач

СОДЕРЖАНИЕ

ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Немного из истории появления термина «алгоритм».....	4
Основные свойства и способы представления алгоритма	6
Базовые структуры программирования.....	10

ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ НА АЛГОРИТМЫ

1. Линейный алгоритм	16
2. Разветвляющийся алгоритм	18
3. Неполное ветвление	19
4. Цикл с предусловием	20
5. Цикл с постусловием	22

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

1. Задачи на линейный алгоритм.....	24
2. Задачи на разветвляющийся алгоритм	26
3. Задачи на цикл с предусловием.....	28
4. Задачи на цикл с постусловием	30
5. Дополнительные задачи	31

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ.....40

ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Большинство действий, совершаемых человеком, выполняются по определенным правилам. Их эффективность во многом зависит от того, насколько он представляет, что делать в каждый момент времени, в какой последовательности, каким должен быть итог его действий. Так, например, применение в производстве и быту различных автоматов, компьютеров требует от человека строгого соблюдения определенной последовательности действий при их использовании, что невозможно без предварительного составления алгоритмов.

Таким образом, осмысление и разработка алгоритмов выполняемых действий становится существенным компонентом деятельности человека, составной частью его культуры мышления и поведения. Алгоритм – одно из фундаментальных понятий, которое используется в различных областях знания, но изучается оно в математике и информатике. Его освоение начинается уже в начальной школе на уроках математики, где ученики овладевают алгоритмами арифметических действий, знакомятся с правилами вычитания числа из суммы, суммы из числа и др.

Немного из истории появления термина «алгоритм»

Происхождение термина «алгоритм» связано с математикой. История его возникновения такова. В IX веке в Багдаде жил

ученый ал(аль)-Хорезми (полное имя – Мухаммед бен Муса ал-Хорезми), математик, астроном, географ. В одном из своих трудов он описал десятичную систему счисления и впервые сформулировал правила выполнения арифметических действий над целыми числами и обыкновенными дробями. Арабский оригинал этой книги был утерян, но остался латинский перевод XII в., по которому Западная Европа ознакомилась с десятичной системой счисления и правилами выполнения арифметических действий.



Правила в книгах ал-Хорезми в латинском переводе начинались словами «Алгоризми сказал». В других латинских переводах автор именовался как Алгоритмус. Со временем было забыто, что Алгоризми (Алгоритмус) – это автор правил, и эти правила стали называть алгоритмами. Многие столетия разрабатывались алгоритмы для решения все новых и новых классов задач, но само понятие алгоритма не имело точного математического определения.

В настоящее время понятие алгоритма уточнено.

Алгоритм – понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение цели.

Основные свойства и способы представления алгоритма

Любой алгоритм должен обладать следующими **свойствами**:

- **определенностью** – за конечное число шагов либо должен быть получен результат, либо доказано его отсутствие;

- **результативностью** – обязательным получением некоторого результата (числа, таблицы, текста, звука, изображения и т. д.) или сигнала о том, что данный алгоритм неприменим для решения поставленной задачи;

- **массовостью** – возможностью получения результата при различных исходных данных для некоторого класса сходных задач;

- **формальностью** – отвлечение от содержания поставленной задачи и строгое выполнение некоторого правила, инструкции;

- **дискретностью** — возможностью разбиения алгоритма на отдельные элементарные действия.

Существуют следующие *формы представления алгоритма*:

- Словесная (вербальная) на неформальном языке.
- На языках программирования.
- Графическая.

Словесная форма представления алгоритма имеет ряд недостатков. Для достаточно сложных алгоритмов описание

становится слишком громоздким и не наглядным. Эта форма представления обычно используется лишь на начальных стадиях разработки алгоритма.

Пример словесной формы описания алгоритма: *Чтобы перейти улицу, нужно посмотреть налево, убедиться в отсутствии приближающегося транспорта, дойти до середины улицы, посмотреть направо, убедиться в отсутствии близко идущего транспорта, продолжить движение через улицу. При наличии движущихся транспортных средств нужно ждать, когда транспорт проедет.*

Алгоритм, записанный на **языке программирования**, называется *программой*.

Графическая форма представления алгоритмов является более наглядной и строгой. Алгоритм изображается в виде последовательности связанных между собой блоков, каждый из которых соответствует выполнению одного или нескольких операторов. Такое графическое представление называется **блок-схемой алгоритма**.

Условные графические обозначения символов, используемых для составления блок-схемы алгоритма, стандартизированы [2]. Некоторые, часто используемые обозначения, приведены в табл. 1.

Таблица 1

Условные графические обозначения символов

Название блока	Обозначение	Название блока	Обозначение
Начало или конец алгоритма		Решение	
Процесс (действие или серия действий)		Предопределенный процесс (вспомогательный алгоритм)	
Ввод/вывод данных		Модификация (заголовок цикла)	
Линии потока		Комментарии	

Отдельные блоки алгоритмов соединяются между собой линиями потоков, которые проводятся параллельно внешней рамке чертежа. Направления линий потока сверху вниз и слева направо принимаются за основные и, если линии потоков не имеют изломов, стрелками не обозначаются. Обратные направления линий потока помечаются стрелкой.

«Процесс» (этап вычисления) изображается прямоугольником, внутри которого записывается набор действий. Ромбом изображается «решение», внутри которого осуществляется проверка условия. Ввод исходных данных и вывод результатов изображаются параллелограммами, внутри которых пишутся слова

«ввод» или «вывод» и перечисляются переменные, подлежащие вводу или выводу.

Представление алгоритма в виде блок-схемы является промежуточным, так как алгоритм в таком виде не может быть непосредственно выполнен ЭВМ, но помогает пользователю при создании (написании) программы для ПК.

