

Глава 7. Алгоритмы. Алгоритмизация. Алгоритмические языки

7.1. Что такое алгоритм?

Понятие алгоритма такое же основополагающее для информатики, как и понятие информации. Именно поэтому важно в нем разобраться.

Название "алгоритм" произошло от латинской формы имени величайшего среднеазиатского математика **Мухаммеда ибн Муса ал-Хорезми** (Alhorithmi), жившего в 783—850 гг. В своей книге "Об индийском счете" он изложил правила записи натуральных чисел с помощью арабских цифр и правила действий над ними "столбиком", знакомые теперь каждому школьнику. В XII веке эта книга была переведена на латынь и получила широкое распространение в Европе.

Человек ежедневно встречается с необходимостью следовать тем или иным правилам, выполнять различные инструкции и указания. Например, переходя через дорогу на перекрестке без светофора надо сначала посмотреть направо. Если машин нет, то перейти полдороги, а если машины есть, ждать, пока они пройдут, затем перейти полдороги. После этого посмотреть налево и, если машин нет, то перейти дорогу до конца, а если машины есть, ждать, пока они пройдут, а затем перейти дорогу до конца.

В математике для решения типовых задач мы используем определенные правила, описывающие последовательности действий. Например, правила сложения дробных чисел, решения квадратных уравнений и т. д. Обычно любые инструкции и правила представляют собой последовательность действий, которые необходимо выполнить в определенном порядке. Для решения задачи надо знать, что дано, что следует получить и какие действия и в каком порядке следует для этого выполнить. Предписание, определяющее порядок выполнения действий над данными с целью получения искоемых результатов, и есть алгоритм.

Алгоритм — заранее заданное понятное и точное предписание возможному исполнителю совершить определенную последовательность действий для получения решения задачи за конечное число шагов.

Это — не определение в математическом смысле слова, а, скорее, **описание** интуитивного понятия алгоритма, раскрывающее его сущность.

Понятие алгоритма является не только одним из главных понятий математики, но одним из главных понятий современной науки. Более того, с наступлением эры информатики алгоритмы становятся одним из важнейших факторов цивилизации [56].

7.2. Что такое "Исполнитель алгоритма"?

Исполнитель алгоритма — это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнять действия, предписываемые алгоритмом.

Исполнителя характеризуют:

- среда;
- элементарные действия;
- система команд;
- отказы.

Среда (или обстановка) — это "место обитания" исполнителя. Например, для исполнителя Робота из школьного учебника [1] среда — это бесконечное клеточное поле. Стены и закрашенные клетки тоже часть среды. А их расположение и положение самого Робота задают конкретное **состояние среды**.

Система команд. Каждый исполнитель может выполнять команды только из некоторого строго заданного списка — системы команд исполнителя. Для каждой команды должны быть заданы **условия применимости** (в каких состояниях среды может быть выполнена команда) и описаны **результаты выполнения команды**. Например, команда Робота "вверх" может быть выполнена, если выше Робота нет стены. Ее результат — смещение Робота на одну клетку вверх.

После вызова команды исполнитель совершает соответствующее **элементарное действие**.

Отказы исполнителя возникают, если команда вызывается при недопустимом для нее состоянии среды.

Обычно исполнитель ничего не знает о цели алгоритма. Он выполняет все полученные команды, не задавая вопросов "почему" и "зачем".

В информатике универсальным исполнителем алгоритмов является **компьютер**.

7.3. Какими свойствами обладают алгоритмы?

Основные свойства алгоритмов следующие:

1. Понятность для исполнителя — исполнитель алгоритма должен понимать, как его выполнять. Иными словами, имея алгоритм и произвольный вариант исходных данных, исполнитель должен знать, как надо действовать для выполнения этого алгоритма.

2. Дискретность (прерывность, раздельность) — алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов).

3. Определенность — каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

4. Результативность (или конечность) состоит в том, что за конечное число шагов алгоритм либо должен приводить к решению задачи, либо после конечного числа шагов останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов.

5. Массовость означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

7.4. В какой форме записываются алгоритмы?

На практике наиболее распространены следующие формы представления алгоритмов:

- **словесная** (запись на естественном языке);
- **графическая** (изображения из графических символов);
- **псевдокоды** (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- **программная** (тексты на языках программирования).

7.5. Что такое словесный способ записи алгоритмов?

Словесный способ записи алгоритмов представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке.

Например. Записать алгоритм нахождения **наибольшего общего делителя (НОД)** двух натуральных чисел (алгоритм Эвклида).

Алгоритм может быть следующим:

1. задать два числа;
2. если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;
3. определить большее из чисел;
4. заменить большее из чисел разностью большего и меньшего из чисел;
5. повторить алгоритм с шага 2.

Описанный алгоритм применим к любым натуральным числам и должен приводить к решению поставленной задачи. Убедитесь в этом самостоятельно, определив с помощью этого алгоритма наибольший общий делитель чисел 125 и 75.

Словесный способ не имеет широкого распространения, так как такие описания:

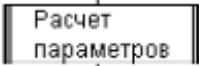

- строго не формализуемы;
- страдают многословностью записей;
- допускают неоднозначность толкования отдельных предписаний.

7.6. Что такое графический способ записи алгоритмов?

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным.

При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.

Такое графическое представление называется схемой алгоритма или **блок-схемой**. В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде **блочного символа**. Блочные символы соединяются **линиями переходов**, определяющими очередность выполнения действий. В таблице приведены наиболее часто употребляемые символы.

Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Вычислительное действие или последовательность действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисления по подпрограмме, стандартной подпрограмме
Ввод-вывод		Ввод-вывод в общем виде
Пуск-останов		Начало, конец алгоритма, вход и выход в подпрограмму
Документ		Вывод результатов на печать

Блок **"процесс"** применяется для обозначения действия или последовательности действий, изменяющих значение, форму представления или размещения данных. Для улучшения наглядности схемы несколько отдельных блоков обработки можно объединять в один блок. Представление отдельных операций достаточно свободно.

Блок **"решение"** используется для обозначения переходов управления по условию. В каждом блоке "решение" должны быть указаны вопрос, условие или сравнение, которые он определяет.

Блок **"модификация"** используется для организации циклических конструкций. (Слово модификация означает видоизменение, преобразование). Внутри блока записывается параметр цикла, для которого указываются его начальное значение, граничное условие и шаг изменения значения параметра для каждого повторения.

Блок **"предопределенный процесс"** используется для указания обращений к вспомогательным алгоритмам, существующим автономно в виде некоторых самостоятельных модулей, и для обращений к библиотечным подпрограммам.

7.7. Что такое псевдокод?

Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов.

Псевдокод занимает промежуточное место между естественным и формальным языками. С одной стороны, он близок к обычному естественному языку, поэтому алгоритмы могут на нем записываться и читаться как обычный текст. С другой стороны, в псевдокоде используются некоторые формальные конструкции и математическая символика, что приближает запись алгоритма к общепринятой математической записи.

В псевдокоде не приняты строгие синтаксические правила для записи команд, присущие формальным языкам, что облегчает запись алгоритма на стадии его проектирования и дает возможность использовать более широкий набор команд, рассчитанный на абстрактного исполнителя.

Однако в псевдокоде обычно **имеются некоторые конструкции, присущие формальным языкам**, что облегчает переход от записи на псевдокоде к записи алгоритма на формальном языке. В частности, в псевдокоде, так же, как и в формальных языках, есть служебные слова, смысл которых определен раз и навсегда. Они выделяются в печатном тексте жирным шрифтом, а в рукописном тексте подчеркиваются.

Единого или формального определения псевдокода не существует, поэтому возможны различные псевдокоды, отличающиеся набором служебных слов и основных (базовых) конструкций.

Примером псевдокода является школьный алгоритмический язык в русской нотации (школьный АЯ), описанный в учебнике А.Г. Кушниренко и др. "Основы информатики и вычислительной техники", 1991. Этот язык в дальнейшем мы будем называть просто "алгоритмический язык".

7.8. Как записываются алгоритмы на школьном алгоритмическом языке?

Основные служебные слова

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

Общий вид алгоритма:

```
алг название алгоритма (аргументы и результаты)
  дано условия применимости алгоритма
  надо цель выполнения алгоритма
нач описание промежуточных величин
| последовательность команд (тело алгоритма)
кон
```

Часть алгоритма от слова **алг** до слова **нач** называется **заголовком**, а часть, заключенная между словами **нач** и **кон** — **телом** алгоритма.

В предложении **алг** после названия алгоритма в круглых скобках указываются **характеристики** (**арг**, **рез**) и **тип значения** (**цел**, **вещ**, **сим**, **лит** или **лог**) всех **входных** (аргументы) и **выходных** (результаты) переменных. При описании массивов (таблиц) используется служебное слово **таб**, дополненное **граничными парами** по каждому индексу элементов массива.

Примеры предложений **алг**:

```
алг Объем и площадь цилиндра ( арг вещ R, H, рез вещ V, S )
алг Корни КвУр ( арг вещ a, b, c, рез вещ x1, x2, рез лит t )
алг Исключить элемент ( арг цел N, арг рез вещ таб A[1:N] )
алг Диагональ ( арг цел N, арг цел таб A[1:N, 1:N], рез лит Otvet )
```

Предложения **дано** и **надо** не обязательны. В них рекомендуется записывать утверждения, описывающие состояние среды исполнителя алгоритма, например:

1. **алг** Замена (**арг** **лит** Str1, Str2, **арг** **рез** **лит** Text)
2. **дано** | длины подстрок Str1 и Str2 совпадают
3. **надо** | всюду в строке Text подстрока Str1 заменена на Str2
- 4.
5. **алг** Число максимумов (**арг** **цел** N, **арг** **вещ** **таб** A[1:N], **рез** **цел** K)
6. **дано** | N>0
7. **надо** | K — число максимальных элементов в таблице A
- 8.
9. **алг** Сопротивление (**арг** **вещ** R1, R2, **арг** **цел** N, **рез** **вещ** R)
10. **дано** | N>5, R1>0, R2>0
11. **надо** | R — сопротивление схемы

Здесь в предложениях **дано** и **надо** после знака "|" записаны **комментарии**. Комментарии можно помещать в конце любой строки. Они не обрабатываются транслятором, но существенно облегчают понимание алгоритма.

Команды школьного АЯ

Команда присваивания. Служит для вычисления выражений и присваивания их значений переменным. Общий вид: **A := B**, где знак "!=" означает команду **заменить** прежнее значение переменной, стоящей в левой части, на вычисленное значение выражения, стоящего в правой части.

Например, `a := (b+c) * sin(Pi/4); i := i+1.`

Команды ввода и вывода.

