

№ 9

1-15

мая 2011

Основана в 1995 г

inf.1september.ru

УЧЕБНО-МЕТОДИЧЕСКАЯ ГАЗЕТА ДЛЯ УЧИТЕЛЕЙ ИНФОРМАТИКИ

# ИНФОРМАТИК А

/ тема номера:

**ЕГЭ.**

**Алгоритмизация.  
Базовые задачи**



№ 9

издательский дом

**Первое сентября**

1 september.ru

ИНФОРМАТИКА

Индексы подписки Почта России – 79006 (инд.); – 79574 (орг.) Роспечать – 32291 (инд.);

32591 (орг.)

## О ТЕМЕ НОМЕРА

► Базовые алгоритмические задачи ЕГЭ рассматриваются на основе Кодификатора (он имеет длинное название, здесь мы не приводим его полностью — все и так понимают, о чем речь).

Перечень задач в Кодификаторе имеет как минимум два применения. Во-первых, это формальный ориентир: этим задачам точно надо научить, их спросят. Во-вторых, это ориентир содержательный — в нем содержится то содержательное алгоритмическое ядро, которому надо научить не “для галочки”, а по делу. Конечно, до ЕГЭ осталось не так много времени (а пробные сдачи и вовсе уже прошли), но маловероятно, что перечень основных задач претерпит существенные изменения, например, к следующему году (он же содержательный!). Поэтому, надеемся, этот номер надолго останется в вашей методической копилке.

## В НОМЕРЕ

- 3** **НОВОСТЬ № 1**  
Сервер! Лежать!
- 4** **НЕ В ТЕМЕ**  
Wiki: свобода создавать книги
- 8** **ТЕМА НОМЕРА**  
Задачи по программированию в ЕГЭ по информатике.  
Избранное
- 16** **МНЕНИЯ**  
КуМир и школьная информатика
- 18** **ТЕМА НОМЕРА**  
Задачи на обработку строк в ЕГЭ по информатике
- 22** **ГАЗЕТА ДЛЯ ПЫТЛИВЫХ УЧЕНИКОВ И ИХ ТАЛАНТЛИВЫХ УЧИТЕЛЕЙ**  
“В мир информатики” № 163
- 31** **ИНФОРМАЦИЯ**  
Продолжается подписка на электронные версии предметно-методических журналов ИД “Первое сентября»
- 32** **ИНФОРМАТИКА В ЛИЦАХ**  
Джон Соча: известный неизвестный

## НА ДИСКЕ



Материалы к этому номеру будут размещены на диске, вложенном в № 10.

## ЭЛЕКТРОННЫЕ МАТЕРИАЛЫ:

- | Реализации всех программ из номера на КуМире, Паскале и Бейсике.
- | Презентации к урокам по КуМиру, предоставленные К.Ю. Поляковым.
- | Презентации к статьям номера.

## ИНФОРМАТИКА

ПОДПИСНЫЕ ИНДЕКСЫ: по каталогу «Газеты. Журналы» агентства «Роспечать» – 32291 (инд); – 32591 (орг.); по каталогу «Почта России» – 79006 (инд); – 79574 (орг.)

<http://inf.1september.ru>

Учебно-методическая газета для учителей информатики  
*Основана в 1995 г.*  
Выходит два раза в месяц

## РЕДАКЦИЯ:

гл. редактор С.Л. Островский  
редакторы

Е.В. Андреева,

Д.М. Златопольский

(редактор вкладки  
“В мир информатики”)

Дизайн макета И.Е. Лукьянов

верстка Н.И. Пронская

корректор Е.Л. Володина

секретарь Н.П. Медведева

Фото: фотобанк Shutterstock

Газета распространяется по подписке

Цена свободная

Тираж 3000 экз.

Тел. редакции: (499) 249-48-96

E-mail: [inf@1september.ru](mailto:inf@1september.ru)

<http://inf.1september.ru>

## ИЗДАТЕЛЬСКИЙ ДОМ

## “ПЕРВОЕ СЕНТЯБРЯ”

## Главный редактор:

Артем Соловейчик  
(Генеральный директор)

## Коммерческая деятельность:

Константин Шмарковский  
(Финансовый директор)

## Развитие, IT

## и координация проектов:

Сергей Островский  
(Исполнительный директор)

## Реклама и продвижение:

Марк Сартан

## Мультимедиа, конференции

## и техническое обеспечение:

Павел Кузнецов

## Производство:

Станислав Савельев

## Административно-

## хозяйственное обеспечение:

Андрей Ушков

## Дизайн:

Иван Лукьянов, Андрей Балдин

## Педагогический университет:

Валерия Арсланьян (ректор)

## ГАЗЕТЫ

## ИЗДАТЕЛЬСКОГО ДОМА

Первое сентября – Е.Бирюкова

Английский язык – А.Громушкина

Библиотека в школе – О.Громова

Биология – Н.Иванова

География – О.Коротова

Дошкольное

образование – М.Аромштам

Здоровье детей – Н.Сёмина

Информатика – С.Островский

Искусство – М.Сартан

История – А.Савельев

Классное руководство

и воспитание школьников –

О.Леонтьева

Литература – С.Волков

Математика – Л.Рослова

Начальная школа – М.Соловейчик

Немецкий язык – М.Бузова

Русский язык – Л.Гончар

Спорт в школе – О.Леонтьева

Управление школой – Я.Сартан

Физика – Н.Козлова

Французский язык – Г.Чесновицкая

Химия – О.Блохина

Школьный психолог – И.Вачков

## УЧРЕДИТЕЛЬ:

## ООО “ЧИСТЫЕ ПРУДЫ”

## Зарегистрировано

ПИ № 77-72230

от 12.04.2001

в Министерстве РФ

по делам печати

Подписано в печать:

по графику 06.04.2011,

фактически 06.04.2011

Заказ №

Отпечатано в ОАО “Чеховский

полиграфический комбинат”

ул. Полиграфистов, д. 1,

Московская область,

г. Чехов, 142300

## АДРЕС ИЗДАТЕЛЯ:

ул. Киевская, д. 24,

Москва, 121165

Тел./факс: (499) 249-31-38

## Отдел рекламы:

(499) 249-98-70

<http://1september.ru>

## ИЗДАТЕЛЬСКАЯ ПОДПИСКА:

Телефон: (499) 249-47-58

E-mail: [podpiska@1september.ru](mailto:podpiska@1september.ru)

Документооборот

Издательского дома

“Первое сентября” защищен

антивирусной программой

Dr.Web



## Сервер! Лежать!

ЖЖ был на продолжительное время выведен из строя посредством так называемой “DDoS-атаки”. Что это такое, знает каждый учитель. Допустим, закончился урок, вам надо поставить оценки, а вокруг вас сгрудились ученики. Одному надо то, другому — это, третьему — отпроситься, четвертому — ..., пятому — ..., шестому — ... Все! “Мозг взорвался!”. Это “на пальцах”.

► Согласитесь, не каждый день и не по каждому случаю Президент страны находит время, чтобы лично отреагировать на рутинное в общем-то событие в Сети — DDoS-атаку на популярный ресурс. Но эхо атаки, которой в конце марта подвергся ЖЖ (LiveJournal), дошло даже до Президента. Так что это было, и не повторится ли произошедшее в дальнейшем?

Сначала ответ на второй вопрос: обязательно повторится! В этом нет практически никаких сомнений. Дальше — больше. Мишенями атак станут (и уже являются) не только популярные и посещаемые ресурсы, что можно рассматривать просто как хулиганство ради “хакерского искусства”, но и сайты коммерческих конкурентов и политических оппонентов. Возможно, скоро услуги “завалю сайт” будут популярнее “раскручу, выведу на первые позиции”.

Но что за безобразие такое, почему даже на государственном уровне так трудно бороться с нападениями из виртуального пространства? Давайте кратко разберемся. Во-первых, нападение нападению рознь. Например, так называемые “дефейсы” популярны достаточно давно, фактически с момента появления Интернета. “Дефейс” в прямом смысле слова — замена главной (обычно главной, но можно любой) страницы на нечто подобное “Здесь был злой хакер Вася”. “Дефейс” практически всегда является результатом прямого взлома атакуемой системы, при котором злоумышленник получает контроль над всем сервером или его частью. 100%-ной гарантии от подобного не даст ни один системный администратор. Даже при аккуратной установке всех патчей безопасности и ответственном подходе к разграничению прав доступа всегда могут найтись новые уязвимости в различном программном обеспечении. На эту тему написаны толстые книжки, а спе-

циалистов по компьютерной безопасности сейчас учат сотням методов взлома систем. Но... все это не имеет прямого отношения к атаке на ЖЖ, поскольку сам ЖЖ не был взломан.

ЖЖ был на продолжительное время выведен из строя посредством так называемой “DDoS-атаки”. Что это такое, знает каждый учитель. Допустим, закончился урок, вам надо поставить оценки, а вокруг вас сгрудились ученики. Одному надо то, другому — это, третьему — отпроситься, четвертому — ..., пятому — ..., шестому — ... Все! “Мозг взорвался!”. Это “на пальцах”.

Более строго DDoS расшифровывается как *Distributed Denial of Service* — распределенная атака, имеющая целью отказ в обслуживании “обычных” пользователей. Эффективные DDoS-атаки требуют участия многих сотен, тысяч и даже десятков и сотен тысяч компьютеров, которые одновременно отправляют запросы на атакуемый ресурс. Почему нужно так много? Ну, во-первых, запросов должно быть много. Во-вторых, одним из эффективных методов противодействия DDoS-атакам является фильтрация трафика. Грубо говоря, мусорные запросы отсекаются неким промежуточным оборудованием и не доходят до атакуемой системы. Если атаковать небольшим числом компьютеров, то запросы от них можно достаточно просто отфильтровать (по IP, иными методами, не суть важно, главное — можно). А вот если атакующих очень много и их невозможно отличить от обычных пользователей, отправляющих обычные запросы, — тогда и возникают настоящие проблемы.

Но неужели найдутся десятки тысяч людей, готовых предоставить свои компьютеры для организации DDoS-атаки? Найдутся. Только они знать ничего про это не знают ☺. Для атак используют так называемые “ботнеты” — сети компьютеров, на которые без ведома пользователей (например, вирусными или троянскими методами) устанавливается специальное программное обеспечение. Не хотите, чтобы вашим компьютером управляли? Своевременно обновляйте антивирус ☺. ◀



## Wiki: свобода создавать книги

Д.Ю. Усенков,  
ст. н. с. Института  
информатизации  
образования  
Российской  
академии  
образования,  
Москва

► Web-технология Wiki, которая является основой функционирования всемирно известной онлайн-энциклопедии “Википедия” (и многих других проектов, реализуемых в рамках нового направления “Web 2”), известна, наверное, почти всем. Она позволяет практически всем желающим читателям “Википедии” принять участие в наполнении этой энциклопедии и в редактировании ее материалов, — причем все правки запоминаются и могут позже обсуждаться и корректироваться другими участниками, а при необходимости неудачные правки могут быть отменены. Другим преимуществом Wiki-движка является его интуитивная понятность (с работой в среде Wiki может справиться даже пользователь, не знакомый с основами разметки текста на языке HTML и с принципами создания web-сайтов) и его гибкость, позволяющая реализовать в Wiki-среде множество удобных возможностей.

Общие принципы работы с “Википедией” известны, наверное, всем:

— при помощи панели поиска (справа вверху) можно искать требуемые статьи либо воспользоваться для этого алфавитным указате-

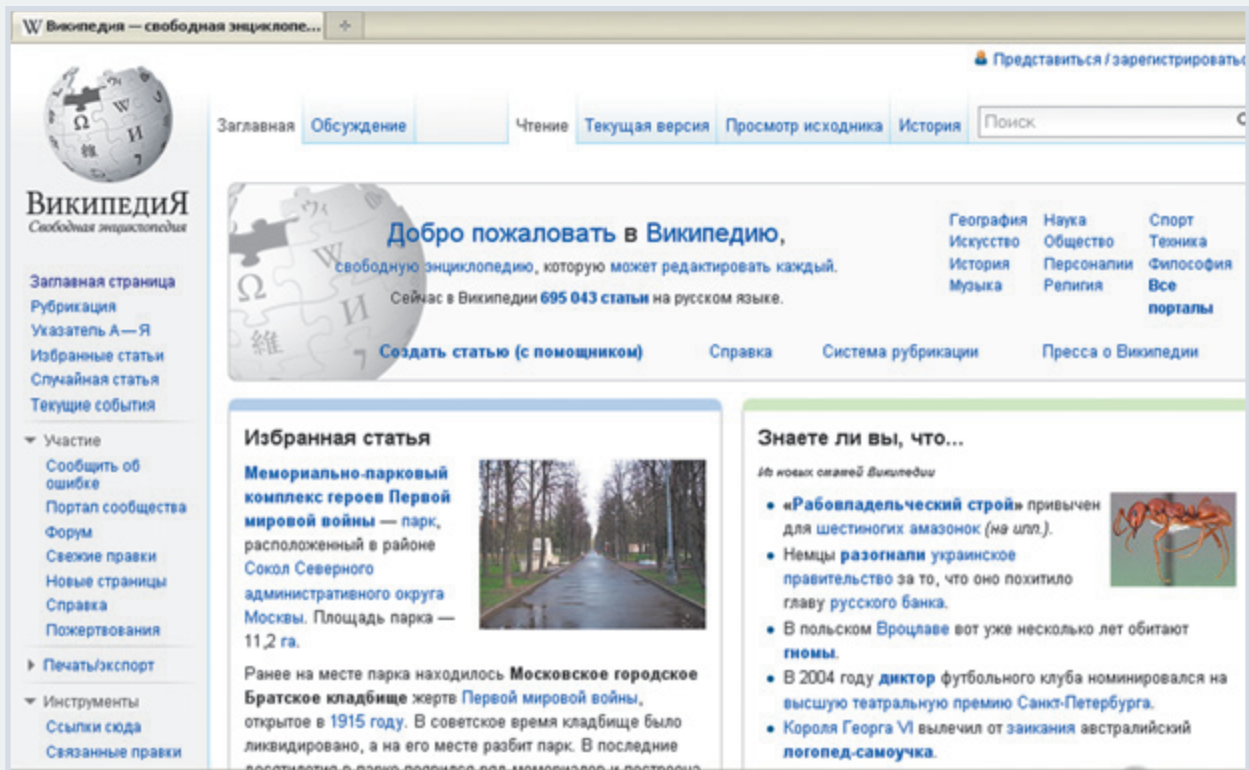
лем и рубрикаторм (их можно раскрыть, выбрав соответствующие ссылки в верхней части меню слева);