Использование блок-схем дает возможность:

- наглядно отобразить базовые конструкции алгоритма;
- сосредоточить внимание на структуре алгоритма, а не на синтаксисе языка;
- анализировать логическую структуру алгоритма;
- преобразовывать алгоритм методом укрупнения (сведения к единому блоку) или детализации – разбиения на ряд блоков;
- использовать принцип блочности при коллективном решении сложной задачи;
- осуществить быструю проверку разработанного алгоритма (на уровне идеи);
- разобрать большее число учебных задач.

Составление блок-схемы алгоритма является важным и в большинстве случаев необходимым этапом решения сложной и большой задачи на ЭВМ, значительно облегчающим процесс составления программ.

Базовые структуры программирования

Выделяют три основные структуры алгоритмов:

1. Линейная.
2. Разветвляющаяся (альтернатива «если–то–иначе» или «если–то»).
3. Циклическая (повторение).

Линейная структура – является основной. Она означает, что действия выполняются друг за другом (рис. 1).



Рис. 1. Линейная структура

Прямоугольник, показанный на рисунке, может представлять как одну единственную команду, так и множество операторов, необходимых для выполнения сложной обработки данных, где **F1** и **F2** – некоторые команды для соответствующего исполнителя. Команды записываются с помощью операции присваивания.

Присваивание переменной какого-либо значения или присваивание одной переменной значения другой переменной является наиболее часто выполняемым действием в программе, написанной на любом языке программирования. В языке программирования присваивание является операцией и обозначается как «:=». Это означает, что при выполнении этой операции происходит не только присваивание значения определенной переменной, но и возвращается некоторое значение.

Разветвляющаяся структура (ветвление) – это структура, обеспечивающая альтернативный выбор в зависимости от заданного условия. Выполняется проверка условия, а затем выбирается один из путей (рис. 2), где *P* – это условие, в зависимости от истинности (Да)¹ или ложности (Нет) которого управление передается по одной из двух ветвей.

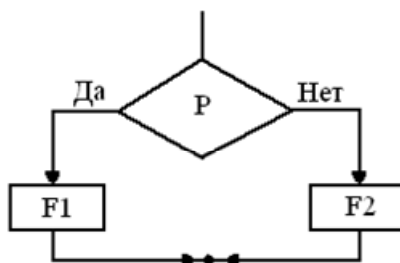


Рис. 2. Ветвление

Может оказаться, что для одного из результатов проверки условия ничего предпринимать не надо. В этом случае можно применять только один обрабатывающий блок (рис. 3).

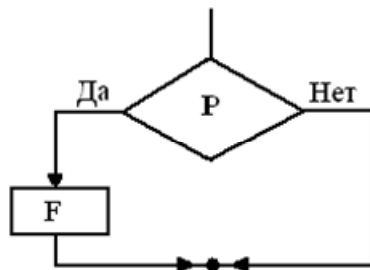


Рис. 3. Неполное ветвление

Эта структура называется также **ЕСЛИ – ТО – ИНАЧЕ**. Каждый из путей (ТО или ИНАЧЕ) ведет к общей точке слияния, так что выполнение программы продолжается независимо от того, какой путь был выбран.

Циклическая структура (или повторение) предусматривает повторное выполнение некоторого набора действий.

¹ В литературе часто обозначают вместо слова «Да» знак «+», а вместо слова «Нет» пишут «-».

Циклы позволяют записать длинные последовательности операций обработки данных с помощью небольшого числа повторяющихся команд.

Итерационным называется цикл, число повторений которого не задается, а определяется в ходе выполнения цикла. В этом случае одно повторение цикла называется **итерацией**.

Рекурсия – это такая ситуация, когда некоторый алгоритм непосредственно или через другие алгоритмы вызывает себя в качестве вспомогательного. Сам алгоритм при этом называется рекурсивным.

В качестве примера использования рекурсии рассмотрим задачу поиска файлов. Пусть нужно получить список всех файлов, например, с расширением bmp, которые находятся в указанном пользователем каталоге и во всех подкаталогах этого каталога.

Словесно алгоритм обработки каталога может быть представлен так:

1. Вывести список всех файлов, удовлетворяющих критерию запроса.
2. Если в каталоге есть подкаталоги, то обработать каждый из этих каталогов.

Приведенный алгоритм, блок-схема которого представлена на рис. 4, является рекурсивным. Для того чтобы обработать подкаталог, процедура обработки текущего каталога должна вызвать сама себя.

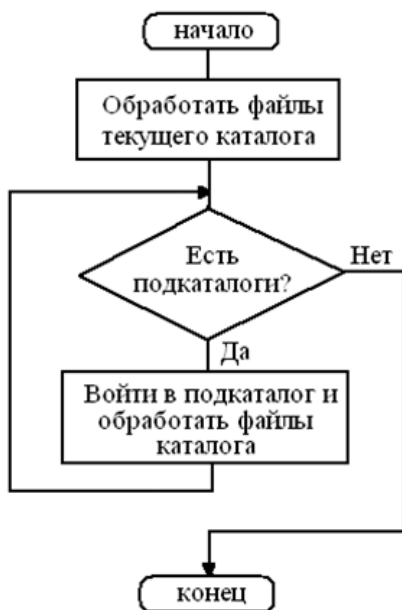


Рис. 4

Различают **цикл с предусловием** и **цикл с постусловием**.

Цикл начинается с проверки логического выражения « P ». Если оно истинно, то выполняется « F », затем все повторяется снова, до тех пор, пока логическое выражение сохраняет значение «истина». Как только оно становится ложным, выполнение операций « F » прекращается и управление передается по программе дальше.

Так как выражение, управляющее циклом, проверяется в самом начале, то в случае, если условие сразу окажется ложным, операторы циклической части « F » могут вообще не выполняться. Операторы циклической части « F » должны изменять

переменную (или переменные), влияющую на значение логического выражения, иначе программа «заиклится» – будет выполняться бесконечно. Рассмотренная циклическая конструкция называется **цикл «пока»**, или **цикл с предусловием** (см. рис. 5).

