**Муниципальное общеобразовательное учреждение**

 **"Лебяженский центр общего образования"**

**Проектная работа на тему:**

**«КуМир в школе»**

**Работу выполнила: Кононова Мария 9 класс**

**Руководитель работы: Ковалева Инна Борисовна**

 **2020**

**Паспорт проектной работы**

|  |  |
| --- | --- |
| Название проекта  | «Решение простейших математических задач в системе программирования «КуМир»» |
| Руководитель проекта и консультант  | Ковалева Инна Борисовна |
| Образовательная область  | Информатика |
| Состав проектной группы  | Кононова Мария Ивановна |
| Тип проекта (информационный, исследовательский, ролевой, практико-ориентированный, творческий)  | информационный  |
| Заказчик проекта (гимназия, родители, учитель-предметник)  |  учитель-предметник |
| Цель проект  | * научиться решать простейшие математические задачи в системе программирования Кумир;
* освоить основы программирования при помощи программы КуМир;
* составить сборник решения заданий с помощью программирования для 5-6 класса.
 |
| Задачи проекта  | - изучить систему команд КуМира;- рассмотреть способы решения  задач с использованием разных типов алгоритма (линейный, ветвление, циклический)- составить подборку задач по математике 5-6 класса, которые можно решить в системе Кумир, представить тексты всех задний в электронной форме.- решить задачи в системе программирования Кумир. |
| Предполагаемый продукт(ы) проекта. | Сборник алгоритмов (программ) по решению математических задач. |
| Этапы работы над проектом. | Подготовительный: - определение темы проекта; - уточнение цели и задач; - определение источников информации. Основной этап: - распределение обязанностей по времени;- определение шагов действий;- поиск и сбор информации с помощью литературы, средств массовой информации, сети интернета, собственного опыта и исследования. Заключительный: - анализ полученной информации и её систематизации - подведение выводов; - изготовление буклетов и презентации как продуктов проекта; - защита проекта; - обсуждение результатов работы. |
| Образовательные и культурно-просветительские учреждения, на базе которых выполнялся проект. |  Лебяженский Центр Общего Образования. |

**Содержание.**

1. Введение. ........................................................................................5стр
2. Основная часть.
	1. Изучаем алгоритмы. Алгоритмы и исполнители. ..............6стр
	2. Система программирования КуМир. Простые команды. ...7стр
	3. Величины в алгоритмах. Описание переменных. ..............8стр
3. Заключение. ....................................................................................11стр
4. Список литературных источников. ……………..……………..............12стр
5. Приложение. …………...…………………………………………………………..13стр

**Актуальность:**

Информационные технологии занимают все более значимую роль в человеческом обществе. Они проникли во все сферы деятельности, для упрощения и системизации работы человека.

Программирование предполагает собой такое представление, что для многих людей не обладает совершенно никаким практическим значением. Безусловно, почти любой индивид знаком с данным термином, однако при этом некоторые даже далеко не представляют для себя, чем оно является на самом деле.

В наш информационный век профессия программиста является не только популярной, но и очень востребованной. Само название профессии происходит от слова «программа», оно здесь ключевое. Мы, как и многие современные школьники, тоже мечтаем быть программистами. Работа любого программиста начинается с написания первой простейшей программы. Создавать свои программы, и быть с компьютером на «ты», что может быть интереснее. Но так ли проста работа программиста, легко ли писать программы? Попробуем с этим разобраться.

**Проблема:** Можно ли простейшие алгебраические задачи решить средствами программирования?

**Цель:** научиться решать простейшие математические задачи в системе программирования Кумир; составить электронный сборник заданий для 5-6 класса.

**Задачи:**

* Изучить систему команд Кумира;
* Рассмотреть способы решения задач с использованием разных типов алгоритма (линейный, ветвление, циклический)
* Составить подборку решения задач по математике 5-6 класса, которые можно решить в системе Кумир.
* Решить задачи в системе программирования Кумир.

**Методы исследования:** изучение теории по выбранной теме, анализ литературы, практическое применение.

**Материалы и ресурсы:** ватман, бумага формата А4, клей, карандаши.

Программные средства: веб-браузер, текстовый процессор OpenOfficeWritter, презентация.