— найденный материал можно читать, пользуясь его содержанием и имеющимися в тексте гиперссылками;

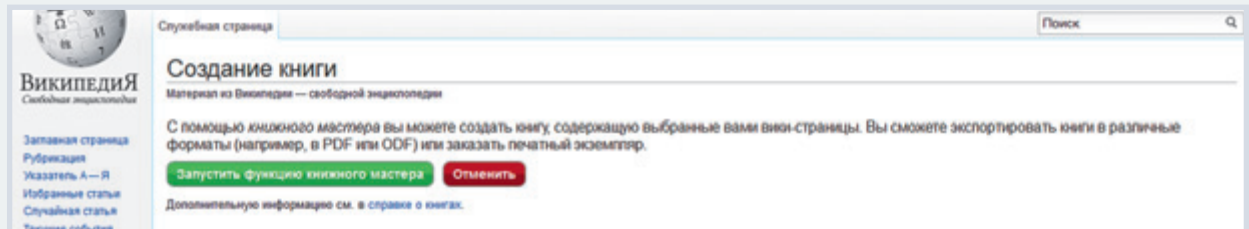
— можно просматривать иллюстрации в более крупном масштабе (щелкнув мышью на иллюстрации; кстати, на открывшейся странице с ней, кроме всего прочего, будет указано, является ли данная иллюстрация свободной для использования, — тогда можно скачать такой рисунок и использовать его в любых своих работах);

— можно перейти на вкладку “Правка” и принять участие в редактировании статьи, и т.д.

Однако среди предоставляемых Wiki-движком возможностей есть и такие, которые при всем их удобстве мало кому известны. Например, если вы захотите переписать на свой компьютер копию тех или иных материалов “Википедии”, то вы можете не просто скопировать их, но и... создать собственную электронную книгу! Для этого нужно в меню слева выбрать пункт “Печать/экспорт”, а затем в списке подразделов этого пункта (если этот список свернут, надо щелчком мыши на пункте “Печать/экспорт” раскрыть его) выбрать пункт “Создать книгу”. В появившейся web-странице нужно затем щелкнуть мышью на кнопке “Запустить функцию книжного ма-

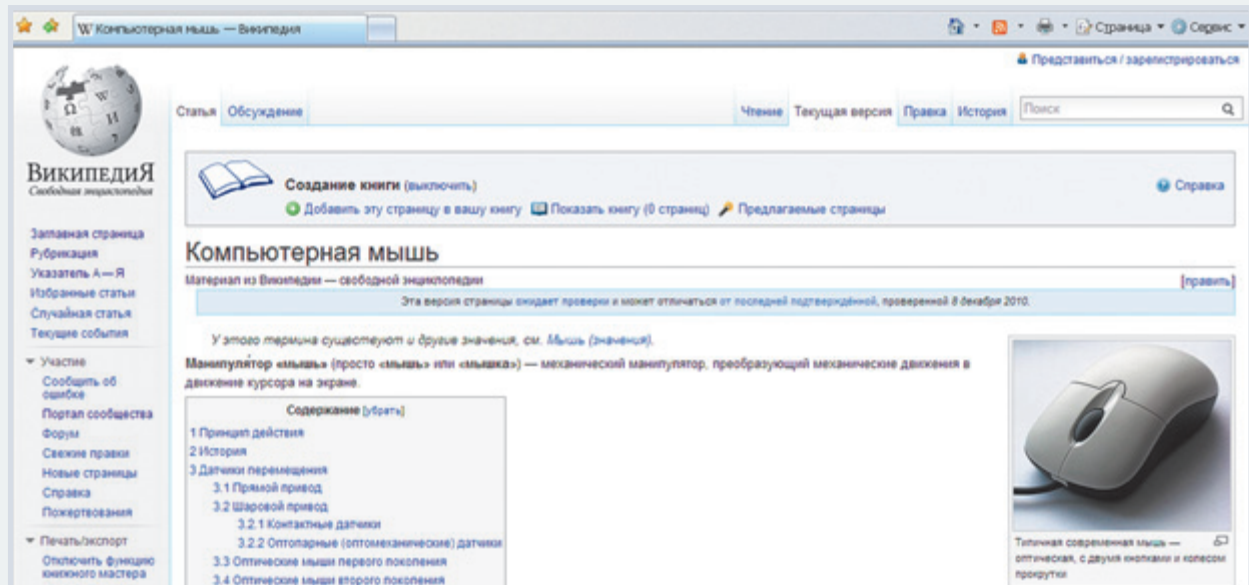


стера”, — тогда выбранная ранее статья “Википедии” появится в браузере вновь, но теперь в ее верхней части вы увидите серую панельку с надписью: “Создание книги”.



С помощью этой панельки Wiki позволяет вам самим создать свою книгу (подборку) из отдельных статей Wiki-сайта.

Когда вы просматриваете какую-либо Wiki-статью и в левой части этой панели имеется надпись-гиперссылка “Добавить эту страницу в вашу книгу”, щелчок мышью на этой надписи добавляет текущую Wiki-статью в формируемую вами подборку. (Для ранее добавленных в книгу статей эта ссылка заменяется на другую, позволяющую удалить данную статью из подборки.)

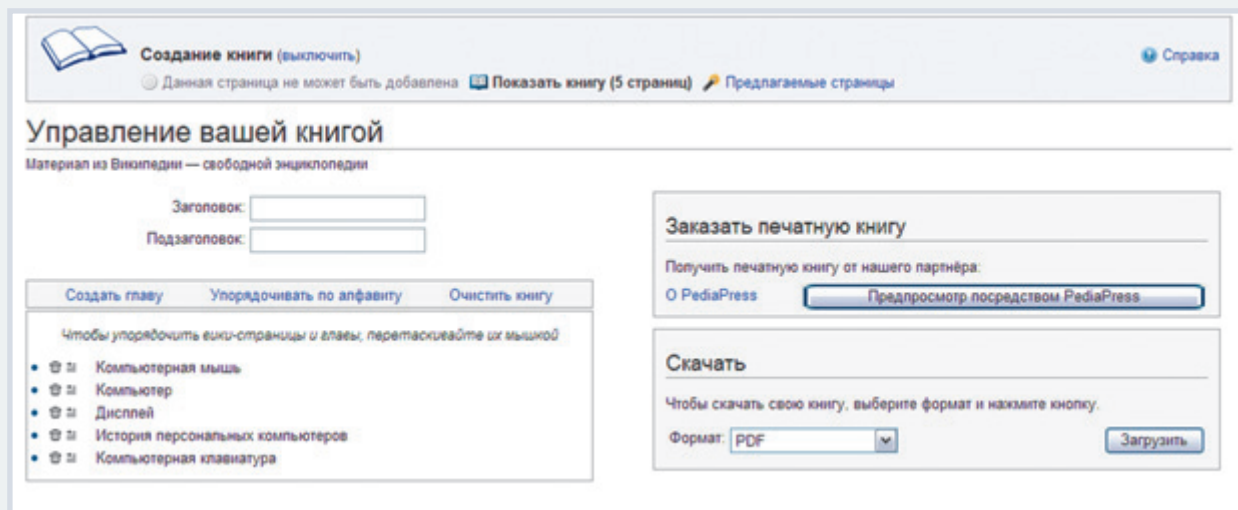


Количество уже имеющихся в книге страниц показано в скобках в средней надписи-гиперссылке “Показать книгу”.

А если щелкнуть на надписи-гиперссылке “Предлагаемые страницы”, то Wiki сама предложит вам список разделов, которые соответствуют вашим интересам (проявившимся в выборе уже добавленных вами статей).

Давайте добавим в создаваемую книгу несколько статей, например, о компьютере, компьютерной клавиатуре, дисплее, компьютерной мыши и об истории компьютеров (все эти статьи можно найти, воспользовавшись панелью поиска вверху справа).

Если теперь выбрать ссылку “Показать книгу”, то в окне браузера появляется специальная страница, посвященная структуре создаваемой подборки.



Здесь можно гибко управлять ее содержанием (которое показано слева): перемещать разделы относительно друг друга, перетаскивая их мышью, создавать новые заголовки глав (чтобы затем перетащить туда “наполнение” в виде соответствующих статей), удалять какие-то разделы и заголовки и т.п.

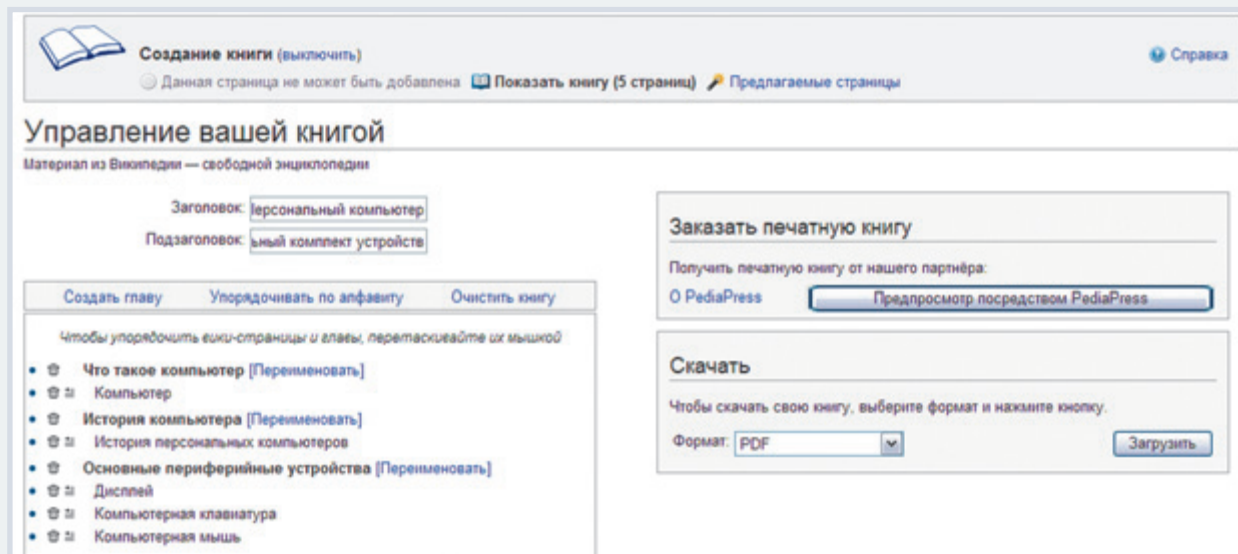
Оформим нашу будущую книгу.

1) Введем в полях “Заголовок” и “Подзаголовок” заглавие книги (например, “Персональный компьютер”) и подзаголовок (“Минимальный комплект устройств”).

2) Воспользовавшись ссылкой “Создать главу” (в “шапке” окна содержания), создадим заголовки глав книги: “Что такое компьютер”, “История компьютера” и “Основные периферийные устройства” (если потребуется, позже можно свободно переименовать эти разделы при помощи соответствующей ссылки, которая выводится рядом с каждым появившимся в содержании названием главы, а любой ненужный заголовок или текст можно удалить из подборки, воспользовавшись кнопкой в виде “мусорной корзины” слева от него).

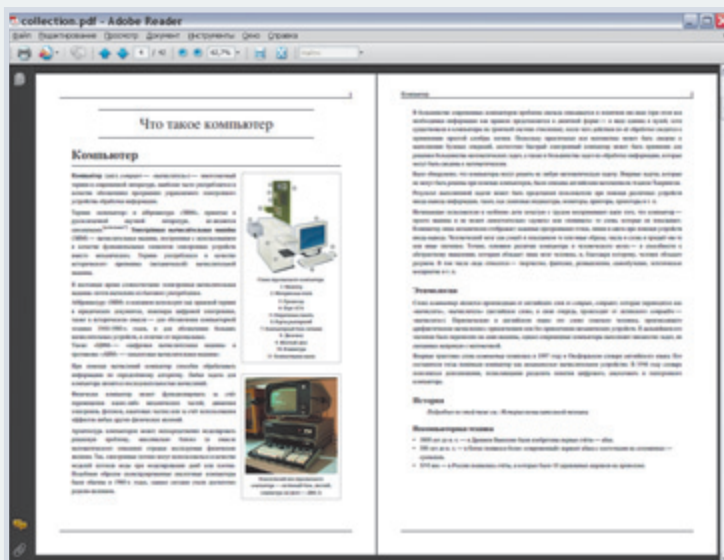
3) Перетаскивая мышью строки содержания, установим в нем нужный порядок следования заголовков глав и “наполним” эти главы содержимым, перетащив под них строки с названиями соответствующих статей “Википедии”.