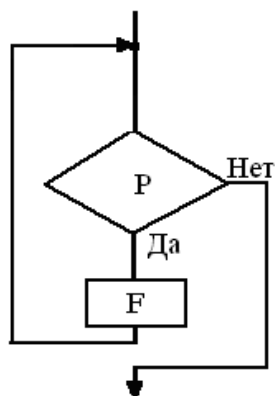


Рис. 5

Существует и иная конструкция цикла, которая предусматривает проверку условия после выполнения команд, встроенных внутрь цикла. Это **цикл с постусловием** (см. рис. 6).

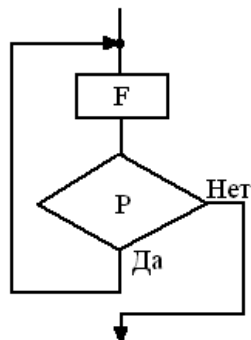


Рис. 6

Циклические структуры можно комбинировать одну с другой – как путем организации их следований, так и путем создания суперпозиций (вложений одной структуры в другую).

Схематические изображения нескольких суперпозиций базовых алгоритмических структур представлены на рис. 7–10.

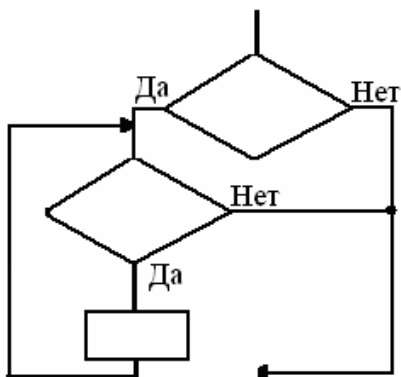


Рис. 7. Алгоритм типа «цикл, вложенный в неполную развилку»

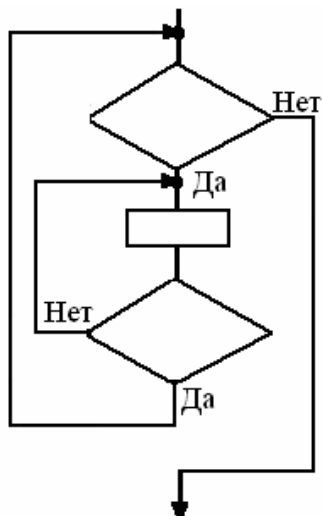


Рис. 8. Алгоритм типа «цикл в цикле»

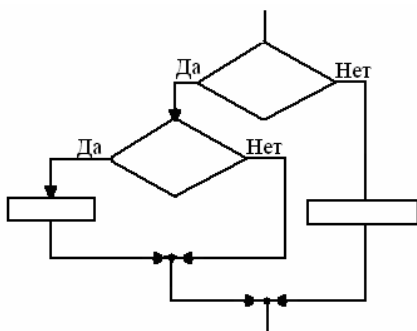


Рис. 9. Алгоритм типа «развилка в развилке»

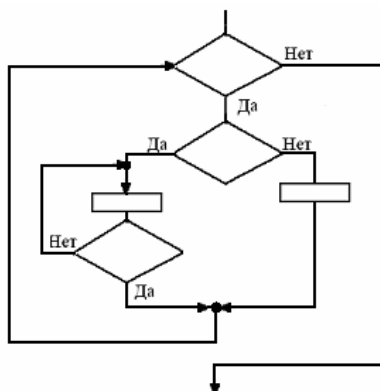


Рис. 10. Иллюстрация трехкратного вложения одной базовой структуры в другую

ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ НА АЛГОРИТМЫ

1. Линейный алгоритм

Пример 1. Дан алгоритм в виде блок-схемы (рис. 11).

Найти A, B, C, D, если изначально:

- а) A=0, B=0, C=5, D=10;
- б) A=0, B=5, C=0, D=10;
- в) A=10, B=20, C=6, D=4;
- г) A=10, B=10, C=4, D=0.

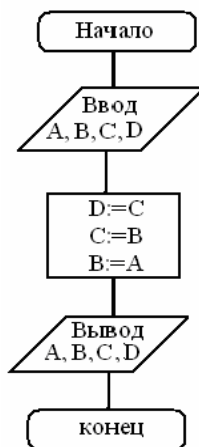


Рис. 11

Результат работы алгоритма определяется с помощью трассировочных таблиц (а, б, в, г):

- а) A=0, B=0, C=5, D=10.

Шаг	1	
Исходные значения	A	0
	B	0
	C	5
	D	10
Результат выполнения	A	0
	B	0
	C	0
	D	5
Вывод значений	0, 0, 0, 5	

б) A=0, B=5, C=0, D=10.

Шаг	1	
Исходные значения	A	0
	B	0
	C	0
	D	10
Результат выполнения	A	0
	B	0
	C	5
	D	0
Вывод значений	0, 0, 5, 0	

в) A=10, B=20, C=6, D=4.

Шаг	1	
Исходные значения	A	10
	B	20
	C	6
	D	4
Результат выполнения	A	10
	B	10
	C	20
	D	6
Вывод значений	10, 10, 20, 6	

г) A=10, B=10, C=4, D=0.

Шаг	1	
Исходные значения	A	10
	B	10
	C	4
	D	0
Результат выполнения	A	10
	B	10
	C	10
	D	4
Вывод значений	10, 10, 10, 4	

2. Разветвляющийся алгоритм

Пример 2. Перед выходным днем папа сказал своему сыну: «Давай спланируем свой завтрашний день. Если будет хорошая погода, то проведем день в лесу. Если же погода будет плохая, то сначала займемся уборкой квартиры, а во второй половине дня сходим в зоопарк».