**Этапы работы над проектом**

**Подготовительный:**

- определение темы проекта;

- уточнение цели и задач;

- определение источников информации.

**Основной этап:**

- распределение обязанностей в группе

- определение шагов действий для каждого участника группы

- высказывание возможных путей разрешения спорных вопросов, обсуждения спорных вопросов;

- поиск и сбор информации каждым участником проекта с помощью литературы, средств массовой информации, сети интернета, собственного опыта и исследования.

**Заключительный:**

- анализ полученной информации и её систематизации

- подведение выводов;

- защита проекта;

-обсуждение результатов работы .

**Основная часть**

**Изучаем алгоритмы.**

## Алгоритмы и исполнители.

Программистов готовят во множестве высших учебных заведений. Хотя в этой сфере успешно работает и множество специалистов с другим образованием, а иногда даже вообще без высшего образования.

Много хороших программистов — самоучки.

Самым важным в процессе обучения является не столько прослушивание лекций, сколько решение задач и участие в различных учебных проектах. Будущему программисту важно научиться думать, находить верные решения, «шевелить мозгами». Информационные технологии развиваются очень быстро, поэтому для таких специалистов необходимы навыки самообразования, чтобы после получения основного диплома идти в ногу со временем и не терять квалификации. Специалисту в этой области нужно знать языки программирования, уметь создавать веб-страницы, владеть иностранным языком (английским как минимум) и постоянно совершенствовать свои знания, так как через несколько лет он может оказаться не удел. Не все программисты создают сайты. Есть те, которые разрабатывают программы специально для работы конкретного предприятия (прикладные программисты).

 Нынешнее программирование многогранно и используется в таких важных сферах как строительство, бизнес и экономика, медицина, биология и физика. Большой процент физического труда в промышленности заменен на машинный и роботизированный труд, который управляется посредством программного обеспечения, что обеспечивает существенный прирост скорости, точности операций и эффективности производства. Такое богатство разнообразия применений обеспечивается солидным выбором языков программирования, у каждого из которых есть свои плюсы и минусы.

Все, что бы мы ни делали, имеет какую-то цель. Не всегда ее удается достигнуть, но для того чтобы это было возможно необходимо как следует сформулировать желаемый для себя результат, а потом продумать четкий план его достижения. Часто план сформулирован в виде предписаний или инструкций.

**Алгоритм – понятное и точное предписание исполнителю выполнить конечную последовательность команд, приводящую к достижению результата.**

 **Исполнитель** – тот объект или субъект, для управления которым составлен алгоритм.
*Характеристики исполнителя:*

* *СКИ —* система команд исполнителя – вся совокупность команд, которые исполнитель умеет выполнять.
* *среда* – обстановка, в которой функционирует исполнитель.

### Свойства алгоритма:

* *Понятность* – алгоритм должен быть составлен только из команд, входящих в СКИ.
* ​ *Дискретность* (детализация) – алгоритм разбивается на отдельные элементарные шаги, которые могут быть исполнены при помощи СКИ.
* *Однозначность* (определенность или детерминированность) – каждый шаг алгоритма имеет единственность толкования выполнения действия и порядка их выполнения.
* ​ *Результативность* (конечность) – выполнение алгоритма должно приводить к результату за конечное число шагов.
* ​ *Массовость* – возможность применения алгоритма к классу однотипных задач, различающихся исходными данными.

Определенная последовательность действий исполнителя всегда применяется к некоторым исходным данным. Например: для приготовления пирога нужны соответствующие продукты, для решения математической задачи - решение квадратного уравнения – нужны числовые данные (значения его коэффициентов). Необходимый и достаточный набор данных для решения поставленной задачи (получения искомого результата) называется полным набором данных.

### Способы записи алгоритмов.

На практике наиболее распространены следующие формы представления алгоритмов:

* словесная (запись на естественном языке);
* графическая (изображения из графических символов блок-схемы);
* 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](https://wiki.lyc84.ru/index.php?title=КуМир) (тексты на языках программирования).

#

# Система программирования КуМир.

КуМир (Комплект Учебных МИРов) - система программирования, предназначенная для поддержки начальных курсов информатики и программирования в средней и высшей школе.