Например, результат работы может быть таким:



Вот и все. А когда книга сформирована, можно воспользоваться панелью загрузки (справа): выбрать в списке желаемый формат файла (всем известный PDF или его аналог ODT — OpenDocument, используемый, в частности, как стандартный формат файлов для приложений пакета OpenOffice.org — свободного аналога популярного коммерческого пакета Microsoft Office). Если нажать кнопку “Загрузить”, то Wiki-движок сначала собирает (“компилирует”) созданную вами книгу (что отображается на выданной специальной страничке изменением процента готовности), а затем выводит ссылку “Загрузить файл”, щелчок на которой позволяет скачать на свой компьютер созданный файл электронной книги (например, щелчок левой кнопки мыши — просмотр PDF-файла, щелчок правой кнопки и выбор в контекстном меню пункта “Сохранить объект как” — сохранение файла на локальном жестком диске).

И вот результат: наша книга готова! Причем это не просто механический набор статей, а настоящая книга, с титульной страницей (на которой размещены введенные нами заголовок и подзаголовок), содержанием, нумерацией страниц, колонтитулами и ровной версткой текста и картинок.



Как видим, результат — вполне профессиональный. А при желании (и наличии денег) можно заказать обычную, печатную версию созданной книги. Достаточно выбрать справа в панели “Заказать печатную книгу” кнопку “Предпросмотр посредством PediaPress”, и созданный вами файл будет загружен на открывшийся сайт фирмы PediaPress и открыт на нем для такого же постраничного просмотра.



И если результат вам понравится, то можно задать цвет обложки, поместить на ней любое из имеющихся в книге изображений, добавить свою фамилию в качестве “редактора” — и заказать книгу с доставкой на дом.

Такая технология — когда любой желающий может заказать печать своей книги, представленной в электронном варианте, — называется “Print of demand” (“печать на заказ”) и становится все более популярной. И, возможно, благодаря ей могут вскорости уйти в прошлое не только “обычные” типографии, но и книжные магазины и даже, может быть, издательства (в том виде, в котором они существуют сегодня). Но это — уже другая история, требующая отдельного разговора...

## Задачи по программированию в ЕГЭ по информатике. Избранное

Д.М. Златопольский,  
Москва

► В Кодификаторе элементов содержания и требований к уровню подготовки выпускников общеобразовательных учреждений для проведения в 2011 году единого государственного экзамена по информатике и ИКТ [1] приведен перечень возможных алгоритмических задач для подраздела 1.1 перечня требований к уровню подготовки выпускников, достижение которого проверяется на едином государственном экзамене по информатике и ИКТ.

Ниже рассмотрены возможные методы решения некоторых из этих задач. описа-

ние проводится с использованием школьного алгоритмического языка. На диске (он будет вложен в № 10) имеются реализации всех программ на языках Паскаль и Бейсик.

**1. Поиск минимума и максимума двух, трех, четырех данных чисел без использования массивов и циклов<sup>1</sup>**

**1.1. Поиск максимума/минимума среди двух чисел ( $a$  и  $b$ )**

Для двух чисел задачи решаются так, как показано на рис. 1–2.

Обращаем внимание на тот факт, что случаи равенства переменных можно не рассматривать, так как нам безразлично, какое из двух равных значений считать максимумом или минимумом.

Соответствующие фрагменты программ на школьном алгоритмическом языке<sup>2</sup> представлены на с. 9.

Поиск максимума

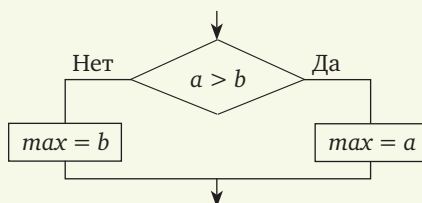


Рис. 1

Поиск минимума

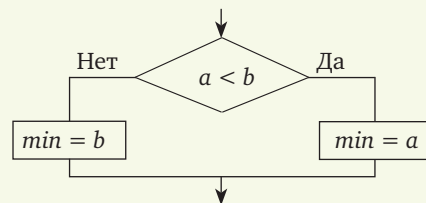


Рис. 2

<sup>1</sup> Формулировки всех задач приведены согласно указанному выше Кодификатору.

<sup>2</sup> Фрагменты программ, связанные с описанием переменных величин, вводом исходных данных, выводом результатов (как правило) и т.п., приводить не будем.



```

если a > b
  то
    макс := a
иначе
  макс := b
все
И
если a < b
  то
    мин := a
иначе
  мин := b
все

```

Можно также использовать неполный условный оператор:

1) при поиске максимума:

```

макс := a
если b > a
  то
    макс := b
все

```

2) при поиске минимума:

```

мин := a
если b < a
  то
    мин := b
все

```

## 1.2. Поиск максимума/минимума среди трех чисел ( $a$ , $b$ и $c$ )

Здесь также можно использовать три неполных условных оператора, каждый из которых соответствует отдельному варианту возможного ответа:

1) при поиске максимума:

```

если a > b и a > c
  то
    макс := a
все
если b > a и b > c
  то
    макс := b
все
если c > a и c > b
  то
    макс := c
все

```

2) при поиске минимума:

```

если a < b и a < c
  то
    мин := a
все
если b < a и b < c
  то
    мин := b
все
если c < a и c < b
  то
    мин := c
все

```

Заметим, что попытка объединить любые два неполных условных оператора в один полный:

1) при поиске максимума:

```

если a > b и a > c
  то
    макс := a
все
если b > a и b > c
  то
    макс := b
иначе
  макс := c
все

```

2) при поиске минимума:

```

если b < a и b < c
  то
    мин := b
все
если c < a и c < b
  то
    мин := c
иначе
  мин := a
все

```

— в ряде случаев дает неправильный результат. Например, результатом выполнения приведенных фрагментов в первом случае при  $a = 5$ ,  $b = 4$ ,  $c = 3$  будет значение величины *макс*, равное 3, а во втором — при  $a = 5$ ,  $b = 4$ ,  $c = 8$  — значение величины *мин*, равное 5.

Правильным вариантом решения является также такой — вместо трех неполных условных операторов применить один оператор, для поиска максимума построенный по следующему принципу:

1) сначала сравниваются два любых числа, например,  $a$  и  $b$ ;

2) если оказалось, что  $a$  больше, чем  $b$ , то именно число  $a$ , как возможный “кандидат” на статус максимума, мы сравниваем с третьим числом  $c$ ; если же, наоборот, число  $b$  больше, чем  $a$ , то именно  $b$  и сравнивается затем с числом  $c$ ;

3) по результатам этого второго сравнения принимается окончательное решение о том, какое число из трех максимальное.

Эти рассуждения можно оформить в виде блок-схемы (см. рис. 3) и фрагмента программы:

```

если a > b
  то
    если a > c
      то
        макс := a
      иначе
        макс := c
    все
  иначе
    если b > c
      то
        макс := b
      иначе
        макс := c
    все
все

```

Заметим, что такую конструкцию называют “вложенный условный оператор”.

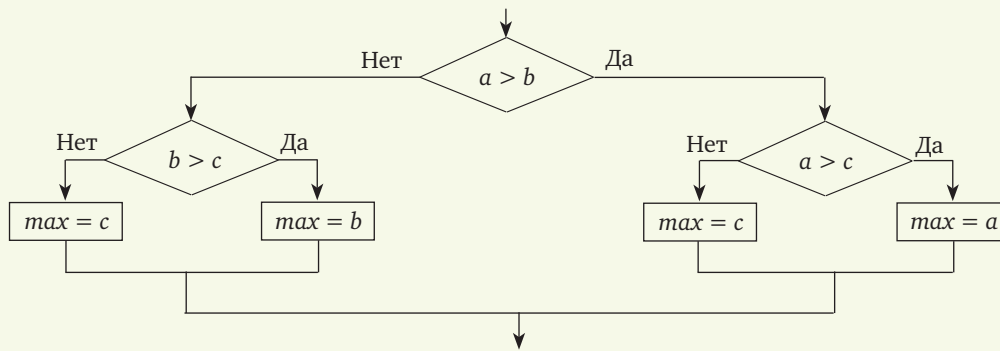


Рис. 3

Аналогично, для поиска минимума также требуется конструкция из нескольких команд **если** (условных операторов), построенная по принципу:

1) сначала сравниваются два любых числа, например,  $a$  и  $b$ ;

2) если оказалось, что  $a$  меньше, чем  $b$ , то именно число  $a$ , как возможный “кандидат” на статус минимума, сравнивается с третьим числом  $c$ ; если же, наоборот, число  $b$  меньше, чем  $a$ , то именно  $b$  и сравнивается затем с числом  $c$ ;

3) по результатам этого второго сравнения принимается окончательное решение о том, какое число из трех является минимальным:

```

если a < b
то
    если a < c
    то
        мин := a
    иначе
        макс := c
все
иначе
    если b < c
    то
        мин := b
    иначе
        мин := c
все
все

```

### 1.3. Поиск максимума/минимума среди четырех чисел ( $a, b, c$ и $d$ )

Ситуация с четырьмя числами — наиболее сложная. Здесь при поиске максимума нужно вести сравнение (тоже при помощи цепочки многократно вложенных команд **если**) в следующем порядке:

- $a$  сравнивается с  $b$
- если  $a > b$ , то  $a$  сравнивается с  $c$ 
  - если  $a > c$ , то  $a$  сравнивается с  $d$ 
    - если  $a > d$ , то макс =  $a$
    - иначе макс =  $d$
  - иначе  $c$  сравнивается с  $d$
- если  $c > d$ , то макс =  $c$
- иначе макс =  $d$

и т.д.

Приведем соответствующие фрагменты программ:

1) поиск максимума:

```

если a > b
то
    если a > c

```

```

то
    если a > d
    то
        макс := a
    иначе
        макс := d
    все
иначе
    если c > d
    то
        макс := c
    иначе
        макс := d
    все
иначе
    если b > c
    то
        если b > d
        то
            макс := b
        иначе
            макс := d
        все
    иначе
        если c > d
        то
            макс := c
        иначе
            макс := d
        все
    все

```

2) поиск минимума:

```

если a < b
то
    если a < c
    то
        если a < d
        то
            мин := a
        иначе
            мин := d
        все
    иначе | c < a
        если c < d
        то
            мин := c
        иначе
            мин := d
        все
    иначе | b < a
        если b < c
        то
            если b < d
            то
                мин := b
            иначе
                мин := d
            все
        иначе
            мин := d
        все

```

```

иначе
  если  $c < d$ 
    то
      мин :=  $c$ 
    иначе
      мин :=  $d$ 
  все
все

```

Полученные фрагменты программ — очень громоздкие и трудночитаемые.

Можно упростить структуру, если использовать логические переменные<sup>3</sup> [1].

Итак, пусть описаны переменные логического типа  $x_1, x_2, x_3, x_4, x_5, x_6$ .

Сначала мы присваиваем каждой такой переменной результат одного сравнения каждой возможной пары чисел:

```

 $x_1 := a > b$ ;  $x_2 := a > c$ ;  $x_3 := a > d$ ;
 $x_4 := b > c$ ;  $x_5 := b > d$ ;
 $x_6 := c > d$ ;

```

Первые три логические переменные хранят результаты сравнения значения исходной переменной  $a$  с каждой из трех остальных. Далее мы должны сравнить со всеми остальными значение переменной  $b$ , но вспомним, что с переменной  $a$  мы ее уже сравнивали и что требуемое сравнение  $b > a$  можно считать эквивалентным записи  $\text{не}(a > b)$ . Поэтому достаточно записать сравнения переменной  $b$  только с переменными  $c$  и  $d$ . Аналогично, переменную  $c$  нужно сравнивать только с переменной  $d$ .

Сформировав таким способом значения логических переменных, на их основе можно сформулировать логические условия, при которых та или иная из четырех имеющихся переменных является максимальной. Например, для переменной  $a$ :

```

если  $x_1$  и  $x_2$  и  $x_3$ 
  то
    макс :=  $a$ 
все

```

А когда максимальным является значение  $b$ ? Это имеет место, когда  $b > a$  (то есть значение переменной  $x_1$  — ложное) и при этом значение переменных  $x_4$  и  $x_5$  — истинное:

```

если  $\text{не } x_1$  и  $x_4$  и  $x_5$ 
  то
    макс :=  $b$ 
все

```

Рассуждая аналогично, можем записать:

```

если  $\text{не } x_2$  и  $\text{не } x_4$  и  $x_6$ 
  то
    макс :=  $c$ 
все
если  $\text{не } x_3$  и  $\text{не } x_5$  и  $\text{не } x_6$ 
  то
    макс :=  $d$ 
все

```

<sup>3</sup> Естественно, в средах программирования, в которых это возможно. В языке Бейсик (вариант QBasic) можно использовать переменные числового типа. В результате выполнения оператора присваивания, например  $x_1 = a > b$ , переменная  $x_1$  будет равна  $-1$  в случае, когда условие в правой части будет истинным, и равна  $0$  — когда ложным.