Рис. 12

Что получится на выходе блок-схемы (рис. 12), если:

- а) погода хорошая;
- б) погода плохая?

Для определения результата воспользуемся трассировочными таблицами (а, б):

- а) погода хорошая:

Шаг	1
Исходные значения	Погода хорошая
Результат выполнения	Прогулка в лесу
Вывод значений	Прогулка в лесу

- б) погода плохая:

Шаг	1
Исходные значения	Погода плохая
Результат выполнения	Уборка квартиры Поход в зоопарк
Вывод значений	Поход в зоопарк

3. Неполное ветвление

Пример 3. Из ряда чисел 15, 16, 17, 18 выписать значения x , удовлетворяющие условию (см. блок-схему на рис. 13).

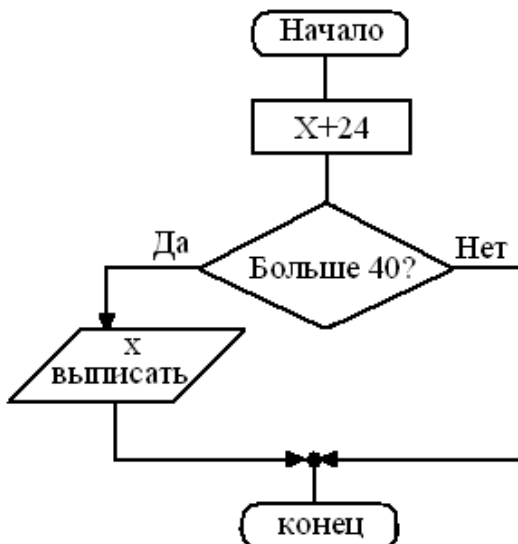


Рис. 13

Используя трассировочную таблицу, получим:

Шаг	1	2	3	4
Исходное значение x	15	16	17	18
Результат выполнения	$15+24$	$16+24$	$17+24$	$18+24$
Тело цикла	$15+24 > 40$ (Нет)	$16+24 > 40$ (Нет)	$17+24 > 40$ (Да)	$18+24 > 40$ (Да)
Вывод x	–	–	17	18

4. Цикл с предусловием

Пример 4. Дана блок-схема (рис. 14). Какое значение будет иметь N на выходе, если:

- а) $S=1,1$; б) $S=2,09$?

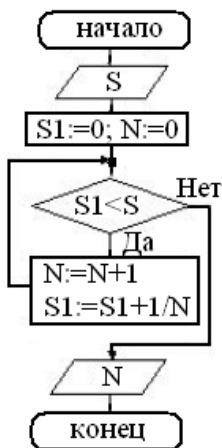


Рис. 14

Для определения результата воспользуемся трассировочными таблицами (а, б):

- а) $S=1,1$:

Шаг	1	2	3
Значение S1	0	1	1,5
Значение N	0	1	2
Тело цикла	$0 < 1,1$ (Да)	$1 < 1,1$ (Да)	$1,5 < 1,1$ (Нет)
Результат выполнения	$N=0+1=1$ $S1=0+1/1=1$	$N=1+1=2$ $S1=1+1/2=1,5$	$N = 2$
Вывод значения N			2

б) $S=2,09$:

Шаг	1	2	3	4	5	6
Значение S_1	0	1	1,5	1,8333	2,0833	2,2833
Значение N	0	1	2	3	4	5
Шаг	$0 < 2,09$ (Да)	$1 < 2,09$ (Да)	$1,5 < 2,09$ (Да)	$1,8333 < 2,09$ (Да)	$2,0833 < 2,09$ (Да)	$2,2833 < 2,09$ (Нет)
Результат выполнения	$N=0+1=1$ $S_1=0+1/1=1$	$N=1+1=2$ $S_1=1+1/2 = 1,5$	$N=2+1=3$ $S_1=1,5+1/3 = 1,8333$	$N=3+1=4$ $S_1=1,8333+1/4 = 2,0833$	$N=4+1=5$ $S_1=2,0833+1/5 = 2,2833$	$N = 5$
Вывод значения N						5

5. Цикл с постусловием

Пример 5. Дана блок-схема (рис. 15). Какое значение будет иметь z на выходе, если: а) $x=2$; б) $x=4$; в) $x=6$?

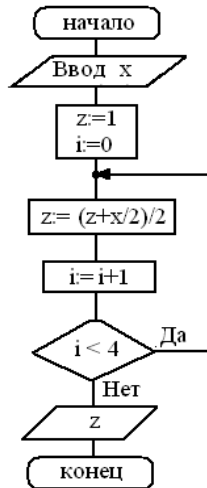


Рис. 15

Для определения результата воспользуемся трассировочными таблицами (а, б, в):

а) $x=2$:

Шаг	1	2	3	4
Начальное значение z	1	1	1	1
Значение i	0	1	2	3
Результат выполнения z	$z = (1+2/2)/2=1$	$z = (1+2/2)/2=1$	$z = (1+2/2)/2=1$	$z = (1+2/2)/2=1$
Результат выполнения i	$i = 0+1=1$	$i = 1+1=2$	$i = 2+1=3$	$i = 3+1=4$
Тело цикла	$1 < 4$ (Да)	$2 < 4$ (Да)	$3 < 4$ (Да)	$4 < 4$ (Нет)
Вывод z				1

б) $x=4$:

Шаг	1	2	3	4
Начальное значение z	1	1,5	1,75	1,875
Значение i	0	1	2	3
Результат выполнения z	$z = (1+4/2)/2 = 1,5$	$z = (1,5+4/2)/2 = 1,75$	$z = (1,75+4/2)/2 = 1,875$	$z = (1,875+4/2)/2 = 1,9375$
Результат выполнения i	$i = 0+1=1$	$i = 1+1=2$	$i = 2+1=3$	$i = 3+1=4$
Тело цикла	$1 < 4$ (Да)	$2 < 4$ (Да)	$3 < 4$ (Да)	$4 < 4$ (Нет)
Вывод z				1,9375