Сайт: [http://www.niisi.ru/kumir/](http://www.niisi.ru/kumir/#_blank) и [http://lpm.org.ru/kumir2/](http://lpm.org.ru/kumir2/#_blank)

В системе КуМир используется школьный алгоритмический язык с русской лексикой и встроенными исполнителями Робот и Чертёжник.

При вводе программы КуМир осуществляет постоянный полный контроль ее правильности, сообщая на полях программы об всех обнаруженных ошибках.

При выполнении программы в пошаговом режиме КуМир выводит на поля результаты операций присваивания и значения логических выражений. Это позволяет ускорить процесс освоения азов программирования.

В простейшем случае программа на КуМире выглядит так:

**алг** Первый

**нач**

.

**кон**

После «алг» (алгоритм) пишется имя алгоритма, между «нач» (начало) и «кон» (конец) пишутся команды алгоритма. При написании автоматически создаются отступы вначале строки, это делает программу более наглядной, удалять или пытаться сделать отступ самим не надо.

Приведенный алгоритм «Первый» не будет ничего делать, т. к. между «нач» и «кон» у него нет  команд.

Вот пример уже работоспособного алгоритма:

**алг** Площадь прямоугольника

**нач**

. **вещ** длина, ширина, площадь

. **вывод** "введите значения длины и ширины прямоугольника"

. **ввод** длина, ширина

. площадь := длина \* ширина

. **вывод** "Площадь прямоугольника равна ", площадь

.**кон**

Его уже можно запустить на выполнение, он запросит у пользователя значения длины и ширины, вычислит и напечатает результат вычислений...

# Простые команды

Эти команды используются практически во всех алгоритмах.

* команда описания переменных
* команда присваивания
* команды ввода-вывода

## Величины в алгоритмах.

Для запоминания информации в памяти используют величины.

Компьютер работает с информацией, хранящейся в его памяти. Отдельный информационный объект (число, символ, строка, таблица и пр.) называется величиной.

Величины в программировании, как и в математике, делятся на переменные и константы. Значение константы остается неизменной в течении всей работы программы, значение переменной величины может изменяться.

У каждой переменной есть **имя, тип и текущее значение**.

Прежде чем писать последовательность команд алгоритма, надо определить количество и тип переменных, участвующих в решении задачи.

Для этого из условия задачи выделить:

\* *что дано* (например, сколько чисел и какие они: целые или вещественные)

\* что требуется вывести как *результат*.

Придумывать имена переменным, как и самим алгоритмам, не обязательно, но желательно так, чтобы по ним было понятно назначение переменной в алгоритме. **Имя** – это последовательность слов, разделенных пробелами. Первое слово имени не должно начинаться с цифры. Ни одно из слов не должно быть ключевым словом (уже имеющим значение в АЯ, например: цел, кон и др.)
**В именах можно использовать:**

* буквы (русские и латинские, прописные и строчные)
* цифры
* два специальных знака: @ \_

Примеры возможных имен: m, x2, площадь, погода на завтра, Ноябрь 7, Седьмое ноября, дом\_57б.

Также будьте внимательны при использовании имен, одинаково выглядящих на русском языке и записанных латинскими буквами. Переменные "x" (икс) и "x" (хэ) - это разные переменные.

### Типы переменных

Существуют три основных типа величин, с которыми работает компьютер: *числовой, символьный и логический*.

Тип величины определяет какие значения она может принимать и какие действия с ней можно выполнять. В зависимости от типа переменной в памяти компьютера будет выделена определенная область. В КуМире числовые типы бывают двух видов: *целочисленные* и *вещественные*.

**Числовые типы**

* **цел** - целые числа от -2147483647 до 2147483647
* **вещ** - действительные числа от -1.797693 × 10308 до 1.797693 × 10308

**Текстовые типы**

* **сим** - один любой символ
* **лит** - строка символов

## Описание переменных

Для того чтобы компьютер мог работать с величиной, нужно указать тип и имя величины, например «цел n». Такое указание называется *описанием величины*.

**алг**

**нач**

 **цел** а, б

 **вещ** частное

...