- **ввод** имени переменных
- **вывод** имени переменных, выражения, тексты.

Команды если и выбор. Применяют для организации [ветвлений](#).

Команды для и пока. Применяют для организации [циклов](#).

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 + ... + 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
```

7.9. Что такое базовые алгоритмические структуры?

Алгоритмы можно представлять как некоторые структуры, состоящие из отдельных **базовых** (т.е. основных) **элементов**. Естественно, что при таком подходе к алгоритмам изучение основных принципов их конструирования должно начинаться с изучения этих базовых элементов. Для их описания будем использовать язык схем алгоритмов и школьный алгоритмический язык.

Логическая структура любого алгоритма может быть представлена комбинацией трех базовых структур: следование, ветвление, цикл.

Характерной особенностью базовых структур является наличие в них **одного входа и одного выхода**.

1. Базовая структура "следование". Образуется последовательностью действий, следующих одно за другим:

действие 1
действие 2
.....
действие n



2. Базовая структура "ветвление". Обеспечивает в зависимости от результата проверки условия (**да** или **нет**) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к **общему выходу**, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран. Структура **ветвление** существует в четырех основных вариантах:

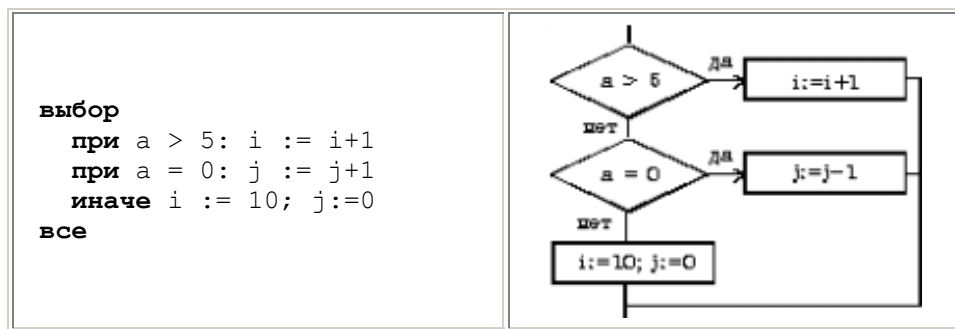
- если—то;
- если—то—иначе;
- выбор;
- выбор—иначе.

Школьный алгоритмический язык	Язык блок-схем
1. если—то	
<pre> если условие то действия все </pre>	
2. если—то—иначе	
<pre> если условие то действия 1 иначе действия 2 все </pre>	

3. выбор	
<p>выбор при условие 1: действия 1 при условие 2: действия 2 при условие N: действия N все</p>	
4. выбор—иначе	
<p>выбор при условие 1: действия 1 при условие 2: действия 2 при условие N: действия N иначе действия N+1 все</p>	

Примеры структуры **ветвление**

Школьный алгоритмический язык	Язык блок-схем
<pre> если x > 0 то y := sin(x) все </pre>	
<pre> если a > b то a := 2*a; b := 1 иначе b := 2*b все </pre>	
<pre> выбор при n = 1: y := sin(x) при n = 2: y := cos(x) при n = 3: y := 0 все </pre>	



3. Базовая структура "цикл". Обеспечивает многократное выполнение некоторой совокупности действий, которая называется телом цикла. Основные разновидности циклов представлены в таблице:

Школьный алгоритмический язык	Язык блок-схем
<p align="center">Цикл типа пока.</p> <p align="center">Предписывает выполнять тело цикла до тех пор, пока выполняется условие, записанное после слова пока.</p>	
<pre> нц пока условие тело цикла (последовательность действий) кц </pre>	
<p align="center">Цикл типа для.</p> <p align="center">Предписывает выполнять тело цикла для всех значений некоторой переменной (параметра цикла) в заданном диапазоне.</p>	
<pre> нц для i от i1 до i2 тело цикла (последовательность действий) кц </pre>	

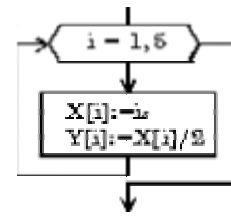
Примеры структуры **цикл**

Школьный алгоритмический язык	Язык блок-схем
<pre> нц пока i <= 5 S := S+A[i] i := i+1 кц </pre>	

```

нц для i от 1 до 5
  X[i] := i*i*i
  Y[i] := X[i]/2
кц

```



7.10. Какие циклы называют итерационными?

Особенностью итерационного цикла является то, что число повторений операторов тела цикла заранее неизвестно. Для его организации используется цикл типа **пока**. Выход из итерационного цикла осуществляется в случае выполнения заданного условия.

На каждом шаге вычислений происходит **последовательное приближение к искомому результату и проверка условия достижения последнего**.

Пример. Составить алгоритм вычисления бесконечной суммы

$$S = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots (-1)^{i-1} \frac{x^i}{i} + \dots$$

с заданной точностью ε (для данной знакопеременной бесконечной суммы требуемая точность будет достигнута, когда очередное слагаемое станет по абсолютной величине меньше ε).

Вычисление сумм — типичная циклическая задача. Особенностью же нашей конкретной задачи является то, что число слагаемых (а, следовательно, и число повторений тела цикла) заранее неизвестно. Поэтому выполнение цикла должно завершиться в момент достижения требуемой точности.

При составлении алгоритма нужно учесть, что знаки слагаемых чередуются и степень числа x в числителях слагаемых возрастает.

Решая эту задачу "в лоб" путем вычисления на каждом i -ом шаге частичной суммы

$$S := S + ((-1)^{(i-1)}) * (x^i) / i ,$$

мы получим очень неэффективный алгоритм, требующий выполнения большого числа операций. Гораздо лучше организовать вычисления следующим образом: если обозначить числитель какого-либо слагаемого буквой p , то у следующего слагаемого числитель будет равен $-p*x$ (знак минус обеспечивает чередование знаков слагаемых), а само слагаемое m будет равно p/i , где i — номер слагаемого.

Сравните эти два подхода по числу операций.

Алгоритм на школьном АЯ	Блок-схема алгоритма
<pre> алг Сумма (арг вещ x, Eps, рез вещ S) дано $0 < x < 1$ надо $S = x - x^2/2 + x^3/3 - \dots$ нач цел i, вещ m, p ввод x, Eps S := 0; i := 1 начальные значения m := 1; p := -1 нц пока abs(m) > Eps p := -p*x p - числитель очередного слагаемого m := p/i m - очередное слагаемое S := S + m S - частичная сумма i := i + 1 i - номер очередного слагаемого кц вывод S кон </pre>	<pre> graph TD Start([начало]) --> Input[/x, Eps/] Input --> Init[S:=0; i:=1; m:=1; p:=-1] Init --> Decision{abs(m) > Eps} Decision -- да --> Loop[p:=-p*x; m:=p/i; S:=S+m; i:=i+1] Loop --> Decision Decision -- нет --> Output[/S/] Output --> End([конец]) </pre>

Алгоритм, в состав которого входит итерационный цикл, называется **итерационным алгоритмом**. Итерационные алгоритмы используются при реализации итерационных численных методов.

В итерационных алгоритмах необходимо обеспечить обязательное достижение условия выхода из цикла (**сходимость итерационного процесса**). В противном случае произойдет "**зацикливание**" алгоритма, т.е. не будет выполняться основное свойство алгоритма — результативность.

7.11. Что такое вложенные циклы?

Возможны случаи, когда внутри тела цикла необходимо повторять некоторую последовательность операторов, т. е. организовать внутренний цикл. Такая структура получила название цикла в цикле или **вложенных циклов**. Глубина вложения циклов (то есть количество вложенных друг в друга циклов) может быть различной.

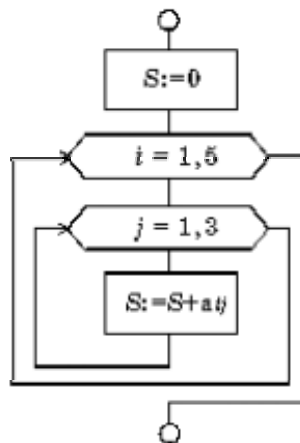
При использовании такой структуры для экономии машинного времени необходимо выносить из внутреннего цикла во внешний все операторы, которые не зависят от параметра внутреннего цикла.

Пример вложенных циклов **для**

Вычислить сумму элементов заданной матрицы A(5,3).

Матрица A

	1	2	3
1			
2			
3			
4			
5			

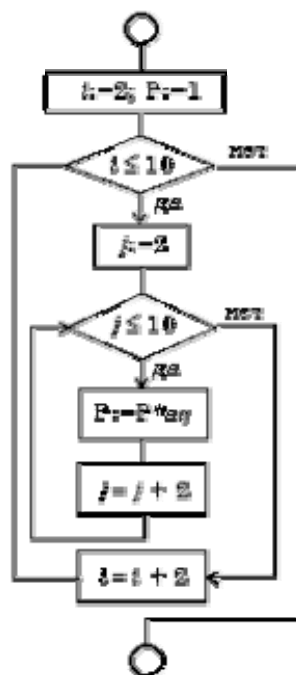


```

S := 0;
нц для i от 1 до 5
  нц для j от 1 до 3
    S:=S+A[i,j]
  кц
кц
  
```

Пример вложенных циклов пока

Вычислить произведение тех элементов заданной матрицы A(10,10), которые расположены на пересечении четных строк и четных столбцов.