в) $x=6$:

Шаг	1	2	3	4
Начальное значение z	1	2	2,5	2,75
Значение i	0	1	2	3
Результат выполнения z	$z = (1+6/2)/2 = 2$	$z = (2+6/2)/2 = 2,5$	$z = (2,5+6/2)/2 = 2,75$	$z = (2,75+6/2)/2 = 2,875$
Результат выполнения i	$i = 0+1=1$	$i = 1+1=2$	$i = 2+1=3$	$i = 3+1=4$
Тело цикла	$1 < 4$ (Да)	$2 < 4$ (Да)	$3 < 4$ (Да)	$4 < 4$ (Нет)
Вывод z				2,875

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

1. Задачи на линейный алгоритм

Задача 1. Реализован некоторый алгоритм в виде блок-схемы (рис. 16). Найти A, B на выходе блок-схемы, если изначально:

- а) $A=0, B=0$;
- б) $A=0, B=5$;
- в) $A=10, B=20$;
- г) $A=10, B=10$.

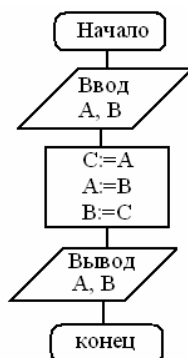


Рис. 16

Задача 2. Даны длины двух катетов (a, b) прямоугольного треугольника. Определить периметр этого треугольника (P) (см. блок-схему на рис. 17), если:

- а) $a=3, b=4$;
- б) $a=0, b=3$;
- в) $a=6, b=8$;
- г) $a=9, b=12$.

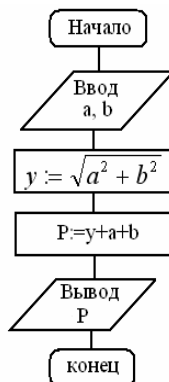


Рис. 17

Задача 3. Реализован некоторый алгоритм в виде блок-схемы (рис. 18). По данной блок-схеме вычислить S, если:

а) $a=1, b=2, c=3$;

б) $a=9, b=0, c=1$;

в) $a=5, b=6, c=9$.

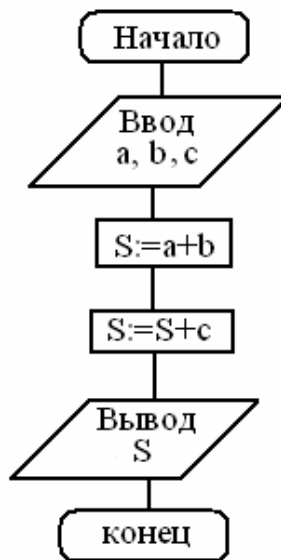


Рис. 18

Задача 4. Дана блок-схема (рис.19). Начальные условия: $a=8, b=2$. Тогда после исполнения алгоритма значение переменной g будет равно ...

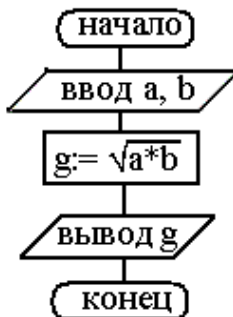


Рис. 19

2. Задачи на разветвляющийся алгоритм

Задача 5. Вычислить значение функции: $y = \begin{cases} \sqrt{x}, & x \geq 0 \\ x^2, & x < 0 \end{cases}$

(см. блок-схему на рис. 20), если: а) $x=0$; б) $x=1$; в) $x=-5$.

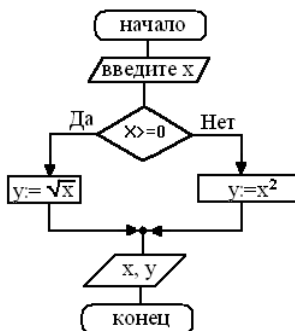


Рис. 20

Задача 6. Используя блок-схему (рис. 21), найти корни уравнения $ax^2 + bx + c = 0$ ($a \neq 0$), если:

а) $a=1$, $b=2$, $c=-3$;

б) $a=1$, $b=4$, $c=5$;

в) $a=3$, $b=-8$, $c=3$.

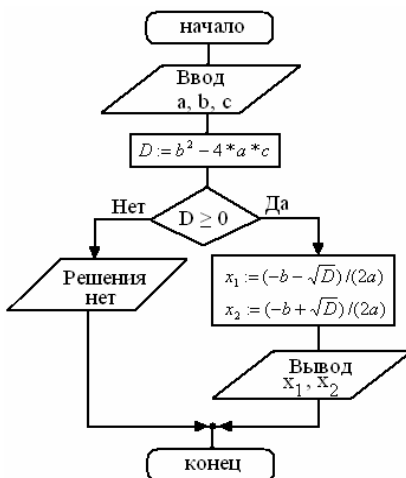


Рис. 21

Задача 7. Реализован некоторый алгоритм в виде блок-схемы (рис. 22). Что получится на выходе блок-схемы, если:

- а) $x=0, y=1$; б) $x=2, y=4$; в) $x=6, y=0$?

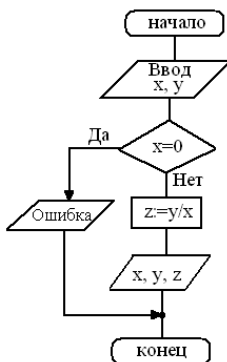


Рис. 22

Задача 8. На блок-схеме (рис. 23) представлен алгоритм вычисления стоимости покупки с учетом скидки, где a – цена, b – количество, s – сумма. Какой будет результат на выходе блок-схемы, если:

- а) $a=50, b=8$;
 б) $a=200, b=5$;
 в) $a=300, b=1$;
 г) $a=800, b=4$?