**кон**

## Команда присваивания

Для того чтобы запомнить или изменить значение величины есть специальная команда — команда присваивания, которая записывается в виде:

**имя величины := выражение**

Например:

частное := а/б

с:= div(а,б)

k:= sqrt(a)

## Команда вывода

Позволяет выводить на экран текстовые сообщения, значения переменных и результаты выражений.

**вывод** "тексты", имена величин, выражения, **нс**

Обратите внимание:

* текстовые сообщения берутся в кавычки;
* имена переменных и выражения перечисляются через запятую без кавычек;
* **нс** - команда перехода на новую строку

## Команда ввода

Позволяет вводить с клавиатуры значения переменных перечисленных в этой команде

**ввод** а, б

**Заключение**

**Вывод:**

Изучение истории языков программирования, их разнообразия и особенностей позволяет сделать правильный выбор при выборе языка для решения определенной задачи.

Если проследить эволюционный путь развития языков программирования, то можно заметить аналогию с развитием живого мира. Когда сначала существовали лишь простые организмы (и языки программирования), а потом появились более сложные.

Саморазвитие очень важно для достижения новых высот, а программирование является очень приятным, удобным, а главное — эффективным способом саморазвития.

В ходе работы над проектом я научилась планировать свою деятельность и распределять время.

Выяснила, что такое алгоритм решения задачи, познакомились с системой программирования КуМир и изучила ее команды, составила свой сборник заданий, на основе учебников математики 5-6 класса и сопроводила его подробным способом решения каждой задачи.

**Практическая значимость проекта:** разработанный сборник заданий можно использовать на уроках информатики как дидактический материал, а также для самостоятельного изучения курса программирования в среде КуМир.

**Литературные источники**

1. Сайт К. Полякова. Практикум по программированию в среде КумМир. URL: <http://kpolyakov.spb.ru/school/kumir.htm>
2. Программирование в среде КуМир. URL: <http://pskumir.blogspot.ru/>
3. Wiki. Изучаем алгоритмы. URL: <https://wiki.lyc84.ru/index.php>?
4. Алгоритмизация в среде КуМир. URL: <https://edusar.soiro.ru/course/view.php?id=475>
5. Практикум по программированию Д. Кириенко. URL: <http://server.179.ru/wiki/?page=DenisKirienko/Kumir>

**Приложение**

**Решаем задачи и примеры.**

 **Сложение.**

**алг** **сложение**
**нач**
**.** **цел** а, б, сумма
**.** **вывод** **"введите два целых числа"**
**.** **ввод** а, б
**.** сумма**:=**а+б
**.** **вывод** **"Сумма равна "**, а,**"+"**,б,**"="**,сумма
**кон**

 **Вычитание.**

**алг вычитание
нач
. цел а, б, разность
. вывод "введите два целых числа"
. ввод а, б
. разность:=а-б
. вывод "Разность равна ", а,"-",б,"=",разность
кон**

**Умножение.**

**алг умножение
нач
. цел а, б, произвидение
. вывод "введите два целых числа"
. ввод а, б
. произвидение:=а\*б
. вывод "Произведение равно ", а,"\*",б,"=",произвидение
кон**

**Деление.**

**алг деление
нач
. вещ а, б, частное
. вывод "введите три числа"
. ввод а, б
. частное:=а/б
. вывод "Частное равно ", а,"/",б,"=",частное
кон**

**Периметр прямоугольника.**

**алг Периметр прямоугольника
нач
. цел а, б, в, *P* |а-длина б-ширина в-2 P-периметр
. вывод "введите три целых числа а, б, в"
. ввод а, б, в
. *P*:=(а+б)\*в
. вывод "Периметр равен ", " (", а, "+", б, ")", "\*", в, "=", *P*
кон**

**Площадь прямоугольника .**

**алг площадь прямоугольника
нач
. цел а, б, *S*
. ввод а, б
. *S*:=а\*б
. вывод "площадь равна ", *S*
кон**

**Объём параллелепипеда.**

**алг объем
нач
. цел *a*, *b*, *h*, *V*
. вывод "Введите три целых числа"
. ввод *a*, *b*, *h*
. *V*:=*a*\**b*\**h*
. вывод "Объём равен ", *a*,"\*",*b*,"\*",*h*,"=",*V*
кон**