В заключение заметим следующее. Рассмотренные задачи могут быть решены следующим образом:

### 1.1

1) при поиске максимума:

```

макс :=  $a$ 
если  $b > \text{макс}$ 
  то
    макс :=  $b$ 
все
макс :=  $a$ 
если  $c > \text{макс}$ 
  то
    макс :=  $c$ 
все

```

2) при поиске минимума:

```

мин :=  $a$ 
если  $b < \text{мин}$ 
  то
    макс :=  $b$ 
все
если  $c < \text{мин}$ 
  то
    макс :=  $c$ 
все

```

### 1.2

1) при поиске максимума:

```

макс :=  $a$ 
если  $b > \text{макс}$ 
  то
    макс :=  $b$ 
все
макс :=  $a$ 
если  $c > \text{макс}$ 
  то
    макс :=  $c$ 
все
если  $d > \text{макс}$ 
  то
    макс :=  $d$ 
все

```

2) при поиске минимума:

```

мин :=  $a$ 
если  $b < \text{мин}$ 
  то
    макс :=  $b$ 
все
если  $c < \text{мин}$ 
  то
    макс :=  $c$ 
все
если  $d < \text{мин}$ 
  то
    макс :=  $d$ 
все

```

Правда, здесь возникает вопрос: “Можно ли считать, что при их решении не использованы циклы?”. (Операторы цикла не применялись, но есть повторяющиеся, точнее — аналогичные, действия.)

## 2. Нахождение всех корней заданного квадратного уравнения

Если коэффициент  $a$  квадратного уравнения считать не равным нулю, то задача не должна представлять трудностей для решения.

Квадратное уравнение с коэффициентами  $a, b, c$  может иметь от 0 до 2 корней в зависимости от значения дискриминанта  $d = b^2 - 4ac$ :

— при  $d > 0$  корней — два, и они вычисляются по формулам:

$$x_1 = \frac{-b + \sqrt{d}}{2a}; \quad x_2 = \frac{-b - \sqrt{d}}{2a};$$

— при  $d = 0$  корень — один:

$$x_1 = \frac{-b}{2a};$$

— при  $d < 0$  корней нет.

Так как возможны три варианта ответа, то, как и при решении задачи 1.2, в программе можно использовать три неполных условных оператора:

```

если d > 0
  то
    x1 := (-b + sqrt(d))/(2 * a)
    x2 := (-b - sqrt(d))/(2 * a)
    вывод нс, "Уравнение имеет два корня: "
    вывод "x1=", x1, " x2=", x2
все
если d = 0
  то
    x1 := -b/(2 * a)
    вывод нс, "Уравнение имеет
      один корень: "
    вывод "x1=", x1
все
если d < 0
  то
    вывод нс, "Уравнение
      не имеет корней"
все

```

Здесь, как и при решении задачи 1.2, объединять любые два неполных условных оператора в один полный нельзя.

Можно вместо трех неполных условных операторов применить один вложенный условный оператор:

```

если d > 0
  то
    x1 := (-b + sqrt(d))/(2 * a)
    x2 := (-b - sqrt(d))/(2 * a)
    вывод нс, "Уравнение имеет два корня"
    вывод "x1=", x1, " x2=", x2
  иначе
    если d = 0
      то
        x1 := -b/(2 * a)
        вывод нс, "Уравнение имеет
          один корень"
        вывод "x1=", x1
      иначе
        вывод нс, "Уравнение
          не имеет корней"
    все
  все

```

Обратим внимание на то, что распространенной ошибкой учащихся при разработке программ решения обсуждаемой задачи является отсутствие скобок в знаменателе выражения для расчета корней.

### 3. Нахождение наибольшего общего делителя двух натуральных чисел (алгоритм Евклида)

Древнегреческий математик Евклид в своей знаменитой работе “Начала” (330–320 гг.

до нашей эры) изложил алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел. Суть этого (считающегося самым древним) алгоритма состоит в следующем. Если число  $a$  больше числа  $b$ , то нужно от  $a$  отнять  $b$ , в противном случае, наоборот, от  $b$  отнять  $a$  и повторять эти действия до тех пор, когда  $a$  станет равно  $b$ . После этого искомым НОД будет равен одному из полученных чисел. Сказанное иллюстрирует следующая схема:

$$\text{НОД}(a, b) = \begin{cases} \text{НОД}(a - b, b), & \text{если } a > b \\ \text{НОД}(a, b - a), & \text{если } b < a \\ a \text{ (или } b), & \text{если } a = b \end{cases}$$

Например, при  $a = 26$ ,  $b = 18$  имеем:  
 $\text{НОД}(26, 18) = \text{НОД}(8, 18) = \text{НОД}(8, 10) =$   
 $= \text{НОД}(8, 2) = \text{НОД}(6, 2) = \text{НОД}(4, 2) =$   
 $= \text{НОД}(2, 2) = 2.$

Приведем фрагмент программы, работающий по этому алгоритму:

```

нц пока a <> b
  если a > b
    то
      a := a - b
    иначе
      b := b - a
  все
кц
НОД := a | или НОД := b

```

Как можно улучшить эту программу? Прежде всего можно многократные вычитания заменить на определение остатка от деления одного целого числа на другое. Эти действия должны повторяться, пока ни одно из чисел не равно нулю:

```

нц пока a <> 0 и b <> 0
  если a > b
    то
      a := mod(a, b)
    иначе
      b := mod(b, a)
  все
кц
НОД := a | или НОД := b

```

— где  $mod$  — функция, возвращающая остаток от деления своего первого аргумента на второй. В других языках программирования для этого используется не функция, а специальная операция (как правило, знак этой операции также обозначается  $mod$ ).

Можно также применить оператор цикла с пост-условием:

```

нц
  если a > b
    то
      a := mod(a, b)
    иначе
      b := mod(b, a)
  все
кц_при a = 0 или b = 0

```

После окончания работы оператора цикла (в обоих вариантах) искомое значение НОД может быть определено так:

```

если a = 0
  то
    НОД := b

```

```
иначе
НОД := a
```

```
все
```

или, короче (и эффективно!), так:

```
NOD := a + b
```

Дальнейшее усовершенствование: можно отказаться от многократного сравнения величин  $a$  и  $b$  — достаточно сделать это один раз, а потом учесть, что получаемый в цикле остаток всегда будет меньше, чем второй параметр функции `mod`. В приведенной далее программе, учитывающей это обстоятельство, использованы следующие новые величины:

*макс* — максимальное из двух заданных чисел;

*мин* — то же, минимальное;

*остаток* — остаток от деления *макс* на *мин*.

```
| Определяем значения макс и мин
```

```
если a > b
```

```
то
```

```
макс := a
```

```
мин := b
```

```
иначе
```

```
макс := b
```

```
мин := a
```

```
все
```

```
остаток := mod(макс, мин)
```

```
нц пока остаток <> 0
```

```
макс := мин
```

```
мин := остаток
```

```
остаток := mod(макс, мин)
```

```
кц
```

```
НОД := мин
```

И, наконец, самый короткий вариант решения задачи:

```
нц пока b <> 0
```

```
остаток := mod(a, b)
```

```
a := b
```

```
b := остаток
```

```
кц
```

```
НОД := a
```

В его особенностях разберитесь самостоятельно. В частности, проанализируйте, правильно ли будет работать программа при  $b > a$  и при  $a = b$ .

#### 4. Запись натурального числа в позиционной системе с основанием меньшим или равным 10. Обработка и преобразование такой записи числа

Обозначим заданное натуральное десятичное число —  $n$ , а основание системы счисления, в которую нужно перевести заданное, — *основание*. Идея решения первой части задачи — получить все цифры записи числа в системе с основанием *основание* и записать каждую из них в массив. Конечно, нужно получить и всю новую запись числа. После этого можно решать вторую часть задачи — обрабатывать полученную запись, точнее, массив с ее цифрами. Можно будет найти сумму цифр, максимальную цифру и т.п. (см. [3]).

Напомним методику перевода целых чисел из десятичной системы счисления в систему с другим

основанием. Необходимо определять остаток от деления заданного числа и всех промежуточных целочисленных частных на *основание* и делать это до тех пор, пока частное не станет равно нулю.

В приведенном ниже фрагменте программы, кроме величин  $n$  и *основание*, использованы также следующие основные переменные величины:

— *цифры* — массив с данными целого типа, в котором будут храниться цифры “нового” числа. Размер этого массива следует с учетом возможной длины новой записи числа;

— *кол\_цифр* — фактическое количество цифр в новой записи числа.

Обратим внимание на то, что в массив цифры записываются, начиная с последней цифры новой записи.

```
кол_цифр := 0
```

```
нц пока n > 0
```

```
| Увеличиваем значение кол_цифр
```

```
кол_цифр := кол_цифр + 1
```

```
| Определяем очередную цифру
```

```
цифры[кол_цифр] := mod(n, основание)
```

```
| Определяем целочисленное частное
```

```
n := div(n, основание)
```

```
кц
```

```
| Выводим ответ
```

```
| (цифры в обратном порядке)
```

```
вывод нс, "Запись числа n в новой системе счисления: "
```

```
нц для i от кол_цифр до 1 шаг -1
```

```
вывод цифры[i]
```

```
кц
```

— где *div* — функция, определяющая целочисленное частное от деления своего первого аргумента на второй (в других языках программирования для этого используется не функция, а специальная операция).

Если же считать, что в условии задачи требуется получить строковую (в терминах языка программирования Паскаль) величину *новое*, представляющую собой запись заданного числа в системе с основанием *основание*, то массив можно не использовать. Можно формировать значение величины *новое*, добавляя очередную найденную цифру в начало уже имеющегося значения: полученного значения величины *новое*:

```
новое := ""
```

```
нц пока n > 0
```

```
| Определяем очередную цифру
```

```
цифра := mod(n, основание)
```

```
| Добавляем ее строковое представление
```

```
| к значению новое
```

```
новое := цел_влит(цифра) + новое
```

```
n := div(n, основание)
```

```
кц
```

```
вывод нс, "Запись числа в новой системе счисления: ", новое
```

В последнем варианте применена величина *цифра* (смысл которой понятен), а также стандартная функция школьного алгоритмического языка `цел_влит`, осуществляющая преобразование числового значения в его символьное представление.

Полученное значение *новое* и можно использовать для решения второй части задачи (обработка и преобразование такой записи числа).

## 5. Нахождение сумм, произведений элементов данной конечной числовой последовательности (или массива)

В приведенных ниже фрагментах программ использованы следующие основные величины:

$n$  — общее количество чисел в последовательности (оно может быть заранее задано в программе или вводиться в ходе ее выполнения);

$a$  — очередное обрабатываемое число последовательности.

Смысл величин *сумма* и *произведение* можно легко определить по их именам.

Так как количество обрабатываемых чисел  $n$  известно, в программах использован оператор цикла с параметром.

### 5.1. Суммирование всех чисел последовательности

```
сумма := 0
нц для i от 1 до n
  | Ввод очередного числа a
  ...
  сумма := сумма + a
кц
```

**Примечание.** В программах на языках программирования Бейсик, Паскаль, Си и ряде других оператор `сумма := 0` не является обязательным. Вместе с тем “правилом хорошего тона” является начальное присваивание величине *сумма* нулевого значения (а в отдельных задачах это является обязательным).

Решение рассмотренной задачи применительно к массиву представлено в [3].

### 5.2. Нахождение произведения всех чисел последовательности

Здесь начальное присваивание `произведение := 1` является обязательным:

```
произведение := 1
нц для i от 1 до n
  | Ввод очередного числа a
  ...
  произведение := произведение * a
кц
```

Применительно к массиву (с именем  $m$ ) задача решается аналогично:

```
произведение := 1
нц для i от 1 до n
  произведение := произведение * m[i]
кц
```

В обоих случаях следует иметь в виду, что значения произведения не должны выходить за пределы, допускаемые для использованных типов данных.

## 6. Использование цикла для решения простых переборных задач (поиск наименьшего простого делителя данного натурального числа, проверка числа на простоту и т.д.)

В этой части мы рассмотрим методику решения трех задач.

### 6.1. Определить количество делителей натурального числа $n$

Прежде чем представлять методику, заметим, что в школьном алгоритмическом языке определить, является ли некоторое число  $b$  делителем числа  $a$ , можно с помощью функции *mod* (см. задачу 3):

```
если mod(a, b) = 0
  то
    число b является делителем числа a
  иначе
    число b не является делителем числа a
все
```

Самый простой способ решения задачи — проверить по очереди делимость  $n$  на каждое из чисел 1, 2, 3, ...,  $n$ . Соответствующий фрагмент программы имеет вид:

```
к := 0 | Искомое количество
нц для i от 1 до n
  если mod(n, i) = 0
    то | i - делитель числа n
      к := к + 1
  все
кц
```

Однако легко убедиться, что в интервале  $|n/2 + 1; n - 1|$  делителей числа  $n$  нет, т.е. количество проверок можно сократить почти вдвое:

```
к := 0
нц для i от 1 до div(n, 2)
  если mod(n, i) = 0
    то
      к := к + 1
  все
кц | Учитываем в числе делителей и число n
к := к + 1
```

— где *div* — функция, возвращающая целую часть частного от деления своего первого аргумента на второй.

Можно также учесть тот факт, что у нечетного  $n$  четных делителей быть не может:

```
к := 0
если mod(n, 2) = 0 | n - четное число
  то
    нц для i от 1 до div(n, 2)
      если mod(n, i) = 0
        то
          к := к + 1
    все
  кц
иначе | n - нечетное число
  i := 1
  нц пока i <= div(n, 3)
    если mod(n, i) = 0
      то
        к := к + 1
    все
    i := i + 2 | Рассматриваем только
                | нечетные возможные делители
  кц
все | Учитываем также число n
к := к + 1
```

Количество проверок возможных делителей можно также существенно сократить, если учесть, что у каждого делителя  $d$  числа  $n$ , не превышающего  $\sqrt{n}$ , есть “симметричный” ему делитель, получающийся в результате деления  $n$  на  $d$ . Например, если  $n = 30$ , достаточно найти делители 1, 2, 3, 5 (целая часть квадратного корня из 30 равна 5), а все прочие делители получаем следующим образом:

```
30 div 1 = 30;      30 div 2 = 15;
30 div 3 = 10;      30 div 5 = 6.
```

Правда, из приведенной методики есть одно исключение — если число  $n$  является точным квадратом, то у делителя  $d$ , равного  $\sqrt{n}$ , “симметричного” ему делителя нет.