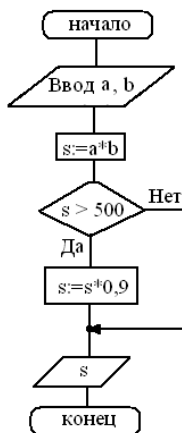


Рис. 23

3. Задачи на цикл с предусловием

Задача 9. На блок-схеме (рис. 24) представлен алгоритм Евклида, определяющий наибольший общий делитель (НОД) для двух натуральных чисел A и B . Найти A на выходе блок-схемы, если:

- а) $A=5, B=10$;
- б) $A=8, B=8$;
- в) $A=12, B=4$;
- г) $A=16, B=36$.

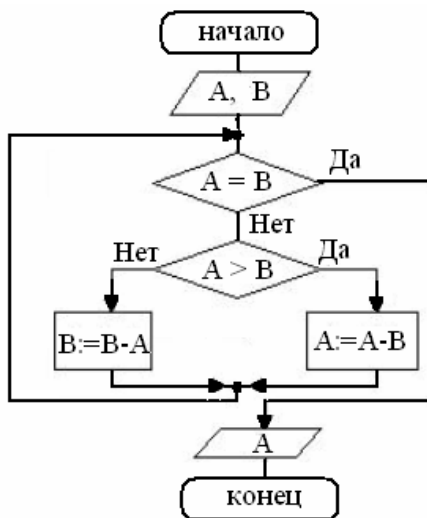


Рис. 24

Задача 10. Реализован некоторый алгоритм в виде блок-схемы (рис. 25). Что получится на выходе блок-схемы, если:

- а) $n=2$; б) $n=0,5$?

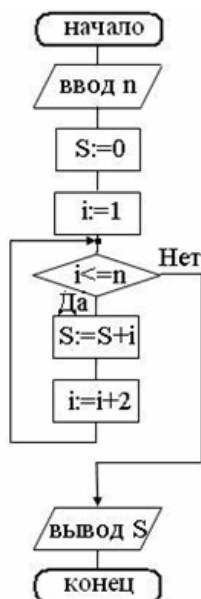


Рис. 25

Задача 11. Дана блок-схема (рис. 26). Тогда после исполнения алгоритма переменная i примет значение ...

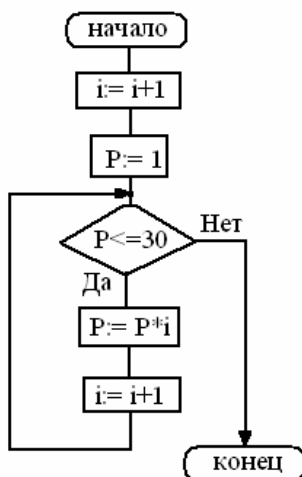


Рис. 26

4. Задачи на цикл с постусловием

Задача 12. Реализован некоторый алгоритм в виде блок-схемы (рис. 27). Что получится на выходе блок-схемы, если:

- а) $Q=2$;
- б) $Q=0,5$?

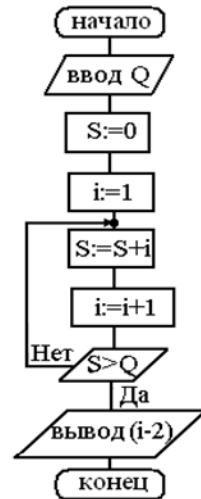


Рис. 27

Задача 13. Дана блок-схема (рис. 28). Тогда после исполнения алгоритма значение переменной S равно ...

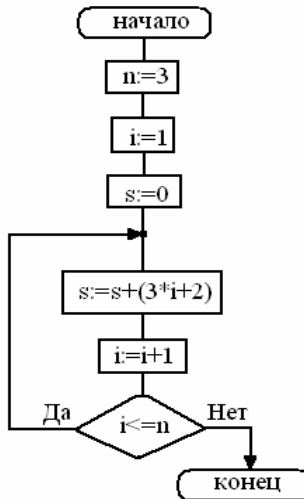


Рис. 28

5. Дополнительные задачи

Задача 14. Дан алгоритм нахождения одного из значений

функции $y = \begin{cases} x^2 + 1, & x < 0 \\ 2x + 1, & 0 \leq x \leq 1 \\ 4x - 1, & x > 1 \end{cases}$ (рис. 29). Какое значение пере-

менная y будет иметь на выходе блок-схемы, если:

а) $x=10$; б) $x=-5$; в) $x=0,5$?

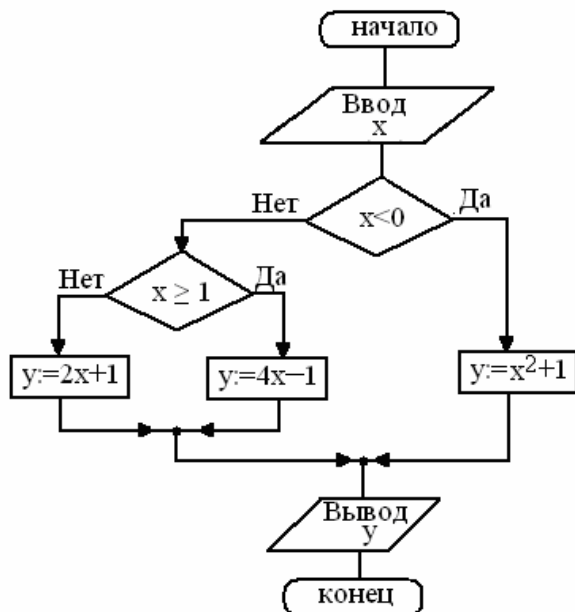


Рис. 29

Задача 15. Реализован некоторый алгоритм в виде блок-схемы (рис. 30). Какой будет результат на выходе блок-схемы, если:

- а) $a = 0$;
- б) $a = -3$;
- в) $a = 3$?