```

i:=2; P:=1
нц пока i <= 10
  j:=2
  нц пока j <= 10
    P:=P*A[i,j]
    j:=j+2
  кц
  i:=i+2
кц
  
```

7.12. Чем отличается программный способ записи алгоритмов от других?

При записи алгоритма в словесной форме, в виде блок-схемы или на псевдокоде допускается определенный произвол при изображении команд. Вместе с тем такая запись точна настолько, что позволяет человеку понять суть дела и исполнить алгоритм.

Однако на практике в качестве исполнителей алгоритмов используются специальные автоматы — компьютеры. Поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на понятном ему языке. И здесь на первый план

выдвигается необходимость точной записи команд, не оставляющей места для произвольного толкования их исполнителем.

Следовательно, **язык для записи алгоритмов должен быть формализован**. Такой язык принято называть **языком программирования**, а запись алгоритма на этом языке — **программой для компьютера**.

7.13. Что такое уровень языка программирования?

В настоящее время в мире существует несколько сотен реально используемых языков программирования. Для каждого есть своя область применения.

Любой алгоритм, как мы знаем, есть последовательность предписаний, выполнив которые можно за конечное число шагов перейти от исходных данных к результату. В зависимости от степени детализации предписаний обычно определяется уровень языка программирования — чем меньше детализация, тем выше уровень языка.

По этому критерию можно выделить следующие уровни языков программирования:

- [машинные](#);
- машинно-ориентированные ([ассемблеры](#));
- машинно-независимые (языки высокого уровня).

Машинные языки и машинно-ориентированные языки — это языки **низкого уровня**, требующие указания мелких деталей процесса обработки данных. Языки же **высокого уровня** имитируют естественные языки, используя некоторые слова разговорного языка и общепринятые математические символы. Эти языки более удобны для человека.

Языки высокого уровня делятся на:

- **процедурные (алгоритмические)** (Basic, Pascal, C и др.), которые предназначены для однозначного описания алгоритмов; для решения задачи процедурные языки требуют в той или иной форме явно записать процедуру ее решения;
- **логические** (Prolog, Lisp и др.), которые ориентированы не на разработку алгоритма решения задачи, а на систематическое и формализованное описание задачи с тем, чтобы решение следовало из составленного описания;
- **объектно-ориентированные** (Object Pascal, C++, Java и др.), в основе которых лежит **понятие объекта, сочетающего в себе данные и действия над ними**. Программа на объектно-ориентированном языке, решая некоторую задачу, по сути описывает часть мира, относящуюся к этой задаче. Описание действительности в форме системы взаимодействующих объектов естественнее, чем в форме взаимодействующих процедур.

7.14. Какие у машинных языков достоинства и недостатки?

Каждый компьютер имеет свой машинный язык, то есть свою совокупность машинных команд, которая отличается количеством адресов в команде, назначением информации, задаваемой в адресах, набором операций, которые может выполнить машина и др.

При программировании на машинном языке программист может держать под своим контролем каждую команду и каждую ячейку памяти, использовать все возможности имеющихся машинных операций.

Но процесс написания программы на машинном языке очень **трудоемкий и утомительный**. Программа получается **громоздкой, труднообозримой, ее трудно отлаживать, изменять и развивать**.

Поэтому в случае, когда нужно иметь эффективную программу, в максимальной степени учитывающую специфику конкретного компьютера, вместо машинных языков используют близкие к ним машинно-ориентированные языки (ассемблеры).

7.15. Что такое язык ассемблера?

Язык ассемблера — это машинно-зависимый язык низкого уровня, в котором короткие мнемонические имена соответствуют отдельным машинным командам. Используется для представления в удобочитаемой форме программ, записанных в машинном коде.

Язык ассемблера позволяет программисту пользоваться **текстовыми мнемоническими** (то есть легко запоминаемыми человеком) **кодами**, по своему усмотрению **присваивать символические имена регистрам компьютера и памяти, а также задавать удобные для себя способы адресации**. Кроме того, он позволяет использовать различные системы счисления (например, десятичную или шестнадцатеричную) для представления числовых констант, использовать в программе комментарии и др.

Программы, написанные на языке ассемблера, требуют значительно меньшего объема памяти и времени выполнения. Знание программистом языка ассемблера и машинного кода дает ему понимание архитектуры машины. Несмотря на то, что большинство специалистов в области программного обеспечения разрабатывают программы на языках высокого уровня, таких, как Object Pascal или C, наиболее мощное и эффективное программное обеспечение полностью или частично написано на языке ассемблера.

Языки высокого уровня были разработаны для того, чтобы освободить программиста от учета технических особенностей конкретных компьютеров, их архитектуры. В противоположность этому, язык ассемблера разработан с целью учесть конкретную специфику процессора. Следовательно, для того, чтобы написать программу на языке ассемблера для конкретного компьютера, важно знать его архитектуру [57].

В качестве примера приведем программу на языке ассемблера для IBM PC. Программа вычисляет значение $a = b + c$ для целых a , b и c :

	.MODEL SMALL	Определяет модель памяти. Модель SMALL означает, что программа и данные помещаются в сегменты.
	.DATA	Определяет сегмент данных. Сегмент данных помещается в сегмент данных.
b	DW 5	Определяет сегмент данных. DW означает, что данные помещаются в сегмент данных.
c	DW 3	Определяет сегмент данных. DW означает, что данные помещаются в сегмент данных.
a	DW ?	Определяет сегмент данных. DW означает, что данные помещаются в сегмент данных.
	.CODE	Определяет сегмент кода. Сегмент кода помещается в сегмент кода.
begin	MOV AX,@DATA	Перемещает указатель на сегмент данных в регистр AX.
	MOV DS,AX	Перемещает указатель на сегмент данных в регистр DS (Data Segment).
	MOV AX,B	Перемещает значение B в регистр AX.
	ADD AX,C	Добавляет значение C к значению в регистре AX.
	MOV A,AX	Перемещает значение AX в регистр A.
	MOV AH,4CH	Перемещает значение 4CH в регистр AH.
	INT 21H	Вызывает прерывание 21H.
	END begin	Завершает программу.

Перевод программы с языка ассемблера на машинный язык осуществляется специальной программой, которая называется **ассемблером** и является, по сути, простейшим [транслятором](#).

7.16. В чем преимущества алгоритмических языков перед машинными?

Основные преимущества таковы:

- **алфавит** алгоритмического языка значительно шире алфавита машинного языка, что существенно повышает наглядность текста программы;
- **набор операций**, допустимых для использования, не зависит от набора машинных операций, а выбирается из соображений удобства формулирования алгоритмов решения задач определенного класса;
- **формат предложений** достаточно гибок и удобен для использования, что позволяет с помощью одного предложения задать достаточно содержательный этап обработки данных;
- требуемые операции задаются с помощью общепринятых математических обозначений;
- данным в алгоритмических языках присваиваются индивидуальные имена, выбираемые программистом;
- в языке может быть предусмотрен значительно более широкий набор типов данных по сравнению с набором машинных типов данных.

Таким образом, алгоритмические языки в значительной мере являются **машинно-независимыми**. Они облегчают **работу программиста** и **повышают надежность создаваемых программ**.

7.17. Какие компоненты образуют алгоритмический язык?

Алгоритмический язык (как и любой другой язык) образуют три его составляющие: *алфавит, синтаксис и семантика*.

Алфавит — это фиксированный для данного языка набор основных символов, т.е. "букв алфавита", из которых должен состоять любой текст на этом языке — никакие другие символы в тексте не допускаются.

Синтаксис — это правила построения фраз, позволяющие определить, правильно или неправильно написана та или иная фраза. Точнее говоря, синтаксис языка представляет собой набор правил, устанавливающих, какие комбинации символов являются осмысленными предложениями на этом языке.

Семантика определяет смысловое значение предложений языка. Являясь системой правил истолкования отдельных языковых конструкций, семантика устанавливает, какие последовательности действий описываются теми или иными фразами языка и, в конечном итоге, какой алгоритм определен данным текстом на алгоритмическом языке.

7.18. Какие понятия используют алгоритмические языки?

Каждое понятие алгоритмического языка подразумевает некоторую синтаксическую единицу (конструкцию) и определяемые ею свойства программных объектов или процесса обработки данных.

Понятие языка определяется во взаимодействии синтаксических и семантических правил. Синтаксические правила показывают, как образуется данное понятие из других понятий и букв алфавита, а семантические правила определяют свойства данного понятия

Основными понятиями в алгоритмических языках обычно являются следующие.