Учтем это в нашей программе.

```

к := 0
нц для i от 1 до int(sqrt(n))
  если mod(n, i) = 0
    то
      Учитываем i и "симметричный"
      ему делитель
      к := к + 2
    все
  кц
  | Является ли число n точным квадратом
  если int(sqrt(n)) = sqrt(n)
    то
      к := к - 1
  все

```

## 6.2. Определить, является ли заданное натуральное число простым

Наиболее простой способ решения задачи такой:

1. Определить количество делителей заданного числа  $n$  (см. задачу 6.1).
2. Если это количество равно двум, то число  $n$  — простое.

## 6.3. Найти наименьший простой делитель данного натурального числа

Можно рассуждать так. Если заданное число  $n$  — четное, то ответ — 2 (число 1 простым не считается). В противном случае надо рассматривать в качестве искомых числа 3, 5, 7, ...,  $n$  до тех пор, пока не встретится первое простое число, являющееся делителем числа  $n$ . Итак, общая схема алгоритма решения такая:

```

если n — четное
  то
    Ответ равен 2
  иначе
    Рассматриваем в качестве искомых
    числа 3, 5, 7, ..., n
    если среди них встретится простое
    число, являющееся делителем
    числа n
      то
        Прекращаем проверку
    все
    Ответ равен встреченному числу
все

```

В программе на школьном алгоритмическом языке используем следующие величины:

- $n$  — заданное натуральное число;
- *ответ* — искомый делитель;
- *возм\_простое* — возможное простое число;
- $i$  — возможный делитель числа *возм\_простое*;
- $k$  — количество делителей числа *возм\_простое*;
- *найден* — величина логического типа, фиксирующая факт нахождения искомого делителя (в языках Бейсик и Си можно использовать величину целого типа).

Соответствующий фрагмент программы:

```

если mod(n, 2) = 0
  то
    ответ := 2
  иначе
    | Ищем простые числа
    возм_простое := 3; найден := нет
    нц пока возм_простое <= n и не найден
      к := 1

```

```

i := 3
нц пока i <= возм_простое
  если mod(возм_простое, i) = 0
    то
      к := к + 1
    все
  i := i + 2
кц
если k = 2 и
  mod(n, возм_простое) = 0
  то
    найден := да
    ответ := возм_простое
  иначе
    возм_простое :=
      возм_простое + 2
все
кц
все
вывод нс, "Минимальный простой
делитель: ", ответ

```

Видно, что проверка числа *возм\_простое* на “простоту” проводится так, как указано в начале обсуждения задачи 6.1.

Но на самом деле задачу можно решить гораздо проще. А именно: полностью избавиться от проверки делителей на простоту. В самом деле, если число  $n$  не делится на первые простые числа, то оно не разделится и на составные числа, делителями которых являются эти простые. Так, если число не делится на 2 и 3, то оно не делится ни на 4, ни на 6, ни на 9. Поэтому можно просто рассматривать все числа подряд (или после рассмотрения двойки проверять только нечетные, чтобы ускорить работу программы) в поисках первого числа, на которое будет делиться  $n$ . Оно и будет искомым. Если такого числа до  $\sqrt{n}$  не нашлось, то число  $n$  — простое.

Приведем фрагмент программы на алгоритмическом языке:

```

если mod(n, 2) = 0
  то
    ответ := 2
  иначе
    возм := 3;
    нц пока возм <= sqrt(n) и mod(n, возм) <> 0
      возм := возм + 2
    кц
    если mod(n, возм) = 0
      то
        ответ := возм
    иначе
      ответ := n
  все
все
вывод нс, "Минимальный простой
делитель: ", ответ

```

## Литература

1. Кодификатор элементов содержания и требований к уровню подготовки выпускников общеобразовательных учреждений для проведения в 2011 году ЕГЭ по информатике и ИКТ. Подготовлен Федеральным государственным научным учреждением “Федеральный институт педагогических измерений” ([http://www3.ege.edu.ru/images/stories/ege\\_2011/kod\\_2011/inf\\_kodif\\_2011.pdf](http://www3.ege.edu.ru/images/stories/ege_2011/kod_2011/inf_kodif_2011.pdf)).
2. Усенков Д.Ю. Поиск наибольшего и наименьшего значений среди двух, трех и четырех чисел. / “Информатика” № 4/2011.
3. Ушаков Д.М. Задачи на обработку массивов в ЕГЭ по информатике. / “Информатика” № 7/2010.



## КуМир и школьная информатика

К.Ю. Поляков,  
д. т. н.,  
ГОУ СОШ № 163,  
Санкт-Петербург

На фото:  
А.Г. Кушниренко  
на семинаре,  
посвященном  
КуМиру, на  
Дне учителя  
информатики

1 апреля  
2011 г.

▶ Как все уже знают, в ближайшие два года будет происходить постепенный переход на компьютеризированный вариант сдачи ЕГЭ по информатике. Компьютерная тестирующая система (КТС ЕГЭ) пока ориентирована на использование кроссплатформенных систем программирования КуМир (Комплект Учебных МИРов, школьный алгоритмический язык) и FreePascal.

Сильный эффект произвело известие об административном внедрении системы КуМир, которая разработана в НИИСИ РАН и распространяется свободно на условиях лицензии GNU GPL.

Конечно, все “старички”, знакомые с учебниками информатики А.Г. Кушниренко 1990 года, знали о КуМире, кое-где даже использовали его DOS-версию, но широкого распространения (по крайней мере в Санкт-Петербурге) она никогда не имела.

Первый сигнал о втором пришествии КуМира появился в апреле 2010 года после семинаров, которые проводили П.А. Якушкин, В.Р. Лещинер, А.Г. Кушниренко и М.А. Ройтберг на Дне учителя информатики 2.04.2010 в рамках Девятого московского педагогического марафона (<http://marathon.1september.ru/2010-04-02->

[1145461#section-19](http://marathon.1september.ru/2010-04-02-1145461#section-19)). Чуть позже в газете “Информатика” (№ 10/2010) была напечатана статья В.Р. Лещинера и П.А. Якушкина, посвященная компьютерному варианту ЕГЭ.

Новость произвела эффект разорвавшейся бомбы среди учителей информатики. На педагогических сайтах развернулись бурные дискуссии: см. тему “Быть КуМиру?” на сайте [it-n.ru](http://www.it-n.ru) ([http://www.it-n.ru/board.aspx?cat\\_no=6361&tmpl=thread&BoardId=46154&ThreadId=263114&page=0](http://www.it-n.ru/board.aspx?cat_no=6361&tmpl=thread&BoardId=46154&ThreadId=263114&page=0)) и аналогичное обсуждение на сайте [pedsovet.org](http://pedsovet.org) (<http://pedsovet.org/forum/index.php?showtopic=6791>). Большинство участников этих словесных баталий КуМир не приняли, было сказано много ругательных слов.

Автору этих строк КуМир тоже никогда не нравился, хотя нравились идеи С.Пейперта, Г.А. Звенигородского и учебник информатики А.Г. Кушниренко 1990 года. Такой вот парадокс. В результате была написана среда “Исполнители” (<http://kpolyakov.narod.ru/school/robots/robots.htm>), которая успешно используется с 1992 года по сей день во многих школах. Кстати, в начале 90-х журнал “Информатика и образование” отклонил статью, посвященную “Исполнителям”, только из-за того, что “среда КуМир уже получила широкое распространение и альтернативы не актуальны”.

Но вернемся к КуМиру. По словам самих авторов, КуМир предназначен для начального обучения алгоритмизации и программированию (6–7-е классы). Пока он занимал эту нишу, все было относительно хорошо и спокойно: практически никто из учителей не трогал КуМир, но и КуМир никого не трогал.



Но теперь встал вопрос о том, что КуМир становится одним из (двух) языков, разрешенных на компьютерном ЕГЭ по информатике, то есть “поднимается” на уровень 10–11-х классов. Это заставило задуматься. Мера явно вынужденная; для компьютерного ЕГЭ требуются:

- кроссплатформенность;
- бесплатность;
- простота установки;
- относительная известность и популярность.

Таких сред не очень много, поэтому “архитекторов” нововведений можно понять.

В начале этого учебного года стало ясно, что возвращение КуМира — это серьезно, и его проталкивают “сверху”. Московский институт открытого образования (МИОО) проводит дистанционный курс “Подготовка выпускников к ЕГЭ по информатике и ИКТ в компьютеризированной форме” (<http://www.mos-cons.ru/course/view.php?id=24>), где КуМиру отводится важнейшая роль.

29 ноября 2010 года тот же МИОО проводит апробацию компьютеризированной системы проведения ЕГЭ по информатике и ИКТ (КТС ЕГЭ) ([http://statgrad.mioo.ru/sg10\\_11/ikt/index.htm](http://statgrad.mioo.ru/sg10_11/ikt/index.htm)).

В газете “Информатика” (№ 24/2010 и № 2/2011) публикуются статьи А.Г. Леонова из серии “Освой КуМир за 6 часов”. До этого, в 2009 году, в “Информатике” был напечатан цикл материалов А.Г. Кушниренко и А.Г. Леонова “Методика преподавания основ алгоритмизации на базе системы КуМир” (<http://inf.1september.ru/article.php?ID=200902401>).

На очередном Дне учителя информатики 02.04.2011 (<http://marathon.1september.ru/2011-04-01-1145461>) продвижение КуМира продолжается (проведены тематический семинар и “круглый стол”).

Все это говорит о том, что с КуМиром придется считаться. Как и любое явление, он имеет достоинства и недостатки. О них много говорили на упомянутом курсе МИОО, даже была сделана Wiki-страничка “Плюсы и минусы КуМира” (<http://www.mos-cons.ru/mod/wiki/view.php?id=1718>). Там представлен широкий спектр мнений, поэтому здесь не будем повторяться.

Автор в текущем учебном году решил попробовать перейти в 7–8-х классах на КуМир. Сразу обнаружились достаточно серьезные проблемы, которые позволяли сделать вывод о непригодности среды в ее текущем состоянии для этих целей. Самые важные среди них:

- чрезвычайная медлительность КуМира, работающего в сотни раз медленнее, чем любая из Паскаль-сред;
- отсутствие “нормальной” (не черепашей) графики;
- совершенно неразвитые средства работы со строками (например, не было функции поиска);
- устаревшая и неудобная работа с файлами (например, перед тем как открыть файл на запись, нужно каждый раз проверять, существует ли он, и если нет, то создавать пустой файл отдельной командой).

К счастью, вскоре удалось выйти на прямую связь с М.А. Ройтбергом — руководителем группы разработчиков КуМира. В результате сотрудничества большинство проблем удалось снять, и появилась версия 1.8.0 (<http://lpm.org.ru/kumir2/>), в которой можно нормально работать, то есть делать то, что раньше мы делали на Паскале и Си. Медлительность, правда, осталась, но А.Г. Кушниренко обещал, что версия 2.0 будет существенно быстрее.

Материалы презентаций к урокам по КуМиру (около 400 слайдов) размещены на диске в приложении к этому номеру, их можно скачать на сайте автора (<http://kpolyakov.narod.ru/school/ppt.htm#algi>).

Необходимо отметить, что ряд проблем все же остаются в текущей версии (1.8.0):

- нельзя менять значения аргументов внутри вспомогательных алгоритмов (например, в реализации алгоритма Евклида как функции приходится заводить две лишние переменные);
- нельзя вызывать функцию как процедуру, игнорируя ее результат (например, когда результат функции — код возврата, и в данном случае он программиста не интересует);

- неудобная и неполная справочная система;

- нет форматного вывода на консоль и в файл, как в Паскале (типа “вывод x:4”), это нужно, например, чтобы вывести на экран матрицу ровными столбиками.

По словам разработчиков, по крайней мере некоторые из этих недостатков будут устранены в версии 2.0.

Автор признателен М.А. Ройтбергу за помощь в решении проблем, связанных с использованием КуМира, и плодотворное сотрудничество.

### Список сетевых ресурсов по КуМиру:

1. Сайт разработчиков КуМира (<http://lpm.org.ru/kumir2/>).
2. КуМир на сайте НИИСИ РАН (<http://www.niisi.ru/kumir/>).
3. Кушниренко А.Г., Леонов А.Г. Методика преподавания основ алгоритмизации на базе системы КуМир (<http://edu.1september.ru/courses/07/010/>).
4. Учебники А.Г. Кушниренко ([http://publ.lib.ru/ARCHIVES/K/KUSHNIRENKO\\_Anatoliy\\_Georgievich/\\_Kushnirenko\\_A.\\_G..html](http://publ.lib.ru/ARCHIVES/K/KUSHNIRENKO_Anatoliy_Georgievich/_Kushnirenko_A._G..html)).
5. Прищепина Т.А. Преподавание программирования в среде КуМир ([http://ido.tsu.ru/other\\_res/school2/osn/metod/prog/index.html](http://ido.tsu.ru/other_res/school2/osn/metod/prog/index.html)).
6. Крепышев О.Ю. Уроки в среде КуМир, включая “Робот” ([http://itdo.my1.ru/index/materialy\\_dlja\\_skachivanija/0-6](http://itdo.my1.ru/index/materialy_dlja_skachivanija/0-6)).
7. Решение задач в системе КуМир (<http://test.kumir.su/>).
8. Кириенко Д.П. Курс алгоритмизации с использованием исполнителей системы КуМир и автоматического тестирования (<http://server.179.ru/wiki/wakka.php?wakka=DenisKirienko/Kumir>).
9. Зайдельман Я.Н. Курс “Алгоритмизация и программирование: от первых шагов до подготовки к ЕГЭ” (<http://edu.1september.ru/courses/107/005/>).