**Немного задач.**

**Скорость сближения.**

**алг скорость сближения
нач
. вещ а, б, р, ч|
. вывод "введите три целых числа"
. ввод а, б, р
. ч:=р/(а+б)
. вывод "СКОРОСТЬ СБЛЖЕНИЕ =", ч
кон**

**Нахождение скорости.**

**алг задача
нач
. цел а, б, в, г | а-скорость лодки, б-скорость лодки, в- разность скоростей, г-скорость гидроцикла=?
. вывод "введите три целых числа"
. ввод а, б, в
. г:=(а+б)\*в
. вывод "скорость гидроцикла-" , "(", а, "+", б, ")", "\*", в, "=", г
кон**

**Найти сумму n-го числа слагаемых.**

 **алг сумма
нач
. цел *i*,*n* |i - счетчик , n - число слагаемых
. вещ *s* | s -сумма
. вывод 'введите число слагаемых n =', нс
. *s*:=0
. ввод *n*
. нц *n* раз
. . вывод 'введите слагаемое',нс
. . ввод *i*
. . *s*:=*s*+*i*
. кц
. вывод 'сумма ',*n*,'-х слагаемых =',*s*:0:2
кон**

**Найти произведение n-го числа множителей.**

 **алг произведение
нач
. цел *i*,*n* |i - счетчик , n - число множителей
. вещ *p* | p -произведение
. вывод 'введите число множителей n =', нс
. *p*:=1
. ввод *n*
. нц *n* раз
. . вывод 'введите множитель',нс
. . ввод *i*
. . *p*:=*p*\**i*
. кц
. вывод 'произведение ',*n*,' множителей =',*p*:0:2
кон**

**Сравнить а) два числа и вывести их в порядке возрастания;**

**алг** **сравнить два числа**
**нач**
**.** **цел** *a*,*b* **| a,b -числа**
**.** **вывод** **'введите два числа'**, **нс**
**.** **ввод** *a*,*b*
**.** **если** *a*>*b*
**.** **.** **то** **вывод** **'число '**,*a*,**' > числа '**,*b* **иначе**
**.** **.** **.** **вывод** **'число '**,*a*,**' < числа '**,*b*
**.** **все**
**кон**

**Сравнить б) три числа и вывести их в порядке возрастания;**

**алг** **сравнить три числа**
**нач**
**.** **цел** *a*,*b*,*c*,*m* **| a,b,c -числа, m - наибольшее число**
**.** **вывод** **'введите три числа'**, **нс**
**.** **ввод** *a*,*b*,*c*
**.** **если** *a*>*b*
**.** **.** **то**
**.** **.** **.** **если** *a*>*c*
**.** **.** **.** **.** **то**
**.** **.** **.** **.** **.** *m***:=***a*
**.** **.** **.** **.** **иначе** *m***:=***c*
**.** **.** **.** **все**
**.** **.** **иначе**
**.** **.** **.** **если** *b*>*c*
**.** **.** **.** **.** **то** *m***:=***b*
**.** **.** **.** **.** **иначе** *m***:=***c*
**.** **.** **.** **все**
**.** **все**
**.** **вывод** **'наибольшое число = '**,*m*
**кон**

**Представить двухзначное число в виде суммы разрядных слагаемых**.

**алг** **сумма разрядных слагаемых**
**нач**
**.** **цел** *x*,*a*,*b***| x - число, a,b - цифры числа**
**.** **вывод** **'введите число'**, **нс**
**.** **ввод** *x*
**.** *a***:=*div***(*x*,**10**)
**.** *b***:=** ***mod***(*x*, **10**)
**.** **вывод** *x*,**'='**,*a*,**'\*10'**,**'+'**,*b*,**'\*1'**
**кон**

**Представить трехзначное число в виде суммы разрядных слагаемых.**

 **алг сумма разрядных слагаемых
нач
. цел *x*,*a*,*b*,*c*| x - число, a,b - цифры числа
. вывод 'введите число', нс
. ввод *x*
. *a*:=*div*(*x*,100)
. *b*:= *div*(*mod*(*x*, 100),10)
. *c*:=*mod*(*x*,10)
. вывод *x*,'=',*a*,'\*100','+',*b*,'\*10','+',*c*,'\*1'
кон**