1. Имена (идентификаторы) — употребляются для обозначения объектов программы (переменных, массивов, функций и др.).

2. Операции. Типы операций:

- **арифметические** операции $+$, $-$, $*$, $/$ и др. ;
- **логические** операции **и** , **или** , **не** ;
- операции **отношения** $<$, $>$, \leq , \geq , $=$, \neq ;
- операция **сцепки** (иначе, "присоединения", "конкатенации") символьных значений друг с другом с образованием одной длинной строки; изображается знаком "+".

3. Данные — величины, обрабатываемые программой. Имеется три основных вида данных: **константы, переменные и массивы**.

- **Константы** — это данные, которые зафиксированы в тексте программы и не изменяются в процессе ее выполнения.

Примеры констант:

- **числовые** 7.5 , 12 ;
 - **логические** да (истина), нет (ложь);
 - **символьные** (содержат ровно один символ) "А" , "+" ;
 - **литерные** (содержат произвольное количество символов) "а0", "Мир", "" (пустая строка).
- **Переменные** обозначаются именами и могут изменять свои значения в ходе выполнения программы. Переменные бывают **целые, вещественные, логические, символьные и литерные**.
 - **Массивы** — **последовательности однотипных элементов, число которых фиксировано и которым присвоено одно имя**. Положение элемента в массиве однозначно определяется его индексами (одним, в случае одномерного массива, или несколькими, если массив многомерный). Иногда массивы называют **таблицами**.

4. Выражения — **предназначаются для выполнения необходимых вычислений**, состоят из констант, переменных, указателей функций (например, $\exp(x)$), объединенных знаками операций.

Выражения записываются в виде **линейных последовательностей символов** (без подстрочных и надстрочных символов, "многоэтажных" дробей и т.д.), что позволяет вводить их в компьютер, последовательно нажимая на соответствующие клавиши клавиатуры.

Различают выражения **арифметические, логические и строковые**.

- **Арифметические выражения** служат для определения одного числового значения. Например, $(1 + \sin(x))/2$. Значение этого выражения при $x=0$ равно 0.5, а при $x=\pi/2$ — единице.
- **Логические выражения** описывают некоторые условия, которые могут удовлетворяться или не удовлетворяться. Таким образом, логическое выражение может принимать только два значения — **"истина"** или **"ложь"** (да или нет). Рассмотрим в качестве примера логическое выражение $x*x + y*y < r*r$, определяющее принадлежность точки с координатами (x, y) внутренней области круга радиусом r с центром в начале координат. При $x=1, y=1, r=2$ значение этого выражения — **"истина"**, а при $x=2, y=2, r=1$ — **"ложь"**.
- **Строковые (литерные) выражения, значениями которых являются тексты**. В строковые выражения могут входить литерные и строковые константы, литерные и строковые переменные, литерные функции, разделенные знаками операции сцепки. Например, $A + B$ означает присоединение строки B к концу строки A. Если $A = \text{"куст"}$, а $B = \text{"зеленый"}$, то значение выражения $A + B$ есть **"куст зеленый"**.

5. Операторы (команды). Оператор — это наиболее крупное и содержательное понятие языка: **каждый оператор представляет собой законченную фразу языка и**

определяет некоторый вполне законченный этап обработки данных. В состав операторов входят:

- ключевые слова;
- данные;
- выражения и т.д.

Операторы подразделяются на исполняемые и неисполняемые. **Неисполняемые** операторы предназначены для описания данных и структуры программы, а **исполняемые** — для выполнения различных действий (например, оператор присваивания, операторы ввода и вывода, условный оператор, операторы цикла, оператор процедуры и др.).

7.19. Что такое стандартная функция?

При решении различных задач с помощью компьютера бывает необходимо вычислить логарифм или модуль числа, синус угла и т.д.

Вычисления часто употребляемых функций осуществляются посредством подпрограмм, называемых стандартными функциями, которые заранее запрограммированы и встроены в транслятор языка.

Таблица стандартных функций школьного алгоритмического языка

Название и математическое обозначение функции		Указатель функции
Абсолютная величина (модуль)	$ x $	abs(x)
Корень квадратный	\sqrt{x}	sqrt(x)
Натуральный логарифм	$\ln x$	ln(x)
Десятичный логарифм	$\lg x$	lg(x)
Экспонента (степень числа $e \sim 2.72$)	e^x	exp(x)
Знак числа x (- 1, если $x < 0$; 0, если $x = 0$; 1, если $x > 0$)	sign x	sign(x)
Целая часть x (т.е. максимальное целое число, не превосходящее x)		int(x)
Минимум из чисел x и y		min(x,y)
Максимум из чисел x и y		max(x,y)
Частное от деления целого x на целое y		div(x,y)
Остаток от деления целого x на целое y		mod(x,y)
Случайное число в диапазоне от 0 до $x - 1$		rnd(x)
Синус (угол в радианах)	$\sin x$	sin(x)
Косинус (угол в радианах)	$\cos x$	cos(x)

Название и математическое обозначение функции		Указатель функции
Тангенс (угол в радианах)	tg x	tg(x)
Котангенс (угол в радианах)	ctg x	ctg(x)
Арксинус (главное значение в радианах)	arcsin x	arcsin(x)
Арккосинус (главное значение в радианах)	arccos x	arccos(x)
Арктангенс (главное значение в радианах)	arctg x	arctg(x)
Арккотангенс (главное значение в радианах)	arcctg x	arcctg(x)

В качестве аргументов функций можно использовать константы, переменные и выражения. Например:

$\sin(3.05)$ $\sin(x)$ $\sin(2 * y + t / 2)$ $\sin((\exp(x) + 1) ** 2)$
 $\min(a, 5)$ $\min(a, b)$ $\min(a + b, a * b)$ $\min(\min(a, b), \min(c, d))$

Каждый язык программирования имеет свой набор стандартных функций.

7.20. Как записываются арифметические выражения?

Арифметические выражения записываются по следующим правилам:

- Нельзя опускать знак умножения между сомножителями и ставить рядом два знака операций.
- Индексы элементов массивов записываются в квадратных (школьный АЯ, Pascal) или круглых (Basic) скобках.
- Для обозначения переменных используются буквы латинского алфавита.
- **Операции выполняются в порядке старшинства:** сначала вычисление функций, затем возведение в степень, потом умножение и деление и в последнюю очередь — сложение и вычитание.
- **Операции одного старшинства выполняются слева направо.** Однако, в школьном АЯ есть одно **исключение** из этого правила: операции возведения в степень выполняются справа налево. Так, выражение $2**(3**2)$ в школьном АЯ вычисляется как $2**(3**2) = 512$. В языке QBasic аналогичное выражение 2^3^2 вычисляется как $(2^3)^2 = 64$. А в языке **Pascal** вообще не предусмотрена операция возведения в степень, в Pascal x^y записывается как $\exp(y * \ln(x))$, а x^y^z как $\exp(\exp(z * \ln(y)) * \ln(x))$.

Примеры записи арифметических выражений

Математическая запись	Запись на школьном алгоритмическом языке
$\frac{xy}{z}$	<code>x * y / z</code>
$\frac{x}{yz}$	<code>x / (y * z)</code> или <code>x / y / z</code>
$\frac{a^3 + b^3}{bc}$	<code>(a**3 + b**3) / (b*c)</code>
$\frac{a_{i+1} + b_{i-1}}{2xy}$	<code>(a[i+1] + b[i-1]) / (2*x*y)</code>
$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$	<code>(-b + sqrt(b*b - 4*a*c)) / (2*a)</code>
$\sqrt[5]{x} (x < 0)$	<code>sign(x) * abs(x) ** (1/5)</code>
$0,49e^{a^2-b^2} + \ln^3 \cos a^2$	<code>0.49 * exp(a*a - b*b) + ln(cos(a*a)) ** 3</code>
$\frac{x}{1 + \frac{x^2}{3 + (2x)^3}}$	<code>x / (1 + x*x / (3 + (2*x)**3))</code>

Типичные ошибки в записи выражений:

<code>5x + 1</code>	Пропущен знак умножения между 5 и x
<code>a + sin x</code>	Аргумент x функции sin x не заключен в скобки
<code>((a + b) / c**3</code>	Не хватает закрывающей скобки

7.21. Как записываются логические выражения?

В записи логических выражений помимо арифметических операций сложения, вычитания, умножения, деления и возведения в степень **используются операции отношения** < (меньше), <= (меньше или равно), > (больше), >= (больше или равно), = (равно), <> (не равно), **а также логические операции и, или, не.**

Примеры записи логических выражений, истинных при выполнении указанных условий.

Условие	Запись на школьном алгоритмическом языке
Дробная часть вещественного числа a равна нулю	$\text{int}(a) = 0$
Целое число a — четное	$\text{mod}(a, 2) = 0$
Целое число a — нечетное	$\text{mod}(a, 2) = 1$
Целое число k кратно семи	$\text{mod}(a, 7) = 0$
Каждое из чисел a, b положительно	$(a > 0)$ и $(b > 0)$
Только одно из чисел a, b положительно	$((a > 0) \text{ и } (b \leq 0))$ или $((a \leq 0) \text{ и } (b > 0))$
Хотя бы одно из чисел a, b, c является отрицательным	$(a < 0)$ или $(b < 0)$ или $(c < 0)$
Число x удовлетворяет условию $a < x < b$	$(x > a)$ и $(x < b)$
Число x имеет значение в промежутке [1, 3]	$(x \geq 1)$ и $(x \leq 3)$
Целые числа a и b имеют одинаковую четность	$((\text{mod}(a, 2) = 0) \text{ и } (\text{mod}(b, 2) = 0))$ или $((\text{mod}(a, 2) = 1) \text{ и } (\text{mod}(b, 2) = 1))$
Точка с координатами (x, y) лежит в круге радиуса r с центром в точке (a, b)	$(x-a)**2 + (y-b)**2 < r*r$
Уравнение $ax^2 + bx + c = 0$ не имеет действительных корней	$b*b - 4*a*c < 0$
Точка (x, y) принадлежит первой или третьей четверти	$((x > 0) \text{ и } (y > 0))$ или $((x < 0) \text{ и } (y > 0))$
Точка (x, y) принадлежит внешности единичного круга с центром в начале координат или его второй четверти	$(x*x + y*y > 1)$ или $((x*x + y*y \leq 1) \text{ и } (x < 0) \text{ и } (y > 0))$