### Общие выводы по результатам года:

▼ школьники воспринимают русские команды намного легче, чем английские;

▼ для изучения основ программирования и алгоритмизации КуМир ничем не хуже Паскаля и Бейсика, в новой версии можно делать практически все, что нужно;

▼ КуМир очень неплохо идет даже в 9–11-х классах на базовом уровне и может быть очень удачным выбором для тех, кто в будущем не будет профессионально программировать.

## Задачи на обработку строк в ЕГЭ по информатике

Д.М. Златопольский,  
Москва

► В Кодификаторе элементов содержания и требований к уровню подготовки выпускников общеобразовательных учреждений для проведения в 2011 году единого государственного экзамена по информатике и ИКТ [1] приведен перечень возможных алгоритмических задач для подраздела 1.1 перечня требований к уровню подготовки выпускников, достижение которого проверяется на едином государственном экзамене по информатике и ИКТ. В их числе представлены задачи на обработку строк<sup>1</sup>:

1. Обработка отдельных символов данной строки. Подсчет частоты появления символа в строке.

2. Работа с подстроками данной строки с разбиением на слова по пробельным символам. Поиск подстроки внутри данной строки, замена найденной подстроки на другую строку.

Ниже рассмотрены возможные методы решения ряда задач указанного типа. Описание проводится с использованием школьного алгоритмического языка. На диске (он будет вложен в № 10) имеются реализации всех программ на языках Паскаль и Бейсик.

### 1. Определить, сколько раз в заданной строке встречается некоторый символ

В школьном алгоритмическом языке величина строкового (в терминах языка Паскаль)

<sup>1</sup> Формулировки задач приведены согласно указанному выше Кодификатору.

типа рассматривается как массив, элементами которого являются отдельные символы. Это позволяет обратиться к тому или иному символу строки как к элементу массива — по имени строковой величины и его (символа) номеру<sup>2</sup>.

Рассмотреть же все символы строки можно, подсчитав их количество, что, в свою очередь, можно сделать с помощью функции `длин`:

```
количество := 0
нц для номер от 1 до длин(строка)
если строка[номер] = симв
то
    количество := количество + 1
все
кц
```

— где *строка* — величина строкового типа — заданная строка;

*симв* — символ, число вхождений которого в строку ищется.

**Примечание.** Фрагменты программ, связанные с описанием переменных величин, вводом исходных данных, выводом результатов (как правило) и т.п., приводить не будем.

### 2. Определить позицию (номер) первого вхождения некоторого символа в заданную строку (если символа в строке нет, то вывести 0)

Попытка по аналогии с предыдущей задачей решить рассматриваемую задачу так:

```
искомый_номер := 0
нц для номер от 1 до длин(строка)
если строка[номер] = симв
то
    искомый_номер := номер
все
кц
если искомый_номер = 0
то
    вывод нс, "В строке нет заданного символа"
```

<sup>2</sup> Так же можно обратиться к отдельному символу строковой величины и в программе на языке Паскаль. В языке Бейсик для этого используется функция `MID$`.

```

иначе
  вывод нс, "Номер первого
    вхождения заданного символа — "
  вывод искомый_номер
все

```

— даст неправильный результат — при наличии в строке нескольких вхождений заданного символа ответом будет номер последнего (а не первого!) вхождения.

Можно решить задачу, используя величину логического типа *впервые*, определяющую, впервые ли в строке встретился заданный символ:

```

искомый_номер := 0
впервые := да
нц для номер от 1 до длин(строка)
  если строка[номер] = симв и впервые
    то
      искомый_номер := номер
      | Следующие вхождения будут уже
      | не первыми
      впервые := нет
    все
  кц
  ...

```

Можно также прекратить проверки, как только встретится заданный символ:

```

номер := 1
найден := нет
нц пока номер <= длин(строка) и не найден
  если строка[номер] = симв
    то
      | Встретился искомый символ
      найден := да
      искомый_номер := номер
    иначе | Переходим к следующему символу
      i := i + 1
    все
  кц
если найден
  то
    вывод нс, "Номер первого вхождения
      заданного символа — "
    вывод искомый_номер
  иначе
    вывод нс, "В строке нет заданного
      символа"
  все

```

— где *найден* — величина логического типа, фиксирующая факт нахождения первого вхождения заданного символа в строку.

В программе на языке Бейсик аналогом величин *впервые* и *найден* могут являться переменные числового типа.

И, конечно, можно решить задачу, исследовав заданную строку с ее конца.

### 3. Определить, есть ли в заданной строке некоторый символ

Прежде всего заметим, что такой вариант решения:

```

нц для номер от 1 до длин(строка)
  если строка[номер] = симв
    то
      вывод "Заданный символ в строке есть"

```

```

иначе
  вывод "В строке нет заданного символа"
все
кц

```

— ошибочный — на экран ответ будет выводиться многократно.

Неправильный результат дает также такой способ, использующий величину логического типа *есть*, которая фиксирует факт наличия заданного символа в заданной строке:

```

нц для номер от 1 до длин(строка)
  если строка[номер] = симв
    то
      есть := да
    иначе
      есть := нет
    все
  кц
если есть
  то
    вывод "Заданный символ в строке есть"
  иначе
    вывод "В строке нет заданного символа"
все

```

Его особенности проанализируйте самостоятельно.

Наиболее простой правильный способ решения задачи такой:

- 1) подсчитать число вхождений заданного символа в заданную строку (см. задачу 1);
- 2) по найденному значению получить ответ.

Можно также не рассматривать весь массив, а прекратить проверки, как только встретится заданный символ (см. решение задачи 2 с использованием величины *найден*).

```

номер := 1
найден := нет
нц пока номер <= длин(строка) и не найден
  если строка[номер] = симв
    то
      найден := да
    иначе | Переходим к следующему символу
      i := i + 1
    все
  кц
если найден
  то
    вывод "Заданный символ в строке есть"
  иначе
    вывод "В строке нет заданного символа"
все

```

### 4. Определить, сколько раз в заданной строке встречается некоторая подстрока

Решение задачи во многом аналогично решению задачи 1. Единственное отличие в том, что проверять надо не по одному символу, а по несколько. При анализе будем называть несколько последовательных символов строки — “вырезка”. При этом возникает вопрос — чему равен номер последнего символа строки, начиная с которого следует формировать вырезку символов?

Обозначим количество символов в заданной строке — *длина1*, в подстроке — *длина2*. Рассмотрим несколько возможных значений:

длина1	длина2	Номер последнего символа строки для формирования вырезки
10	2	9
10	3	8
15	5	11
15	6	10

Из таблицы можно установить, что номер последнего символа строки для формирования вырезки определяется по формуле:  $длина1 - длина2 + 1$ .

Кроме того, теперь надо сравнивать с заданной подстрокой сформированные вырезки. В школьном алгоритмическом языке вырезку из строки можно получить, указав имя переменной, из которой формируется вырезка, и в квадратных скобках — начальный и конечный номера символов вырезки, например, строка[1:5], строка[10:12], строка[k1:k2] и т.п. В языке Бейсик для этого используется функция MID\$, в языке Паскаль — функция Copy. В этих функциях — три параметра: величина строкового типа, начальный номер символа для вырезки и длина вырезки.

Обратим внимание на то, что, хотя мы используем термин “вырезка”, из самой строки ничего не вырезается — ее значение не изменяется.

Итак, фрагмент программы решения задачи:

```
длина1 := длин(строка)
длина2 := длин(подстрока)
количество := 0
нц для номер от 1 до длина1 - длина2 + 1
  если строка[номер:(номер + длина2 - 1)] =
    подстрока
  то
    количество := количество + 1
все
кц
```

— где *подстрока* — подстрока, число вхождений которой в строку ищется.

### 5. Определить позицию (номер) первого вхождения некоторой подстроки в заданную строку (если подстроки в строке нет, то вывести 0)

Сразу же заметим, что в языке Паскаль имеется стандартная функция POS, используя которую можно решить задачу. Тем не менее считаем полезным решить задачу и на Паскале, условно допустив, что такой функции нет.

Решение данной задачи также во многом аналогично решению задачи 2. Неправильные варианты мы рассматривать не будем, а приведем два правильных варианта.

```
1)
длина1 := длин(строка)
длина2 := длин(подстрока)
искомый_номер := 0
впервые := да
нц для номер от 1 до длина1 - длина2 + 1
  если строка[номер:(номер + длина2 - 1)] =
    подстрока и впервые
  то
    искомый_номер := номер
    | Следующие вхождения будут уже
    | не первыми
    впервые := нет
```

```
все
кц
вывод нс, искомый_номер
```

2)

```
длина1 := длин(строка)
длина2 := длин(подстрока)
искомый_номер := 0
номер := 1
найден := нет
нц пока номер <= длина1 - длина2 + 1 и
  не найден
  если строка[номер:(номер + длина2 - 1)] =
    подстрока
  то
    искомый_номер := номер
    найден := да
  иначе | Переходим к следующему символу
    номер := номер + 1
все
кц
вывод нс, искомый_номер
```

Конечно, и здесь можно решить задачу, исследовав заданную строку с ее конца, точнее — с  $(длина1 - длина2 + 1)$ -го символа.

### 6. Определить, есть ли в заданной строке некоторая подстрока

Наиболее простой правильный способ решения задачи такой:

- 1) подсчитать число вхождений подстроки в заданную строку (см. задачу 4);
- 2) по найденному значению получить ответ.

Можно также не рассматривать весь массив, а прекратить проверки, как только встретится заданный символ (см. решение задачи 3 с использованием величины *найден*).

### 7. Удалить из заданной строки все вхождения некоторой подстроки

В программе на школьном алгоритмическом языке удалить из строки подстроку можно, присвоив вырезке с подстрокой (см. выше) “пустое” значение, например, для подстроки из четырех символов, начиная с 7-го:

```
строка[7:10] := ""
```

Еще две важные особенности программы решения задачи.

1. Поскольку в ходе обработки строки ее длина меняется, использовать оператор цикла с параметром, в конечном значении которого применяется величина *длина1* (количество символов в заданной строке), нельзя. Следует применить оператор цикла с условием следующего вида:

```
номер <= длин(строка) - длина2 + 1
```

— где *длина2* — количество символов в удаляемой подстроке.

2. При рассмотрении начальных номеров подстрок после нахождения искомой подстроки и ее удаления величина *номер* не меняется (убедитесь в этом!), а в случае, когда искомая подстрока не встретилась, — увеличивается на 1.

С учетом сказанного фрагмент программы, решающей задачу, имеет вид:

```
номер := 1
нц пока номер <= длин(строка) - длина2 + 1
  если строка[номер:(номер + длина2 - 1)] =
    подстрока
  то |Встретилась искомая подстрока
    |Удаляем ее
    строка[номер:(номер +
      длина2 - 1)] := ""
    |Величина номер - не меняется
  иначе
    |Переходим к следующему символу
    |для проверки
    номер := номер + 1
все
кц
```

### Язык Паскаль

```
...
номер := 1;
while номер <= Length(строка) -
  длina2 + 1 do
  if Copy(строка, номер, длina2) =
    подстрока then
    {Встретилась искомая подстрока.
    Удаляем ее}
    Delete(строка, номер, длina2) = ''
    {Величина номер - не меняется}
  else
    {Переходим к следующему символу
    для проверки}
    номер := номер + 1;
...
```

Решение задачи на языке Бейсик (вариант QBasic) требует отдельного обсуждения. В этом языке нет процедуры для удаления подстроки, а процедура MID\$, предназначенная для замены одной подстроки на другую, не допускает использования в качестве последней пустой строки.

Задачу можно решить, используя вспомогательную переменную строкового типа *stroka2* и записав в нее последовательно все символы исходной строки, кроме тех, которые входят в удаляемую подстроку:

```
длина1 = LEN(строка)
длина2 = LEN(подстрока)
stroka2 = ""
номер = 1
WHILE номер <= длина1 - длина2 + 1
IF MID$(строка, номер, длina2) <>
  подстрока THEN
  'Добавляем очередной символ исходной
  'строки
  stroka2 = stroka2 +
    MID$(строка, номер, 1)
  'Переходим к следующему символу
  номер = номер + 1
ELSE 'Встретилась удаляемая подстрока
  'Пропускаем ее
  номер = номер + длina2
END IF
WEND
'Так как в операторе цикла
'были рассмотрены не все символы
'исходной строки,
'обрабатываем оставшиеся символы
FOR i = номер TO длina1
```

```
stroka2 = stroka2 + MID$(строка, i, 1)
NEXT i
```

После этого можно полученное значение *stroka2* присвоить величине *stroka*:

```
stroka = stroka2
```

Конечно, описанный способ решения обсуждаемой задачи может быть применен и в программах на других языках программирования.