**Округлить число до тыс., сот., дес., до старшего разряда.**

**алг округлить число
нач
. цел *x*,*a*,*b*,*c*,*d*| x - число, a,b,c,d - цифры числа
. вывод 'введите число', нс
. ввод *x*
. *a*:=*div*(*x*,1000)
. *b*:= *mod*(*div*(*x*, 100),10)
. *c*:=*div*(*mod*(*x*,100),10)
. *d*:=*mod*(*x*,10)
. если *b*>5 или *b*=5
. . то
. . . вывод *x*,'~',(*a*+1)\*1000, 'до тысяч',нс
. . иначе
. . . вывод *x*,'~',*a*\*1000, 'до тысяч',нс
. все
. если *c*>5 или *c*=5
. . то
. . . вывод *x*,'~',*a*\*1000 +(*b*+1)\*100,'до сотен',нс
. . иначе
. . . вывод *x*,'~',*a*\*1000+*b*\*100, 'до сотен',нс
. все
. если *d*>5 или *d*=5
. . то
. . . вывод *x*,'~',*a*\*1000 +*b*\*100+(*c*+1)\*10, 'до десятков',нс
. . иначе
. . . вывод *x*,'~',*a*\*1000+*b*\*100+*c*\*10, 'до десятков', нс
. все
. вывод *x*,'~',*a*\*1000+*b*\*100+*c*\*10+*d*, 'до единиц', нс
кон**

**Сколько существует двузначных чисел, у которых**

**первая цифра больше второй.**

**алг** **двузначные числа**
**нач**
**.** **цел** *i*,*a*,*b*,*s***| i - счетчик, a,b - цифры, s - сумма**
**.** *i***:=10**
**.** *s***:=0**
**.** **нц** **90** **раз**
**.** **.** *a***:=*div***(*i*,**10**)
**.** **.** *b***:=*mod***(*i*,**10**)
**.** **.** **если** *a*>*b*
**.** **.** **.** **то**
**.** **.** **.** **.** *s***:=***s*+**1**
**.** **.** **все**
**.** **.** *i***:=***i*+**1**
**.** **кц**
**.** **вывод** *s*
**кон**

**Найти сумму 1+2+3+4+5+ …+n-го числа слагаемых (с помощью цикла).**

**алг** **сумма n числа слагаемых**
**нач**
**.** **цел** *i*,*n*,*s***| i - счетчик, n - число слагаемых, s - сумма**
**.** **вывод** **'введите число слагаемых n ='**
**.** **ввод** *n*
**.** *i***:=1**
**.** *s***:=0**
**.** **нц** *n* **раз**
**.** **.** *s***:=***s*+*i*
**.** **.** *i***:=***i*+**1**
**.** **кц**
**.** **вывод** *s*
**кон**

**Найти сумму 1+2+3+4+5+ …+n-го числа слагаемых (с помощью метода Гауса).**

 **алг сумма n числа слагаемых
нач
. вещ *i*,*n*,*s*| i - счетчик, n - число слагаемых, s - сумма
. вывод 'введите число слагаемых n ='
. ввод *n*
. *s*:=((1+*n*)\**n*)/2
. вывод *s*
кон**

**Найдите сумму: а) нечетных чисел 1+3+5+ …+n и б) четных чисел 2+4+6+.. +n(с помощью цикла).**

**алг сумма нечетных\_четных слагаемых
нач
. цел *i*,*n*,*s1*,*s2*| i - счетчик, n - число слагаемых, s1 - сумма четных слагаемых, s2 - сумма нечетных слагаемых
. вывод 'введите число слагаемых n ='
. ввод *n*
. *i*:=1
. *s1*:=0
. *s2*:=0
. нц *n* раз
. . если *mod*(*i*,2)=0
. . . то
. . . . *s1*:=*s1*+*i*
. . . иначе
. . . . *s2*:=*s2*+*i*
. . все
. . *i*:=*i*+1
. кц
. вывод 'сумма четных слагаемых = ',*s1*, нс
. вывод 'сумма нечетных слагаемых = ',*s2*, нс
.
кон**