Целые числа a и b являются взаимнопротивоположными	$a = -b$
Целые числа a и b являются взаимнообратными	$a*b = 1$
Число a больше среднего арифметического чисел b, c, d	$a > (b+c+d) / 3$
Число a не меньше среднего геометрического чисел b, c, d	$a \geq (b+c+d) ** (1/3)$
Хотя бы одна из логических переменных $F1$ и $F2$ имеет значение да	$F1$ или $F2$
Обе логические переменные $F1$ и $F2$ имеют значение да	$F1$ и $F2$
Обе логические переменные $F1$ и $F2$ имеют значение нет	не $F1$ и не $F2$

Логическая переменная F1 имеет значение да , а логическая переменная F2 имеет значение нет	F1 и не F2
Только одна из логических переменных F1 и F2 имеет значение да	(F1 и не F2) или (F2 и не F1)

7.22. Упражнения

7.1. Запишите по правилам алгоритмического языка выражения:

а) $\frac{x+y}{x-1/2} - \frac{x-z}{xy};$

е) $\frac{\sqrt{|\sin^2 x|}}{3,01x - e^{2x}};$

б) $(1+z) \frac{x+\frac{y}{z}}{a - \frac{1}{1+x^y}};$

ж) $\frac{|\cos x^2 - \sin^2 y|}{\sqrt[4]{|\ln x|} + xy};$

в) $(x^n)^{m+2} + x^{n^m};$

з) $\ln(y^{-\sqrt{|x+1|}}) \cdot \sin^2 \operatorname{arctg} z;$

г) $\frac{(a+b)^n}{1 + \frac{a}{a^m - b^{m-n}}};$

и) $r_{ij}^{|x-y|} - 0,15|\sin e^{-x^2}|;$

д) $\frac{\left(a_i^{2l} + b_{j+1}^{2k}\right) \cdot (3^n - x^2 y)}{z - \frac{d_{i,j+1}+1}{z + \frac{y}{\sqrt{t^2 + xyz}}}};$

к) $a^{(x+y)/2} - 3\sqrt[3]{\frac{x-1}{|y|+1}} \cdot e^{-(y+u/2)}$

7.2. Запишите в обычной математической форме арифметические выражения:

а) $a / b ** 2;$

л) $5 * \operatorname{arctg}(x) - \operatorname{arctg}(y) / 4;$

б) $a+b/c+1;$

м) $\lg(u * (1/3) + \operatorname{sqrt}(v) + z);$

в) $1/a*b/c;$

н) $\ln(y * (-\operatorname{sqrt}(\operatorname{abs}(x))));$

г) $a**b**c/2;$

о) $\operatorname{abs}(x**(y/x) - (y/x)**(1/3));$

д) $(a**b)**c/2;$

п) $\operatorname{sqrt}((x1-x2)**2 + (y1-y2)**2);$

е) $a/b/c/d*p*q;$

р) $\exp(\operatorname{abs}(x-y)) * (\operatorname{tg}(z)**2+1)**x;$

ж) $x**y**z/a/b;$

с) $\lg(\operatorname{sqrt}(\exp(x-y)) + x**\operatorname{abs}(y) + z);$

з) $4/3*3.14*r**3;$

т) $\operatorname{sqrt}(\exp(a*x) * \sin(x)**n) / \cos(x)**2;$

и) $b/\operatorname{sqrt}(a*a+b);$

у) $\operatorname{sqrt}(\sin(\operatorname{arctg}(u))**2 + \operatorname{abs}(\cos(v)));$

к) $d*c/2/R+a**3;$

ф) $\operatorname{abs}(\cos(x) + \cos(y))** (1 + \sin(y)**2);$

7.3. Вычислите значения арифметических выражений при $x=1$:

а) $\operatorname{abs}(x-3) / \ln(\exp(3)) * 2 / \lg(10000);$

Решение: $\operatorname{abs}(1-3)=2; \ln(\exp(3))=3; \lg(10000)=4; 2/3*2/4=0.33;$

б) $\operatorname{sign}(\operatorname{sqrt}(\operatorname{sqrt}(x+15))) * 2**2**2;$

в) $\operatorname{int}(-2.1) * \operatorname{int}(-2.9) / \operatorname{int}(2.9) + x;$

г) $-\operatorname{sqrt}(x+3)**2**(\operatorname{sign}(x+0.5)*3) + \operatorname{tg}(0);$

- д) $\lg(x) + \cos(x^2 - 1) * \sqrt{x + 8} - \text{div}(2, 5);$
- е) $\text{sign}(x - 2) * \sqrt{\text{int}(4.3)} / \text{abs}(\min(2, -1));$
- ж) $\text{div}(10, x + 2) * \text{mod}(10, x + 6) / \max(10, x) * \text{mod}(2, 5).$

7.4. Запишите арифметические выражения, значениями которых являются:

- а) площадь треугольника со сторонами a, b, c ($a, b, c > 0$) и полупериметром p ;

Ответ: $\sqrt{p * (p - a) * (p - b) * (p - c)};$

- б) среднее арифметическое и среднее геометрическое чисел a, b, c, d ;
- в) расстояние от точки с координатами (x, y) до точки $(0, 0)$;
- г) синус от x градусов;
- д) площадь поверхности куба (длина ребра равна a);
- е) радиус описанной сферы куба (длина ребра равна a);
- ж) координаты точки пересечения двух прямых, заданных уравнениями $a_1x + b_1y + c_1 = 0$ и $a_2x + b_2y + c_2 = 0$ (прямые не параллельны).

7.5. Вычислите значения логических выражений:

- а) $x * x + y * y \leq 9$ при $x = 1, y = -2$

Ответ: да;

- б) $b * b - 4 * a * c < 0$ при $a = 2, b = 1, c = -2$;
- в) $(a > 1)$ **и** $(a \leq 2)$ при $a = 1.5$;
- г) $(a < 1)$ **или** $(a > 1.2)$ при $a = 1.5$;
- д) $(\text{mod}(a, 7) = 1)$ **и** $(\text{div}(a, 7) = 1)$ при $a = 8$;
- е) **не** $((a > b)$ **и** $(a < 9)$ **или** $(a * a = 4))$ при $a = 5, b = 4$.

7.6. Запишите логические выражения, истинные только при выполнении указанных условий:

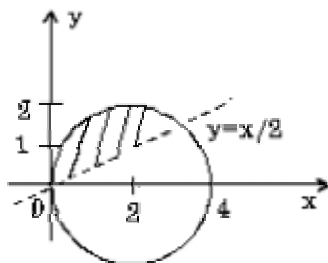
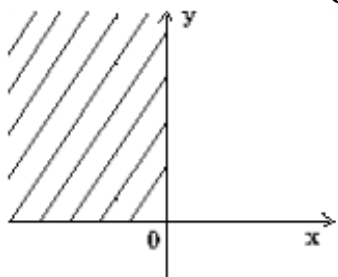
- а) x принадлежит отрезку $[a, b]$

Ответ: $(x \geq a)$ **и** $(x \leq b)$;

- б) x лежит вне отрезка $[a, b]$;
- в) x принадлежит отрезку $[a, b]$ или отрезку $[c, d]$;
- г) x лежит вне отрезков $[a, b]$ и $[c, d]$;
- д) целое k является нечетным числом;
- е) целое k является трехзначным числом, кратным пяти;
- ж) элемент $a_{i,j}$ двумерного массива находится на пересечении нечетной строки и четного столбца;
- з) прямые $a_1x + b_1y + c_1 = 0$ и $a_2x + b_2y + c_2 = 0$ параллельны;
- и) из чисел a, b, c меньшим является c , а большим b ;
- к) среди чисел a, b, c, d есть взаимно противоположные;
- л) среди целых чисел a, b, c есть хотя бы два четных;
- м) из отрезков с длинами a, b, c можно построить треугольник;
- н) треугольники со сторонами a_1, b_1, c_1 и a_2, b_2, c_2 подобны;
- о) точка с координатами (x, y) принадлежит внутренней области треугольника с вершинами $A(0, 5), B(5, 0)$ и $C(1, 0)$;
- п) точка с координатами (x, y) принадлежит области, внешней по отношению к треугольнику с вершинами $A(0, 5), B(1, 0)$ и $C(5, 0)$;
- р) четырехугольник со сторонами a, b, c и d является ромбом.

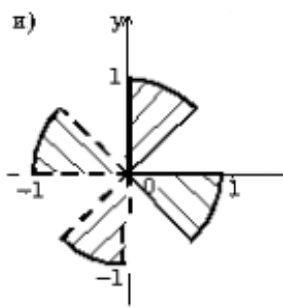
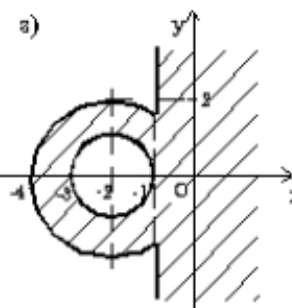
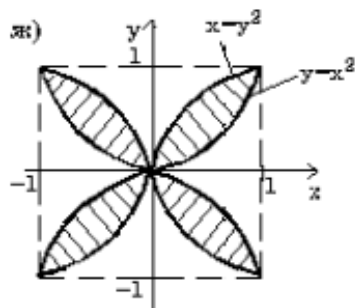
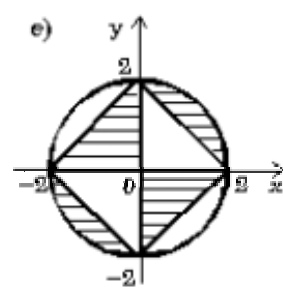
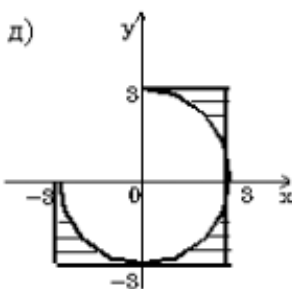
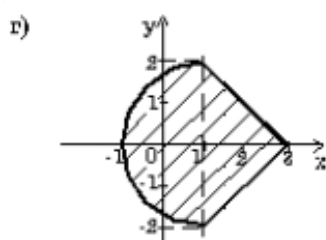
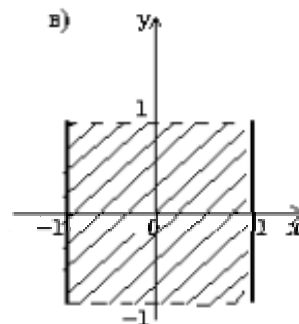
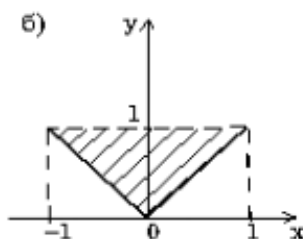
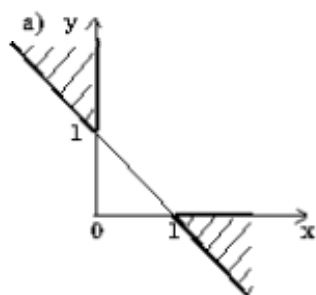
7.7. Начертите на плоскости (x,y) область, в которой и только в которой истинно указанное выражение. Границу, не принадлежащую этой области, изобразите пунктиром.

- а) $(x \leq 0)$ и $(y \geq 0)$ е) $((x-2)^2 + y^2 \leq 4)$ и $(y > x/2)$
Ответ: **Ответ:**



- б) $(x \geq 0)$ или $(y \leq 0)$ ж) $(x^2 + y^2 < 1)$ и $(y > x^2)$;
 в) $x + y \geq 0$ з) $(y > x)$ и $(y + x \geq 0)$ и $(y \leq 1)$;
 г) $(x + y > 0)$ и $(y < 0)$ и) $(\text{abs}(x) \leq 1)$ и $(y < 2)$;
 д) $\text{abs}(x) + \text{abs}(y) \geq 1$ к) $(x^2 + y^2 < 4)$ и $(x^2 + y^2 > 1)$;

7.8. Запишите логическое выражение, которое принимает значение "истина" тогда и только тогда, когда точка с координатами (x, y) принадлежит заштрихованной области.



7.9. Пусть $a=3, b=5, c=7$. Какие значения будут иметь эти переменные в результате выполнения последовательности операторов:

а) $a:=a+1; b:=a+b; c:=a+b; a:=\text{sqrt}(a)$

Решение: $a=3+1=4, b=4+5=9, c=4+9=13, a = \{\text{корень квадратный из } 4\} = 2$.

Ответ: $a=2, b=9, c=13$;

б) $c:=a*b+2; b:=b+1; a:=c-b**2; b:=b*a;$

в) $b:=b+a; c:=c+b; b:=1/b*c;$

г) $p:=c; c:=b; b:=a; a:=p; c:=a*b*c*p;$

д) $c:=a**(b-3); b:=b-3; a:=(c+1)/2*b; c:=(a+b)*a;$

е) $x:=a; a:=b; b:=c; c:=x; a:=\text{sqrt}(a+b+c+x-2);$

ж) $b:=(a+c)**2; a:=\lg(b**2)**2; c:=c*a*b.$

7.10. Задайте с помощью операторов присваивания следующие действия:

а) массив $X=(x_1, x_2)$ преобразовать по правилу: в качестве x_1 взять сумму, а в качестве x_2 — произведение исходных компонент;

Решение: $c:=x[1]; x[1]:=x[1]+x[2]; x[2]:=c*x[2]$

б) поменять местами значения элементов массива $X=(x_1, x_2)$;

в) в массиве $A(N)$ компоненту с номером i ($1 < i < N$) заменить полусуммой исходных соседних с нею компонент, соседнюю справа компоненту заменить на нуль, а соседнюю слева компоненту увеличить на 0.5;

г) $u = \max(x, y, z) + \min(x-z, y+z, y, z);$

7.11. Задайте с помощью команд **если** или **выбор** вычисления по формулам:

$$\text{а) } y = \begin{cases} \sqrt{x} & , \text{ если } x \leq -100, \\ \sqrt[3]{x} & , \text{ если } -100 < x < 100, \\ \sqrt{x} & , \text{ если } x \geq 100; \end{cases}$$

$$\text{б) } z = \begin{cases} x^2 + y^2 & , \text{ если } x^2 + y^2 \leq 1, \\ x + y & , \text{ если } x^2 + y^2 > 1 \text{ и } y \geq x, \\ 0,5 & , \text{ если } x^2 + y^2 > 1 \text{ и } y < x; \end{cases}$$

$$F(z) = \begin{cases} 2z + 1, & \text{ если } z \geq 0, \\ \sin z, & \text{ если } z < 0, \end{cases}$$

в) где

$$z = \begin{cases} \lg(-x), & \text{ если } x < 0, \\ \sqrt{x+1}, & \text{ если } x \geq 0. \end{cases}$$

$$\text{г) } z = \begin{cases} 1 & , \text{ если } c = 0, \\ x & , \text{ если } c = 1, \\ 3x^2 - 1/2 & , \text{ если } c = 2, \\ x^3 - 3x/2 & , \text{ если } c = 3, \\ 2x^4 - 3x/2 & , \text{ в противном случае} \end{cases}$$

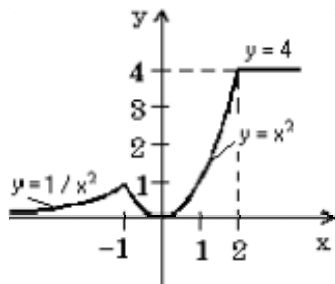
$$д) \quad z = \begin{cases} \sqrt{x^2 + y^2} & , \text{ если } |x| + |y| < r, \\ \max(|x|, |y|) & , \text{ если } |x| + |y| \geq r; \end{cases}$$

$$е) \quad v = \begin{cases} x + y & , \text{ если } x > 1 \text{ и } y > 1, \\ x - y & , \text{ если } x > 1 \text{ и } y \leq 1, \\ -x + y & , \text{ если } x \leq 1 \text{ и } y > 0, \\ -x - y & , \text{ если } x \leq 1 \text{ и } y \leq 0. \end{cases}$$

$$ж) \quad z = \begin{cases} |x| + |y| & , \text{ если точка лежит внутри круга радиусом } r \ (r > 0) \text{ с центром в } (a, b) \\ x + y & , \text{ в противном случае} \end{cases}$$

7.12. Постройте графики функций $y(x)$, заданных командами **если**:

```
а)  если x <= -1
    то y := 1/x**2
    иначе
      если x <= 2
      то y := x*x
      иначе y := 4
    все
все
```



```
б)  если x < -5
    то y := -5
    иначе
      если x < 0
      то y := x
      иначе
        если x < 3
        то y := 2*x
        иначе y := 6
      все
    все
все
```

```
в)  если x < -0.5
    то y := 1/abs(x)
    иначе
      если x < 1
      то y := 2
      иначе y := 1/(x-0.5)
    все
все
```

```
г)  если x < 0
    то y := 1
    иначе
      если x < 3.14
      то y := cos(x)
      иначе y := -1
    все
все
```

```
д)  если abs(x) > 2
    то y := x*x
    иначе
      если x < 0
      то y := -2*x
      иначе
        если x >= 1
        то y := 4
        иначе y := 4*x*x
      все
    все
все
```

Решение

7.13. Определите значение целочисленной переменной S после выполнения операторов:

а) S:=128
 нц для i от 1 до 4
 S:=div(S,2)
 кц

Решение

i	S
	128
1	128/2=64
2	64/2=32
3	32/2=16
4	16/2=8

Ответ: S=8

г) S:=0
 нц для i от 1 до 2
 нц для j от 2 до 3
 S:=S+i+j
 кц
 кц

Решение

i	j	S
		0
1	2	0+1+2=3
	3	3+1+3=7
2	2	7+2+2=11
	3	11+2+3=16

Ответ: S=16

б) S:=1; a:=1
 нц для i от 1 до 3
 S:=S+i*(i+1)*a
 a:=a+2
 кц

д) нц для i от 1 до 3
 S:=0
 нц для j от 2 до 3
 S:=S+i+j
 кц
 кц

в) S:=1; a:=1
 нц для i от 1 до 3
 S := S+i
 нц для j от 2 до 3
 S := S+j
 кц
 кц

е) нц для i от 1 до 2
 S := 0
 нц для j от 2 до 3
 нц для k от 1 до 2
 S := S+i+j+k
 кц
 кц
 кц

7.14. Определите значение переменной S после выполнения операторов:

а) i:=0; S:=0
 нц пока i<3
 i:=i+1;
 S:=S+i*i
 кц

г) S:=0; N:=125
 нц пока N>0
 S:=S+mod(N,10) | S — сумма цифр
 N:=div(N,10) | числа N
 кц

Решение

Условие i < 3	i	S
	0	0
0 < 3? да	1	0+1 ² =1
1 < 3? да	2	1+2 ² =5
2 < 3? да	3	5+3 ² =14
3 < 3? нет(кц)		

Ответ: S=14

б) S:=0; i:=1

Решение

Условие N > 0	S	N
	0	125
125 > 0? да	0+5=5	12
12 > 0? да	5+2=7	1
1 > 0? да	7+1=8	0
0 > 0? нет(кц)		

Ответ: S=8

д) a:=1; b:=1; S:=0;

<pre> нц пока i>1 S:=S+1/i i:=i-1 кц </pre>	<pre> нц пока a<=5 a:=a+b; b:=b+a; S:=S+a+b кц </pre>
<pre> в) S:=0; i:=1; j:=5 нц пока i<j S:=S+i*j i:=i+1 j:=j-1 кц </pre>	<pre> е) a:=1; b:=1 нц пока a+b<10 a:=a+1 b:=b+a кц S:=a+b </pre>

7.15. Составьте алгоритмы решения задач линейной структуры (условия этих задач заимствованы из учебного пособия В.М. Заварыкина, В.Г. Житомирского и М.П. Лапчика "Основы информатики и вычислительной техники", 1989):

а) в треугольнике известны три стороны a , b и c ; найти (в градусах) углы этого треугольника, используя формулы:

$$\cos A = \frac{b^2 + c^2 - a^2}{2bc}; \quad \sin B = \frac{b \sin A}{a}; \quad C = 180^\circ - (A + B).$$

Пояснение. Обратите внимание на то, что стандартные тригонометрические функции \arccos и \arcsin возвращают вычисленное значение в радианной мере.

Решение:

```

алг Углы треугольника (арг вещь a, b, c, рез вещь UголA, UголB, UголC)
нач вещ RadGr, UголARad
  | RadGr — коэф. перевода угла из радианной меры в градусную
  | UголARad — угол A (в радианах)
  RadGr:=180/3.14
  UголARad:=ArcCos ( (b*b+c*c-a*a) / (2*b*c) )
  UголA:=UголARad*RadGr
  UголB:=ArcSin (b*sin (UголARad) / a) *RadGr
  UголC:=180- (UголA+UголB)
кон

```

б) в треугольнике известны две стороны a , b и угол C (в радианах) между ними; найти сторону c , углы A и B (в радианах) и площадь треугольника, используя формулы:

$$\sin A = \frac{a \sin C}{c}; \quad \sin B = \frac{b \sin C}{c};$$

$$S = \frac{bc \sin A}{2}; \quad c^2 = a^2 + b^2 - 2ab \cos C.$$

Пояснение. Сначала нужно найти сторону c , а затем остальные требуемые значения;

в) в треугольнике известны три стороны a , b и c ; найти радиус описанной окружности и угол A (в градусах), используя формулы:

$$S = \frac{bc \sin A}{2}; \quad \operatorname{tg} \frac{A}{2} = \sqrt{\frac{(p-b)(p-c)}{p(p-a)}}, \quad \text{где} \quad p = \frac{a+b+c}{2}.$$

г) в правильной треугольной пирамиде известны сторона основания a и угол A (в градусах) наклона боковой грани к плоскости основания; найти объем и площадь полной поверхности пирамиды, используя формулы:

$$V = S_{\text{осн}} \cdot H/2; \quad S_{\text{п}} = S_{\text{осн}} \cdot \left(1 + \frac{1}{\cos \alpha}\right),$$

$$\text{где } S_{\text{осн}} = \frac{a^2 \sqrt{3}}{4}, \quad H = \frac{a \sqrt{3}}{6} \operatorname{tg} \alpha.$$

д) в усеченном конусе известны радиусы оснований R и r и угол A (в радианах) наклона образующей к поверхности большего основания; найти объем и площадь боковой поверхности конуса, используя формулы:

$$V = \frac{1}{3} \pi H (r^2 + R^2 + rR); \quad S_{\text{бок}} = \pi l (r + R)$$

$$\text{где } H = (R - r) \operatorname{tg} \alpha, \quad l = \frac{R - r}{\cos \alpha}.$$

е) в правильной четырехугольной пирамиде сторона основания равна a , а боковое ребро наклонено к плоскости основания под углом A ; найти объем и площадь полной поверхности пирамиды и площадь сечения, проходящего через вершину пирамиды и диагональ основания d ; использовать формулы:

$$H = \frac{a \sqrt{2}}{2} \cdot \operatorname{tg} \alpha; \quad V = \frac{1}{3} \cdot S_{\text{осн}} \cdot H;$$

$$S_{\text{сеч}} = \frac{1}{2} H a \sqrt{2} = \frac{a H \sqrt{2}}{2}; \quad S_{\text{полн}} = a^2 + a^2 \sqrt{2 \operatorname{tg}^2 \alpha + 1}.$$

7.16. Составьте алгоритм решения задач разветвляющейся структуры:

а) определить, является ли треугольник с заданными сторонами a , b , c равнобедренным;

Решение:

```

алг Треугольник (арг вещ  $a, b, c$ , рез лог  $\text{Otv}$ )
  дано |  $a > 0, b > 0, c > 0, a + b > c, a + c > b, b + c > a$ 
  надо |  $\text{Otv} = \text{да}$ , если треугольник равнобедренный
        |  $\text{Otv} = \text{нет}$ , если треугольник не равнобедренный
нач
  если  $(a = b)$  или  $(a = c)$  или  $(b = c)$ 
  то  $\text{Otv} := \text{да}$ 
  иначе  $\text{Otv} := \text{нет}$ 
все
кон

```

б) определить количество положительных чисел среди заданных чисел a , b и c ;

в) меньшее из двух заданных неравных чисел увеличить вдвое, а большее оставить без изменения;

г) числа a и b — катеты одного прямоугольного треугольника, а c и d — другого; определить, являются ли эти треугольники подобными;

д) даны три точки на плоскости; определить, какая из них ближе к началу координат;

е) определить, принадлежит ли заданная точка (x, y) плоской фигуре, являющейся кольцом с центром в начале координат, с внутренним радиусом r_1 и внешним радиусом r_2 ;

ж) упорядочить по возрастанию последовательность трех чисел a , b и c .

Ответы — Раздел 7. Алгоритмы. Алгоритмизация. Алгоритмические языки

7.1.

а) $(x+y)/(x-1/2) - (x-z)/(x*y)$;
 б) $(1+z)*(x+y/z)/(a-1/(1+x*x))$;
 в) $x**(n*(m+2)) + x**(n**m)$;
 г) $(a+b)**n/(1+a/(a**m-b**(m-n)))$;
 д) $(a[i]**(2*1) + b[j+1]**(2*k)) * (3**n-x*x*y)/(z-(d[i,j+1]+1)/(z+y/sqrt(t*t+x*y*z)))$;
 е) $\sqrt{\text{abs}(\sin(x)**2)/(3.01*x - \exp(2*x))}$;
 ж) $\text{abs}(\cos(x**3) - \sin(y)**2) / (\text{abs}(\ln(x))**(1/4) + x*y)$;
 з) $\ln(y**(-\sqrt{\text{abs}(x+1)})) * \sin(\text{arctg}(z))**2$;
 и) $r[i,j]**\text{abs}(x-y) - 0.15*\text{abs}(\sin(\exp(-z**8)))$;
 к) $a**((x+y)/2) - ((x-1)/(\text{abs}(y)+1))**(1/3)*\exp(-(y+u/2))$.

7.2. а) $\frac{a}{b^2}$; б) $a + \frac{b}{c} + 1$; в) $\frac{b}{a \cdot c}$; г) $\frac{a^{b^c}}{2}$; д) $\frac{a^{bc}}{2}$; е) $\frac{a \cdot p \cdot q}{b \cdot c \cdot d}$; ж) $\frac{x^{y^z}}{a \cdot b}$;
 з) $\frac{4\pi^3}{3}$; и) $\frac{b}{\sqrt{a^2 + b}}$; к) $\frac{dc}{2R} + a^3$; л) $\frac{\text{arctg } y}{4}$; м) $\lg\left(\frac{u}{3} + \sqrt{v} + z\right)$;
 н) $\ln(-y\sqrt{|x|})$; о) $\left|x^{y/k} - 3\sqrt{\frac{y}{x}}\right|$; п) $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$; р) $e^{k-y} \cdot (\text{tg}^2 z + 1)^x$;
 с) $\lg\left(\sqrt{e^{x-y}} + x^{|y|} + z\right)$; т) $\frac{\sqrt{e^{ax} \cdot \sin^n x}}{\cos^2 x}$; у) $\sqrt{\sin^2 \text{arctg } u + |\cos v|}$;
 ф) $|\cos x + \cos y|^{1+\sin^2 y}$.

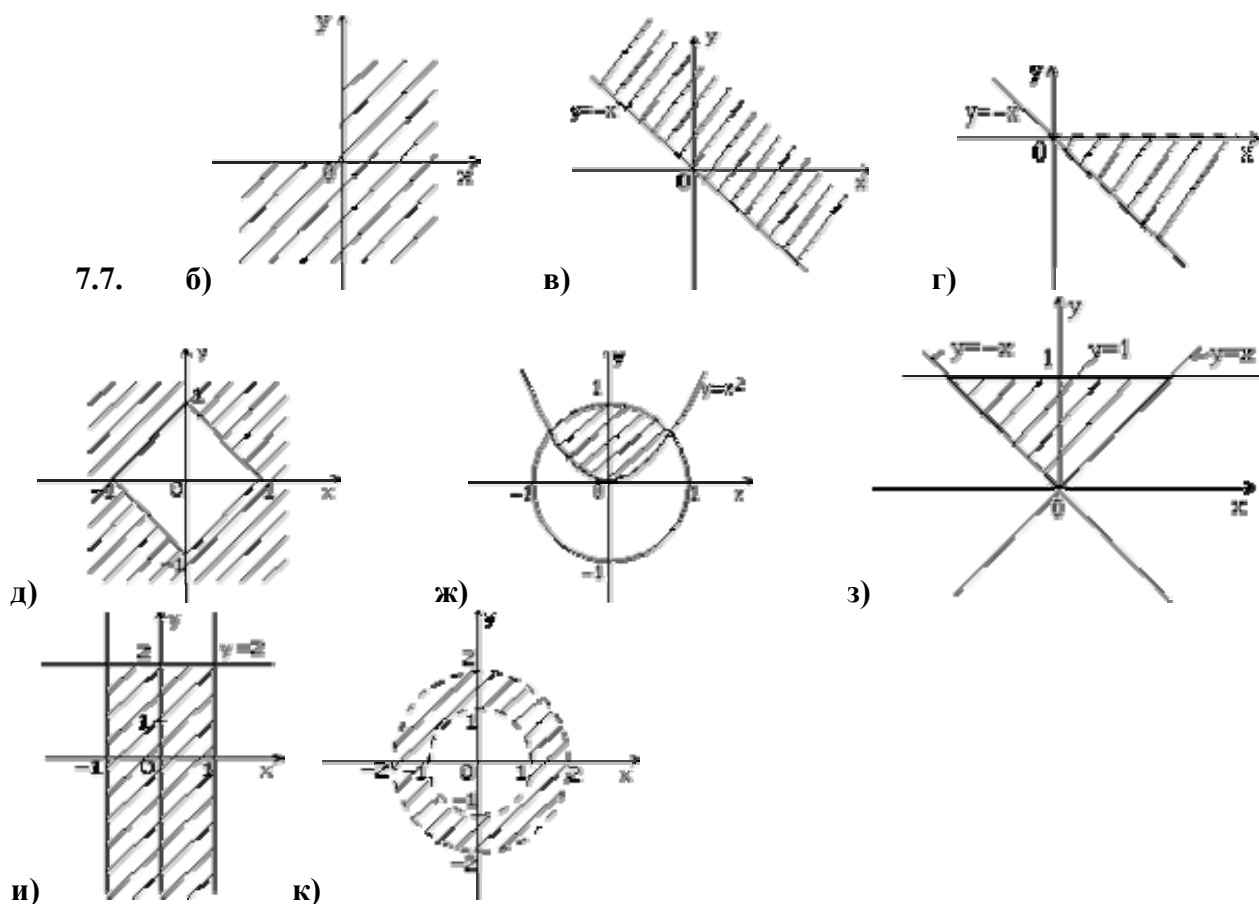
7.3. б) 16; в) 5,5; г) -256; д) 3; е) -2; ж) 1.8.

7.4. б) среднее арифметическое: $(a+b+c+d)/4$; среднее геометрическое: $(a*b*c*d)^{(1/4)}$; в) $\sqrt{x*x+y*y}$; г) $\sin(x*3.14/180)$; д) $6*a*a$; е) $\sqrt{3}*a/2$; ж) абсцисса: $(c1*b2-c2*b1)/(b1*a2-b2*a1)$; ордината: $(c2*a1-c1*a2)/(b1*a2-b2*a1)$.

7.5. б) нет; в) да; г) да; д) да; е) нет;

7.6.

б) $(x < a) \text{ или } (x > b)$;
в) $((x \geq a) \text{ и } (x \leq b)) \text{ или } ((x \geq c) \text{ и } (x \leq d))$;
г) $((x < a) \text{ или } (x > b)) \text{ и } ((x < c) \text{ или } (x > d))$;
д) $\text{mod}(k, 2) = 1$;
е) $(\text{mod}(k, 5) = 0) \text{ и } (k > 99) \text{ и } (k < 1000)$;
ж) $(\text{mod}(i, 2) = 1) \text{ и } (\text{mod}(j, 2) = 0)$;
з) $a1*b2 = a2*b1$;
и) $(c < a) \text{ и } (b > a)$;
к) $(a = -b) \text{ или } (a = -c) \text{ или } (a = -d) \text{ или } (b = -c) \text{ или } (b = -d) \text{ или } (c = -d)$;
л) $((\text{mod}(a, 2) = 0) \text{ и } (\text{mod}(b, 2) = 0)) \text{ или } ((\text{mod}(a, 2) = 0) \text{ и } (\text{mod}(c, 2) = 0))$
или $((\text{mod}(b, 2) = 0) \text{ и } (\text{mod}(c, 2) = 0))$;
м) $(a > 0) \text{ и } (b > 0) \text{ и } (c > 0) \text{ и } (a+b > c) \text{ и } (a+c > b) \text{ и } (b+c > a)$;
н) $((a1*b2 = a2*b1) \text{ и } (a1*c2 = a2*c1)) \text{ или } ((a1*c2 = a2*b1) \text{ и } (a1*b2 = a2*c1))$
или $((a1*c2 = b2*b1) \text{ и } (a1*a2 = b2*c1)) \text{ или } ((a1*a2 = b2*b1) \text{ и } (a1*c2 = b2*c1))$
или $((a1*a2 = c2*b1) \text{ и } (a1*b2 = c2*c1)) \text{ или } ((a1*b2 = c2*b1) \text{ и } (a1*a2 = c2*c1))$;
о) $(y > 5-5*x) \text{ и } (y < 5-x) \text{ и } (y > 0)$;
п) $(y < 5-5*x) \text{ или } (y > 5-x) \text{ или } (y < 0)$;
р) $(a = b) \text{ и } (c = d) \text{ и } (b = c)$.



7.8.

- а) $(y > 1 - x)$ и $((y <= 0) \text{ или } (x <= 0))$;
 б) $(y < 1)$ и $(y >= x)$ и $(y >= -x)$ (вариант ответа: $(y < 1)$ и $(y >= \text{abs}(x))$);
 в) $(\text{abs}(x) <= 1)$ и $(\text{abs}(y) < 1)$;
 г) $((x-1) * (x-1) + y * y <= 4)$ и $(y <= 3 - x)$ и $(y >= x - 3)$;
 д) $(\text{abs}(x) <= 3)$ и $(\text{abs}(y) <= 3)$ и $(x * x + y * y >= 9)$ и $((x >= 0) \text{ или } (y <= 0))$;
 е) $(\text{abs}(x) + \text{abs}(y) <= 2)$ и $(\text{sign}(x) <> \text{sign}(y))$ или $(x * x + y * y <= 4)$
 и $((y >= 2 - x) \text{ или } (y <= -x - 2))$;
 ж) $((y >= x * x) \text{ или } (y <= -x * x))$ и $((x >= y * y) \text{ или } (x <= -y * y))$;
 з) $((x+2) * (x+2) + y * y <= 4)$ и $((x+2) * (x+2) + y * y >= 1)$ или $(x >= -1)$;
 и) $((y <= 0) = (y >= -x))$ или $((x >= 0) = (y >= x))$ и $(x * x + y * y <= 1)$.