### 8. Заменить в заданной строке все вхождения некоторой подстроки на другую подстроку

Используем в программе величины:

— *строка* — заданная строка;

— *подстрока1* — заменяемая подстрока;

— *подстрока2* — новая подстрока;

— *длина1* — количество символов в величине *подстрока1*;

— *длина2* — то же, в величине *подстрока2*.

Приводя фрагмент программы, обращаем внимание на условие в операторе цикла (см. предыдущую задачу), а также на особенности изменения величины *номер*:

```
длина1 := длин(подстрока1)
длина2 := длин(подстрока2)
номер := 1
нц пока номер <= длин(строка) -
  длina1 + 1
  если строка[номер:(номер + длina1 - 1)] =
    подстрока1
  то |Встретилась искомая подстрока
    |Заменяем ее
    строка[номер:(номер + длina1 - 1)] :=
      подстрока2
    |Изменяем значение величины номер
    номер := номер + длina2
  иначе
    |Переходим к следующему символу
    |для проверки
    номер := номер + 1
все
кц
```

В языке Паскаль для замены одной подстроки на другую следует использовать процедуры Delete и Insert.

В языке Бейсик для замены предназначена процедура MID\$, однако (!) она работает только когда число символов в обеих подстроках одинаковое.

Если длины исходной и заменяемой строки разные, то в программе на языке Бейсик можно использовать метод, аналогичный описанному в задаче 7.

### 9. Дана фраза, слова которой отделены друг от друга одним пробелом (начальных и конечных пробелов нет). Получить массив слов этой строки

Решение задачи будет представлено на диске к следующему номеру “Информатики” (№ 10/2011).

### Литература

1. Кодификатор элементов содержания и требований к уровню подготовки выпускников общеобразовательных учреждений для проведения в 2011 году единого государственного экзамена по информатике и ИКТ. Подготовлен Федеральным государственным научным учреждением “Федеральный институт педагогических измерений” (ФИПИ) ([http://www3.ege.edu.ru/images/stories/ege\\_2011/kod\\_2011/inf\\_kodif\\_2011.pdf](http://www3.ege.edu.ru/images/stories/ege_2011/kod_2011/inf_kodif_2011.pdf)).



## ЭТО ПОЛЕЗНО ЗНАТЬ

### У облака в плену

► Когда смотришь на белоснежные облака, медленно плывущие по голубому небу, вряд ли задумаешься о компьютерах. Между тем разработчики новых сетевых технологий об облаках не забывают ни на минуту. Программисты, конечно, не изучают природу воздушных скоплений, они просто совершенствуют “облачные технологии”, которые еще называют “облачными вычислениями”.

Облачные технологии, еще вчера считавшиеся вымыслом, сегодня стали реальностью. Термин “облако” здесь используется как метафора. “Облачная обработка данных” — это парадигма, в рамках которой информация постоянно хранится на серверах в Интернете и временно используется на клиентской стороне, например, на персональных компьютерах, игровых приставках, ноутбуках, смартфонах и т.д. Пройдет немного времени и, возможно, будет пересмотрена концепция компьютера как технического устройства. Настанет час, когда на жестком диске компьютера вы не найдете прикладных программ, — все необходимые функции будут доступны через Интернет.

А пока посмотрим, какие полезные сервисы и приложения, считающиеся частью облачных вычислений, предлагает сегодня Всемирная паутина. Для этого в одной из поисковых систем наберем “Онлайн-сервисы” и почти мгновенно увидим десятки ссылок на сайты, с помощью которых можно воплотить в жизнь любую творческую идею.

С чего начать? Может быть, с орфографии, ведь каждому из нас приходится часто проверять написанное: слово, текст доклада, электронное сообщение. Мы роемся в словарях и книгах, вспоминаем правила грамматики, а рядом дожидается своего часа замечательный сервис. Он находится в Интернете по адресу <http://www.orfo.ru/online/>. “Орфо” — это система проверки правописания.

Заходим на сайт. Сразу же в глаза бросается филин — символ мудрости и благоразумия (рис. 1). Под пернатым — окно. Нетрудно догадаться, что в него нужно вставить текст, предварительно скопи-

рованный из какого-либо документа. Загрузим его, а затем нажмем кнопку **Проверить**, лежащую под клювом красивой птицы. В случае обнаружения ошибки система тут же подчеркнет слово красной волнистой чертой, будто строгий учитель, проверяющий школьную тетрадку. От ошибок не застрахован никто, даже самый начитанный и грамотный пользователь. Повторение слова, пропущенная запятая, лишний пробел — все заметит умный сервис! И не только в русских, но и в иностранных текстах. Попробуйте: очень полезное приложение.

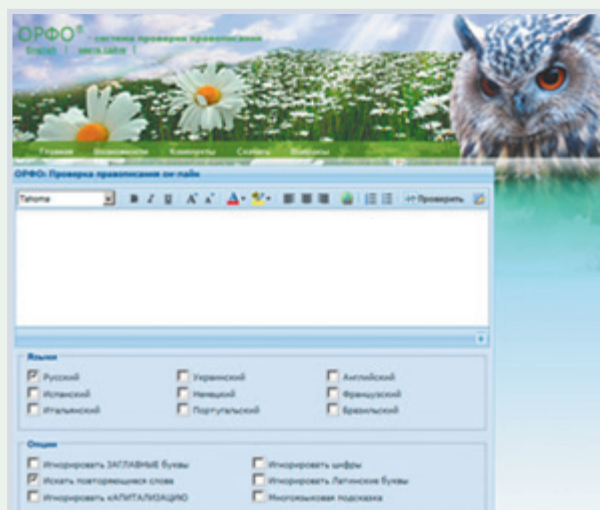


Рис. 1

А что делать, если требуется создать красивый ролик из фотографий, а под рукой нет подходящей программы? Конечно же воспользоваться онлайн-сервисом, например, “Фотофильмом”, расположенным по адресу: <http://fotofilmi.ru/>. Зайдем на сайт, зарегистрируемся. Заранее соберем фотографии в одну папку. В нее же поместим музыкальный файл, который будет служить фоном будущего фотофильма. Нажимаем на кнопку **Создать!**, расположенную на самом видном месте сайта. В открывшееся окно внесем название фильма, коротко опишем его содержание, выберем нужную категорию, чтобы потом было легче найти ролик. Активируем несколько радио-кнопок — разрешений пользователям Интернета просматривать будущее произведение.

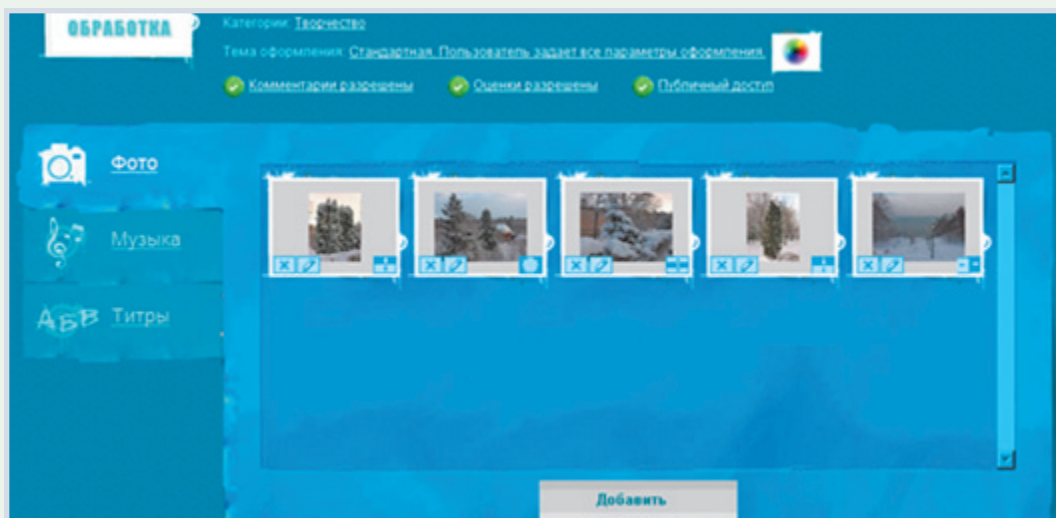


Рис. 2

Активировав вспомогательные функции, перейдем к созданию фильма (рис. 2). Наполняем визуальным содержанием разработанный сценарий: добавляем фотографии, музыку и титры. Все команды — под рукой. В главном окне помещаются изображения. Под каждой фотографией — три значка: удаление, редактирование и демонстрация. Нажмем на карандашик, откроется окно, в котором выберем эффекты просмотра и продолжительность просмотра кадра на экране (рис. 3).



Рис. 3

Настроив изображение, перейдем к добавлению звука. Кликнем последовательно **Музыка**, **Добавить**. Откроем в папке с рабочими материалами музыкальный трек и поместим его в будущий клип. Сохраним изменения и займемся созданием титров. В новом окне, которое откроется сразу после нажатия кнопки **Титры**, зададим подходящий формат текста (рис. 4). Опять впишем в соответствующую ячейку заголовок фильма. Еще раз внимательно все проверим, а затем сгенерируем фильм.

Подождем немного, пока идет его обработка. По окончании процесса появится экран. Просмотрим фотофильм, наслаждаясь результатами проделанной работы. Теперь стоит показать его миру. Как это лучше сделать? Вариантов несколь-

ко. С помощью сервиса ролик можно разместить на своей страничке в одной из социальных сетей. Кроме того, приложение “Фотофильм”, завершая процесс создания, открывает адрес страницы ролика и его HTML-код. Их можно разместить на своем сайте. Фильм можно скачать и на компьютер, нажав на соответствующую кнопку. Все это делается в несколько кликов мышки. Как видим, с помощью данного сервиса сделать фотозарисовку проще простого.

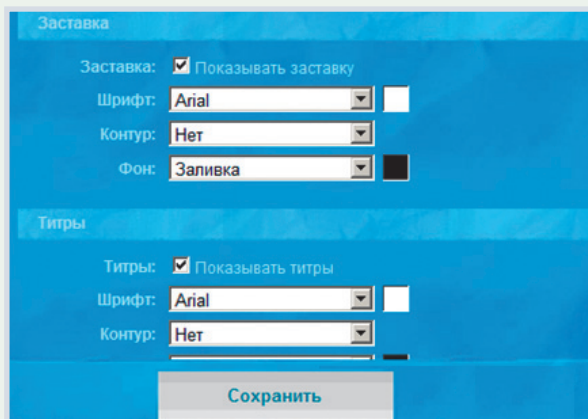


Рис. 4

Расскажем еще об одном сервисе. Для тех, кто много общается в Интернете, просто необходим оригинальный аватар — красивая картинка, вроде узнаваемой визитной карточки. Чтобы ее сделать, не обязательно владеть графическим редактором, достаточно воспользоваться специальным “облачным” сервисом. Его адрес в Интернете: <http://avatar.pho.to/ru>. Зайдя на сайт, можно растеряться от обилия фотографий известной американской поп-дивы Бритни Спирс (рис. 5). Все здесь двигается, подмигивает, переливается. Хотите выглядеть не хуже американской красавицы? Выбирайте понравившийся аватар и вставляйте в него свою фотографию!

После нескольких простейших операций откроется окно, в котором появится анимированный аватар — ваша визитка на сайте, в социальной сети или в популярном чате. Если что-то в картинке не устраивает, доработайте: задайте

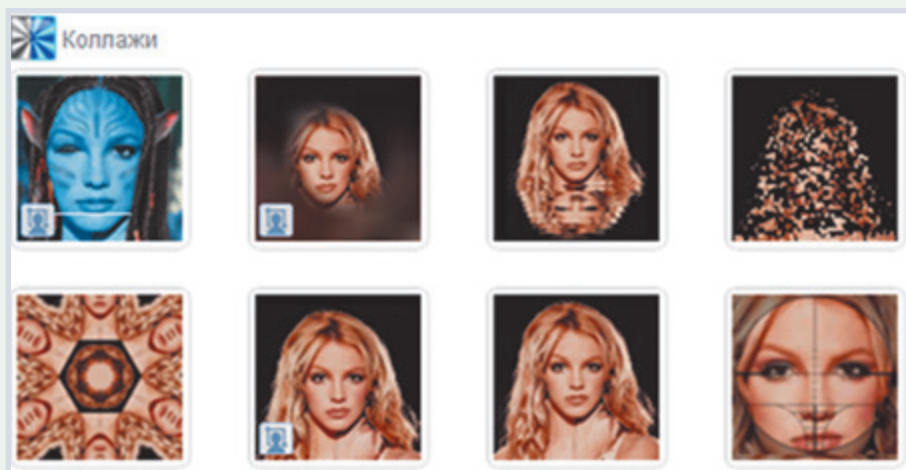


Рис. 5

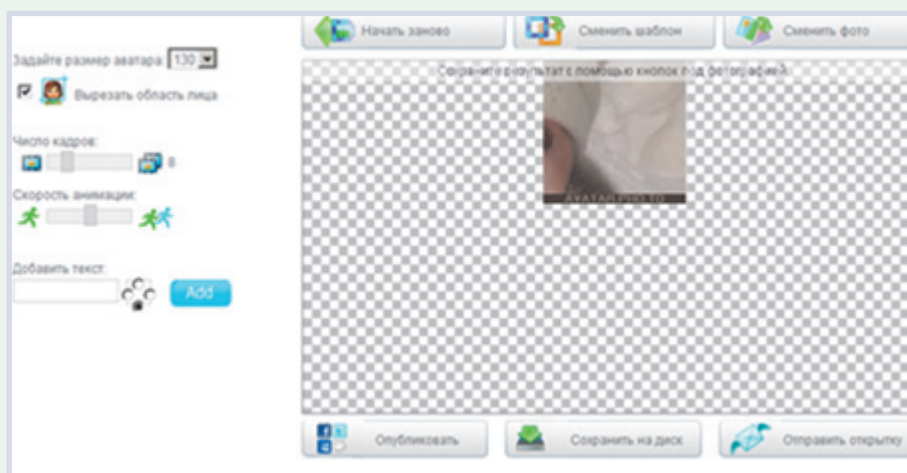


Рис. 6

нужный размер, выберите скорость анимации, добавьте текст (рис. 6). Фантазируйте, изобретайте, пробуйте! Ресурс позволяет.