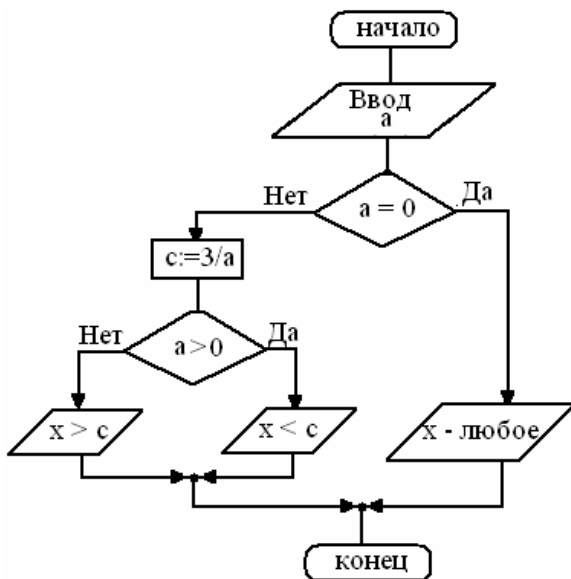


Рис. 30

Задача 16. Реализован некоторый алгоритм в виде блок-схемы (рис. 31). Что получится на выходе блок-схемы, если:

- а) $a=0$, $b=0$;
- б) $a=5$, $b=10$;
- в) $a=2$, $b=0$?



Рис. 31

Задача 17. Реализован некоторый алгоритм в виде блок-схемы (рис. 32). Что получится на выходе блок-схемы, если:

- а) $a = 0$, $b = 0$;
- б) $a = 5$, $b = 10$;
- в) $a = 0$, $b = 5$;
- г) $a = -10$, $b = 20$?

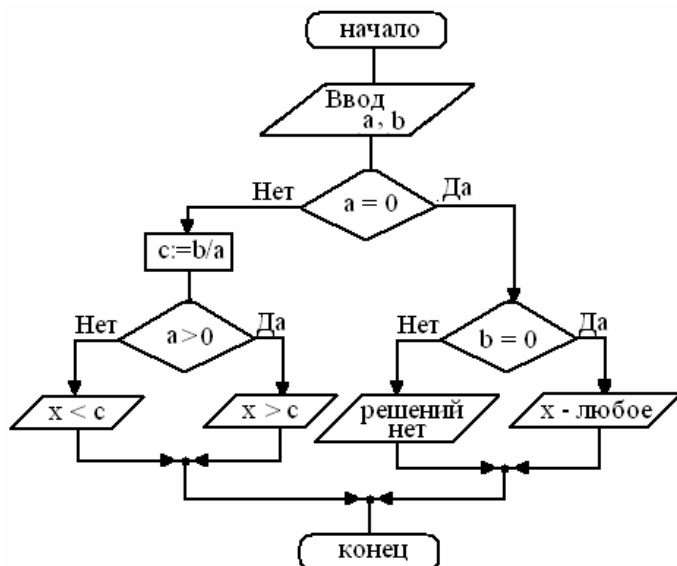


Рис. 32

Задача 18. Реализован некоторый алгоритм в виде блок-схемы (рис. 33). Вывести значение переменной *min*, если:

а) $a=5, b=8, c=3$;

б) $a=9, b=2, c=4$;

в) $a=4, b=2, c=7$;

г) $a=5, b=5, c=1$.

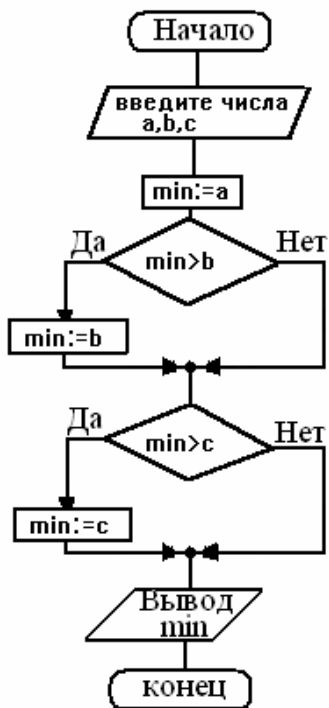


Рис. 33

Задача 19. В блок-схеме реализован алгоритм начисления зарплаты (рис. 34). Какая будет зарплата у сотрудника (ZP), если стаж работы (ST) его составляет:

- а) ST = 4;
- б) ST = 9;
- в) ST = 15;
- г) ST = 20?

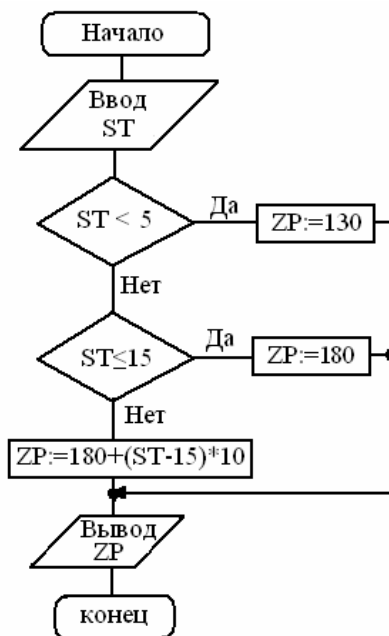


Рис. 34

Задача 20. Вычислить значение y по одной из формул

$$y = \begin{cases} X + A, & \text{если } X < 10 \\ X + B, & \text{если } 10 \leq X < 23 \\ X + A, & \text{если } 23 < X \end{cases} \quad (\text{рис. 35}), \text{ если:}$$

- а) $X=5, A=2, B=3$;
- б) $X=10, A=2, B=3$;
- в) $X=15, A=2, B=3$;
- г) $X=23, A=2, B=3$.