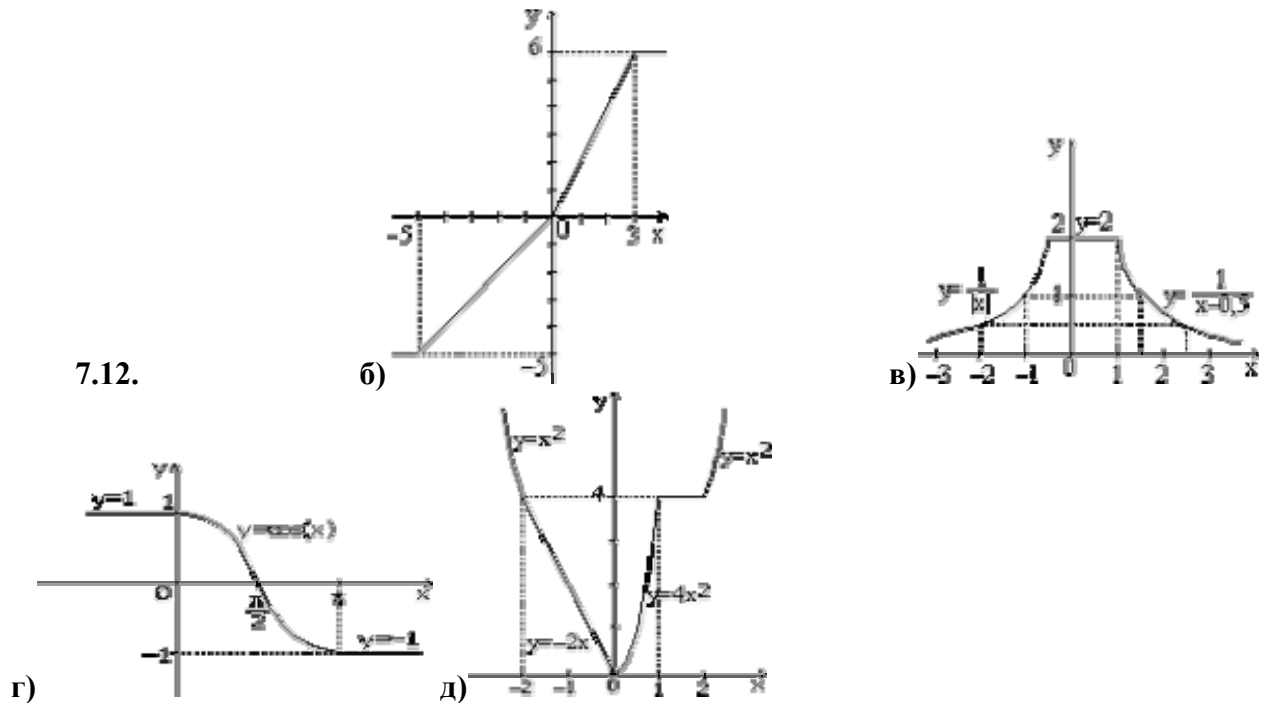
7.9. б) $a=-19$; $b=-114$; $c=17$; в) $a=3$; $b=1,875$; $c=15$; г) $a=7$; $b=3$;
 $c=735$; д) $a=10$; $b=2$; $c=120$; е) $a=4$; $b=7$; $c=3$; ж) $a=16$; $b=100$,
 $c=11200$.

7.10. б) $c:=x[1]$; $x[1]:=x[2]$; $x[2]:=c$; в) $a[i]:=(a[i-1]+a[i+1])/2$; $a[i+1]:=0$; $a[i-1]:=a[i-1]+0.5$; г) $u:=\max(\max(x, y), z) + \min(\min(x-z, y+z), \min(y, z))$.

7.11.

- а) **если** $x \leq -100$
то $y := \text{sign}(x) * \text{abs}(x) ** (1/7)$
иначе если $x < 100$
то $y := \text{sign}(x) * \text{abs}(x) ** (1/3)$
иначе $y := \text{sqrt}(x)$
все
все
- б) **если** $x*x+y*y \leq 1$
то $z := x*x+y*y$
иначе если $y > x$
то $z := x+y$
иначе $z := 0.5$
все
все
- в) **если** $x < 0$
то $z := \lg(-x)$
иначе $z := \text{sqrt}(x+1)$
все
если $z \geq 0$
то $F := 2*z+1$
иначе $F := \sin(z)$
все
- г) **выбор**
при $c=0$: $z:=1$
при $c=1$: $z:=x$
при $c=2$: $z:=3*x*x - 1/2$
при $c=3$: $z:=x*x*x - 3*x/2$
иначе $z:=2*x**4 - 3*x/2$
все
- д) **если** $\text{abs}(x) + \text{abs}(y) < r$
то $z := \text{sqrt}(x*x+y*y)$
иначе $z := \max(\text{abs}(x), \text{abs}(y))$
все
- е) **если** $x > 1$
то если $y > 1$
то $v := x+y$
иначе $v := x-y$
все
иначе если $y > 0$
то $v := y-x$
иначе $v := -x-y$
все
все
- ж) **если** $(x-a)**2 + (y-b)**2 < r*r$
то $z := \text{abs}(x) + \text{abs}(y)$
иначе $z := x+y$
все

7.12.



7.13. б) 81; в) 21; д) 11; е) 44.

7.14. б) 0; в) 13; д) 52; е) 14.

7.15.

- б) **алг** Треугольник1 (**арг** вещ a, b, UголC, **рез** вещ c, UголA, UголB, S)
нач
 ввод a, b, UголC
 $c := \sqrt{a^2 + b^2 - 2 * a * b * \cos(UголC)}$
 $UголA := \arcsin(a * \sin(UголC) / c)$
 $UголB := \arcsin(b * \sin(UголC) / c)$
 $S := b * c * \sin(UголA) / 2$
 вывод c, UголA, UголB, S
кон
- в) **алг** Треугольник2 (**арг** вещ a, b, c, **рез** вещ Radius, UголA)
нач вещ p
 ввод a, b, c
 $p := (a + b + c) / 2$
 $UголA := 2 * \arctg(\sqrt{(p - b) * (p - c) / (p * (p - a))}) * 180 / 3.14$
 $Radius := a * b * c / (4 * \sqrt{p * (p - a) * (p - b) * (p - c)})$
 вывод Radius, UголA
кон
- г) **алг** Объем и Площадь Пирамиды (**арг** вещ a, UголAGrad, **рез** вещ V, S)
нач вещ H, SBase, UголARad
 | H - высота пирамиды; SBase - площадь основания
 ввод a, UголAGrad
 $UголARad := UголAGrad * 3.14 / 180$
 $SBase := a * a * \sqrt{3} / 4$
 $H := a * \sqrt{3} / 6 * \tg(UголARad)$
 $V := SBase * H / 3$
 $S := SBase * (1 + 1 / \cos(UголARad))$
 вывод V, S
кон
- д) **алг** Объем и Площадь конуса (**арг** вещ RBig, RSmall, Uгол, **рез** вещ V, S)
нач вещ H, L
 ввод RBig, RSmall, Uгол
 $H := (RBig - RSmall) * \tg(Uгол)$
 $L := (RBig - RSmall) / \cos(Uгол)$
 $V := 1 / 3 * 3.14 * H * (RSmall^2 + RBig^2 + RSmall * RBig)$
 $S := 3.14 * L * (RBig + RSmall)$
 вывод V, S
кон
- е) **алг** Параметры пирамиды (**арг** вещ a, UголA, **рез** вещ V, S, Section)
нач вещ H
 ввод a, UголA
 $H := a * \sqrt{2} / 2 * \tg(UголA)$
 $V := 1 / 3 * a * a * H$
 $Section := a * H * \sqrt{2} / 2$
 $S := a * a * (1 + \sqrt{2 * \tg(UголA)^2 + 1})$
 вывод V, S, Section
кон
-

7.16.

- б) **алг** Количество положительных (**арг** вещь a, b, c , **рез** цел k)
надо | k - количество положительных чисел среди чисел a, b, c
нач
 ввод a, b, c ; $k := 0$
 если $a > 0$
 то $k := k + 1$
 все
 если $b > 0$
 то $k := k + 1$
 все
 если $c > 0$
 то $k := k + 1$
 все
 вывод k
кон
- в) **алг** Преобразование (**арг** **рез** вещь a, b)
надо | меньшее из a, b увеличено вдвое
нач
 ввод a, b
 если $a > b$
 то $b := b * 2$
 иначе $a := a * 2$
 все
 вывод a, b
кон
- г) **алг** Подобие треугольников (**арг** вещь a, b, c, d , **рез** лог $Otvet$)
дано | a, b и c, d - катеты двух треугольников
надо | $Otvet = \text{да}$, если треугольники подобны
нач
 ввод a, b, c, d
 если $(a * d = c * b)$ **или** $(a * c = d * b)$
 то $Otvet := \text{да}$
 иначе $Otvet := \text{нет}$
 все
 вывод $Otvet$
кон
- д) **алг** Точки (**арг** вещь $x_A, y_A, x_B, y_B, x_C, y_C$, **рез** лит $Otvet$)
нач вещь $DistA, DistB, DistC$
 ввод $x_A, y_A, x_B, y_B, x_C, y_C$
 $DistA := \sqrt{x_A^2 + y_A^2}$
 $DistB := \sqrt{x_B^2 + y_B^2}$
 $DistC := \sqrt{x_C^2 + y_C^2}$
 если $(DistA < DistB)$ **и** $(DistA < DistC)$
 то $Otvet := \text{"Это точка A"}$
 иначе если $DistB < DistC$
 то $Otvet := \text{"Это точка B"}$
 иначе $Otvet := \text{"Это точка C"}$
 все
 все
 вывод $Otvet$
кон
- е) **алг** Принадлежность кольцу (**арг** вещь x, y, r_1, r_2 , **рез** лог $Otvet$)
дано | $r_2 > r_1$
надо | $Otvet = \text{да}$, если точка (x, y) принадлежит кольцу
 | с внутренним радиусом r_1 и внешним радиусом r_2
нач

```
ввод x, y, r1, r2
если (x*x+y*y<=r2*r2) и (x*x+y*y>=r1*r1)
    то Otvet:=да
    иначе Otvet:=нет
все
вывод Otvet
кон
```

```
ж) алг Упорядочение по возрастанию(арг рез вещ a, b, c)
    надо | числа a, b, c упорядочены по возрастанию
    нач вещ t
    ввод a, b, c
    если a>b
        то t:=a; a:=b; b:=t | меняются местами значения a и b
    все
    если a>c
        то t:=a; a:=c; c:=t | меняются местами значения a и c
    все
    если b>c
        то t:=b; b:=c; c:=t | меняются местами значения b и c
    все
    вывод a, b, c
кон
```