**Найдите сумму: а) нечетных чисел 1+3+5+ …+n**

**(по формуле см. № 226 учебника 6 кл.).**

 **алг сумма нечетных слагаемых
нач
. цел *i*,*n*,*s2*| i - счетчик, n - число слагаемых, s2 - сумма нечетных слагаемых
. вывод 'введите число слагаемых n ='
. ввод *n*
. *i*:=1
. *s2*:=0
. нц *n* раз
. . если *mod*(*i*,2)<>0
. . . то
. . . . *s2*:=*s2*+1
. . все
. . *i*:=*i*+1
. кц
. вывод 'сумма нечетных слагаемых = ',*s2*\**s2*, нс
.
Кон**

**Определить вид треугольника(если таковой существует): а) (остроугольный, прямоугольный, тупоугольный), если известные градусные меры всех его углов.**

**алг** **вид треугольника по углам**
**нач**
**.** **цел** *a*,*b*,*c***| a,b,c - градусные меры всех углов треугольника**
**.** **вывод** **'введите величины углов треугольника a =,b=,c='**
**.** **ввод** *a*,*b*,*c*
**.** **если** *a*+*b*+*c*=**180** **и** *a*<>**0** **и** *b*<>**0** **и** *c*<>**0**
**.** **.** **то**
**.** **.** **.** **если** *a*>**90** **или** *b*>**90** **или** *c*>**90**
**.** **.** **.** **.** **то**
**.** **.** **.** **.** **.** **вывод** **'треугольник тупоугольный'**
**.** **.** **.** **.** **иначе**
**.** **.** **.** **.** **.** **если** *a*=**90** **или** *b*=**90** **или** *c*=**90**
**.** **.** **.** **.** **.** **.** **то**
**.** **.** **.** **.** **.** **.** **.** **вывод** **'треугольник прямоугольный'**
**.** **.** **.** **.** **.** **.** **иначе**
**.** **.** **.** **.** **.** **.** **.** **вывод** **'треугольник остроугольный'**
**.** **.** **.** **.** **.** **все**
**.** **.** **.** **все**
**.** **.** **иначе**
**.** **.** **.** **вывод'треугольник не существует'**
**.** **все**
**кон**

**Определить вид треугольника(если таковой существует): б) (равнобедренный, равносторонний, разносторонний), если известны длины всех его сторон.**

**алг** **вид треугольника по сторонам**
**нач**
**.** **цел** *a*,*b*,*c***| a,b,c - длины всех сторон треугольника**
**.** **вывод** **'введите длины сторон треугольника a =,b=,c='**
**.** **ввод** *a*,*b*,*c*
**.** **если** *a*<*b*+*c* **и** *b*<*a*+*c* **и** *c*<*a*+*b*
**.** **.** **то**
**.** **.** **.** **если** *a*=*c*=*b*
**.** **.** **.** **.** **то**
**.** **.** **.** **.** **.** **вывод** **'треугольник равносторонний'**
**.** **.** **.** **.** **иначе**
**.** **.** **.** **.** **.** **если** *a*=*b* **или** *b*=*c* **или** *a*=*c*
**.** **.** **.** **.** **.** **.** **то**
**.** **.** **.** **.** **.** **.** **.** **вывод** **'треугольник равнобедренный'**
**.** **.** **.** **.** **.** **.** **иначе**
**.** **.** **.** **.** **.** **.** **.** **вывод** **'треугольник разносторонний'**
**.** **.** **.** **.** **.** **все**
**.** **.** **.** **все**
**.** **.** **иначе**
**.** **.** **.** **вывод'треугольник не существует'**
**.** **все**
**кон**

**Найти НОД(а,в).**

**алг НОД
нач
. цел *a*,*b*,*x*,*y*| a,b - числа
. вывод 'введите числа '
. ввод *a*,*b*
. *x*:=*a*
. *y*:=*b*
. нц пока *a*<>*b*
. . если *a*>*b*
. . . то *a*:=*a*-*b*
. . . иначе *b*:=*b*-*a*
. . все
. кц
. вывод'НОД(',*x*,',',*y*,')=',*a*
кон**