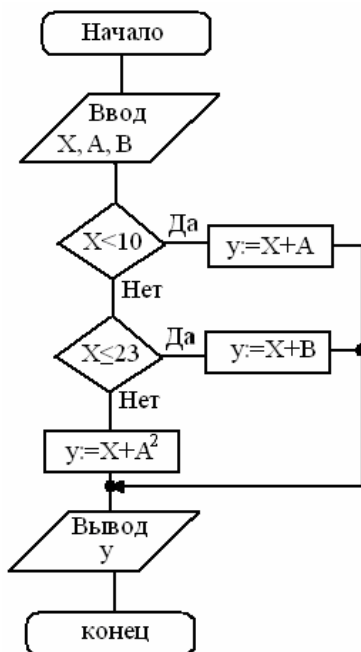


Рис. 35

Задача 21. Реализован некоторый алгоритм в виде блок-схемы (рис. 36). Какой результат получится на выходе блок-схемы, если:

- а) $a=19$ см, $b=27$ см, $c=27$ см;
- б) $a=20$ см, $b=25$ см, $c=27$ см;
- в) $a=5$ см, $b=10$ см, $c=10$ см;
- г) $a=8$ см, $b=13$ см, $c=5$ см?

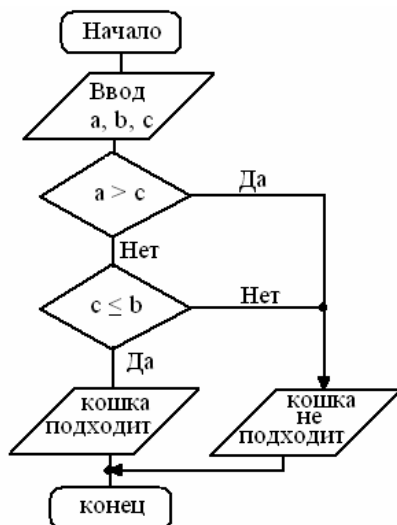


Рис. 36

Задача 22. Реализован некоторый алгоритм в виде блок-схемы (рис. 37). Какой будет результат на выходе блок-схемы, если:

- а) $A=10, B=5, C=11, X=6, Y=15$;
- б) $A=5, B=10, C=4, X=5, Y=3$;
- в) $A=2, B=5, C=8, X=2, Y=4$;
- г) $A=9, B=1, C=4, X=3, Y=7$?

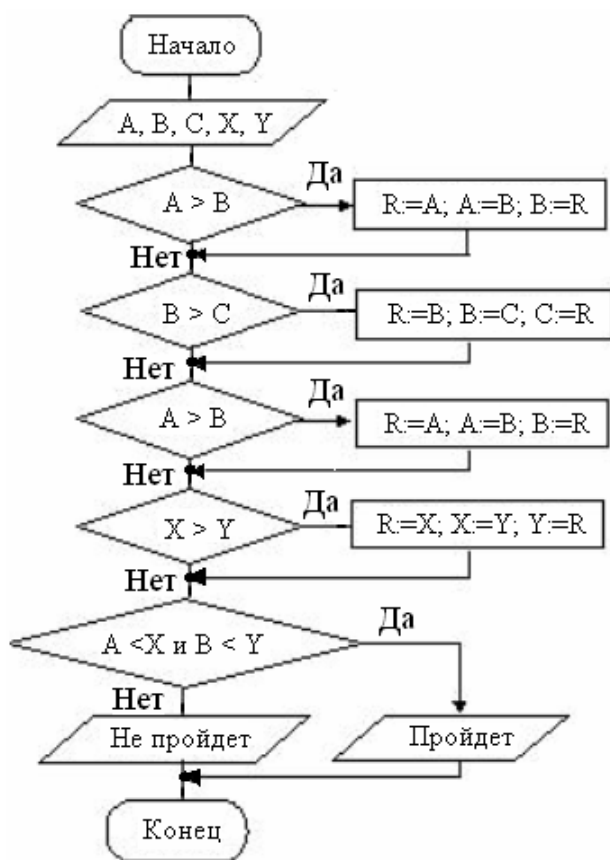


Рис. 37

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Что такое алгоритм?
2. Назовите основные свойства алгоритма.
3. Какие существуют формы представления алгоритма?
Кратко охарактеризуйте их.
4. Чем отличается графическая форма представления алгоритма от других форм?
5. С помощью какой фигуры изображается этап вычисления? Проверка условия? Вывод данных?
6. Какие преимущества дает блок-схема?
7. Назовите три основных структуры алгоритмов.
8. В чем отличие полного ветвления от неполного?
9. Какой цикл называется итерационным? рекурсивным?
10. В чем отличие цикла с предусловием от цикла с постусловием?
11. Что понимается под суперпозицией?

1. Алгоритм дегеніміз не?
2. Алгоритмнің негізгі қасиеттері қандай?
3. Алгоритмді ұсынудың қандай формалары бар?
Оларды қысқаша сипаттаңыз.
4. Ал бейнелеудің графикалық формасы несімен ерекшеленеді-
басқа формалардан горитма?
5. Қай фигураның көмегімен есептеу кезеңі бейнеленген-
сызық? Шартты тексеру? Деректерді шығару?
6. Схема қандай артықшылықтар береді?
7. Алгоритмдердің үш негізгі құрылымын атаңыз.
8. Толық тармақталу мен толық емес тармақталудың
айырмашылығы неде?
9. Қандай цикл итеративті деп аталады? рекурсивті?
10. Айырмашылығы цикла с предусловием желтоқсандағы цикл
бойынша-
Шартпен?
11. Суперпозиция дегеніміз не?