Что еще можно найти в “облаках”? Да что угодно! Каких сервисов в них только нет! Хотите сделать музыкальный рингтон, чтобы удивить своих друзей? Нет проблем! В этом поможет сервис <http://www.madringtones.org/>. Нужно оформить фотографию, применяя различные спецэффекты? Нет ничего проще! Воспользуйтесь приложением <http://www.befunky.com/>. В Интернете есть все!

Среди облачных технологий найдутся даже виртуальные операционные системы! С какого бы компьютера вы не входили в Интернет, ваша операционная система не подведет: ведь она хранится в надежном и мощном сервере!

Одна из лучших виртуальных операционных систем — *iCloud* (<http://icloud.com/ru>). В ней отыщется все для рядового пользователя: почтовый клиент, графический редактор, офисный пакет, проигрыватель, программа для обмена мгновенными сообщениями и многое другое. Программное обеспечение *iCloud* — это интернет-сервис, который предоставляет пользователю удаленное рабочее место (рис. 7). Заходя в систему, он будет чувствовать себя так же комфортно, как за домашним компьютером, в какой бы точке планеты ни находился.

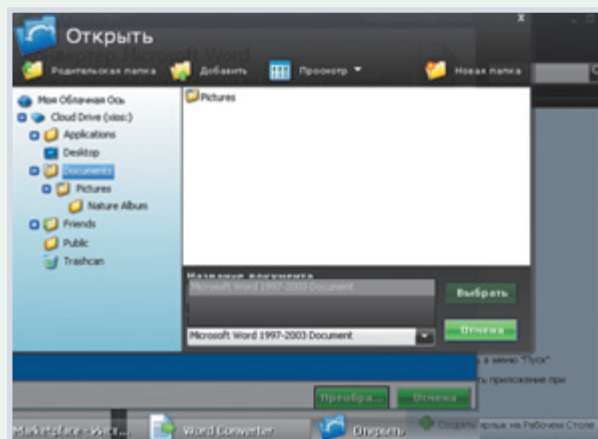


Рис. 7

Появившись несколько лет назад, онлайн-приложения развиваются столь стремительно, что многие из них по своей функциональности уже не отстают от настольных программ. Облаков-сервисов становится все больше. В виртуальном “небе” можно найти “облако” на любой вкус, для решения любой творческой задачи. Благодаря “облачным технологиям” скорость получения и обработки информации увеличивается в разы. И как тут не вспомнить всем известную фразу: “Кто владеет информацией, тот владеет миром!”



## Литература

1. Сетевой подрядчик. / ComputerBild, № 22/2010.

## Источники сети Интернет

1. <http://www.internethalyava.ru/node/1345>
2. <http://webmilk.ru/2009/11/23/onlajn-servisy-dlya-sozdaniya-vizitokportfolio/>
3. [http://clubs.ya.ru/4611686018427405545/replies.xml?item\\_no=6510](http://clubs.ya.ru/4611686018427405545/replies.xml?item_no=6510)
4. <http://www.scrapblog.com/>
5. <http://www.abbeyonline.com/ru/home/index>

► Решения ребусов по информатике, опубликованных в газете “В мир информатики” № 153 (“Информатика” № 23/2010), прислали также Бойчун Антон, Вавилова Кристина, Васильев Дмитрий, Григорьева Владислава, Долгополов Иван, Зайцева Анна, Земнович Любовь, Калинин Алексей, Калининев Александр, Кирьянова Анастасия, Климов Кирилл, Коновалова Юлия, Кривошеева Ольга, Макаров Владислав, Никитина Виктория, Пла-

тонов Дмитрий, Раутанен Яна, Роговская Мария, Савиных Екатерина, Сафонов Евгений, Симкина Мария, Тарасенко Павел, Таргонский Вячеслав, Тепаев Артем, Фавстов Никита, Фурс Валерия, Чернова Ксения, Шубина Галина и Яблонская Анастасия, Республика Карелия, г. Сегежа, школа № 5, учитель **Меньшиков В.В.**

Юхтенко Илья, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**, представил решения задач “Спорт и учеба” и “Кусок мыла”.

Числовой ребус “Сестры” правильно решили также:

— Васильева Юлия, Евченко Мария, Кольтякова Анна, Семенова Наталья и Харитоновна Елена, Республика Башкортостан, г. Стерлитамак, школа № 17, учитель **Орлова Е.В.**;

— Мехнина Наталья, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**

Наталья Мехнина представила также правильные решения ребусов по информатике, опубликованных в 155-м номере нашей газеты.

## ЗАДАЧНИК

### Ответы, решения, разъяснения к заданиям, опубликованным в газете “В мир информатики” ранее

#### 1. Задача “Кто выше?”

Напомним условие: “Курсанты военного училища построены в виде каре (прямоугольника). В каждой колонне выбрали самого высокого курсанта, а из них — самого низкого. Им оказался Петр Иванов. Затем в каждой шеренге выбрали самого низкого курсанта, а из них — самого высокого. Им оказался Иван Петров. Кто выше — Иван или Петр?”

#### Решение

Упомянутое в условии каре (прямоугольник) будем моделировать в виде таблицы, строки которой соответствуют шеренгам с курсантами, а столбцы — колоннам (каждый курсант моделируется клеткой этой таблицы). Пусть Петр Иванов стоял в колонне, выделенной синим, Иван Петров находился в шеренге, выделенной зеленым (рис. 1).


ВП

ПИ

Рис. 1

Рассмотрим место на пересечении указанных колонны и шеренги.

Допустим, на этом месте был Иван Петров (рис. 2). В этом случае он — ниже ростом, чем Петр Иванов, так как Петр — самый высокий в синей колонне.


ВП

ПИ

Рис. 2

Допустим, на этом месте стоял Петр Иванов (рис. 3). В этом случае Иван Петров также ниже Петра, так как Иван — самый низкий в зеленой шеренге.


ВП

ПИ

Рис. 3

Если же на указанном месте стоял кто-то “третий” (пусть это был Дмитрий Сидоров — см. рис. 4), то можем утверждать, что Дмитрий выше Ивана Петрова, так как Иван — самый низкий в зеленой шеренге, и в то же время Дмитрий ниже Петра Иванова, так как Петр — самый высокий в синей колонне. Значит, и в этом случае Иван ниже Петра.


ВП

ДС

ПИ

Рис. 4

Ответ: Петр Иванов выше Ивана Петрова.

Правильный ответ представили:

— Базылев Юрий и Галушкова Карина, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**;

— Бергер Юлия, Свердловская обл., г. Нижняя Салда, школа № 7, учитель **Зорихина Н.Ю.**;

— Бобылев Павел, Москва, гимназия № 1540, учитель **Савенкова Л.С.**;

— Васильева Юлия, Евченко Мария, Кольтякова Анна, Семенова Наталья и Харитоновна Елена, Республика Башкортостан, г. Стерлитамак, школа № 17, учитель **Орлова Е.В.**;

— Глазкова Екатерина и Семяшкин Иван, Республика Коми, г. Сыктывкар, МОУ “Лицей народной дипломатии”, учитель **Гранаткина О.М.**;

— Диков Андрей и Филимонова Галина, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Доминов Иван, Дробизов Олег, Ермоленко Роман, Крамков Сергей, Мусина Вероника, Мутвина Александра, Протасевич Яна, Танкова Оксана, Туркменов Роман, Фролов Сергей и Югов Дмитрий, Красноярский край, г. Канск, школа № 5, учитель **Павлова Н.Н.**;

— Катышева Елизавета, Пыров Егор и Телегин Дмитрий, г. Струнино Владимирской обл., школа № 11, учитель **Волков Ю.П.**;

— Лаврентьева Виктория, Никандрова Анжелика и Сидорина Екатерина, Москва, Центр образования № 1406 (школа для обучающихся с нарушениями слуха), учитель **Миронова А.А.**;

— Мехнина Наталья, Суминова Марина и Ядзевичус Стас, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Тарасюк Степан, средняя школа поселка Озеры Красноярского края, учитель **Филипченко И.С.**;

— Толкунов Василий и Хомякова Анна, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Юхтенко Илья, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**

### Новое задание

Предлагаем читателям смоделировать обсуждаемую задачу средствами электронной таблицы Microsoft Excel. Для этого заполните некоторый диапазон ячеек случайными значениями роста из интервала 160–190 см (используя функцию СЛЧИС). Затем, применяя соответствующие функции, найдите значения, упомянутые в условии, и сравните их. Ответ получите по формуле в одной из ячеек (конечно, можно использовать и другие ячейки). Проведите ряд экспериментов, нажимая функциональную клавишу **F9**.

### 2. Статья “Шесть вопросов. Вариант 2”

#### Ответы

1. 28 декабря 1984 года была открыта станция “Рыбацкое” Ленинградского (Санкт-Петербургского) метро.

2. Сваи многих старинных венецианских домов сделаны из лиственницы.

3. По русским поверьям, через неделю после Троицы наступала так называемая “русальная” неделя.

4. Картина, о которой шла речь в вопросе, называется “Шоколадница” (“Прекрасная шоколадница”). Автор — Жан Этьен Лиотар.

5. Название страны — Китай происходит от слова “кита”, обозначавшего связку жердей, которые применялись при постройке укреплений.

6. Самыми древними бумажными деньгами считаются банкноты Китая.

#### Правильные ответы прислали:

— Алексеева Татьяна и Енин Борис, Совхозная средняя школа, Московская обл., Серебряно-Прудский р-н, поселок Успенский, учитель **Жарикова Е.Н.**;

— Базылев Юрий, Галушкова Карина и Максименко Илья, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**;

— Бергер Юлия, Зорихин Алексей и Шишкина Анастасия, г. Нижняя Салда, школа № 7, учитель **Зорихина Н.Ю.**;

— Бобылев Павел, Москва, гимназия № 1540, учитель **Савенкова Л.С.**;

— Васильева Юлия, Евченко Мария, Кольтякова Анна, Ноздрин Данила, Семенова Наталья и Харитоновна Елена, Республика Башкортостан, г. Стерлитамак, школа № 17, учитель **Орлова Е.В.**;

— Глазкова Екатерина и Семяшкин Иван, Республика Коми, г. Сыктывкар, МОУ “Лицей народной дипломатии”, учитель **Гранаткина О.М.**;

— Диков Андрей и Филимонова Галина, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Довгалёва Кристина, Комлева Вероника, Кузьменко Алексей, Мерзляков Никита, Метелкин Сергей, Савостьянов Влад, Старкова Анастасия, Чернышенко Дмитрий и Шемет Татьяна, Красноярский край, г. Канск, школа № 5, учитель **Павлова Н.Н.**;

— Кирченкова Надежда, Москва, Центр образования № 1406 (школа для обучающихся с нарушениями слуха), учитель **Миронова А.А.**;

— Манохина Татьяна, основная школа поселка Михинский, Воронежская обл., Таловский р-н, учитель **Удалова А.А.**;

— Мехнина Наталья, Полухович Максим, Суминова Марина и Ядзевичус Стас, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Микулик Илья, г. Астрахань, школа № 33 им. Н.А. Мордвориной, учитель **Лепехина С.М.**;

— Неофитова Елена, средняя школа села Янтиково, Чувашская Республика, учитель **Неофитова Н.Н.**;

— Печерский Александр, Москва, кадетская школа-интернат № 5 “Преображенский кадетский корпус”, учитель **Хлопков Г.К.**;

— Шадрин Юлия, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Воеводина Р.В.**;

— Юхтенко Илья, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**

Большинство приславших ответы привели развернутые сведения по заданным вопросам.

### 3. Задача “В одном купе”

*Ответ.* Антонов — поэт, читал прозу. Борисенко — историк, читал пьесу. Симонян — прозаик, читал стихи. Дмитриев — драматург, читал книгу по истории.

*Ответ представили:*

— Базылев Юрий и Галушкова Карина, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**;

— Васильева Юлия, Евченко Мария, Кольтякова Анна, Ноздрин Данила, Семенова Наталья, Харитоновна Елена и Шафиева Алина, Республика Башкортостан, г. Стерлитамак, школа № 17, учитель **Орлова Е.В.**;

— Диков Андрей и Филимонова Галина, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Ермоленко Роман, Кожевина Татьяна, Кочергин Андрей, Майснер Роза и Фролов Сергей, Красноярский край, г. Канск, школа № 5, учитель **Павлова Н.Н.**;

— Катышева Елизавета и Синицын Никита, г. Струнино Владимирской обл., школа № 11, учитель **Волков Ю.П.**;

— Лаврентьева Виктория, Никандрова Анжелика и Сидорина Екатерина, Москва, Центр образования № 1406 (школа для обучающихся с нарушениями слуха), учитель **Миронова А.А.**;

— Мехнина Наталья и Ядзевичюс Стас, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Толкунов Василий и Хомякова Анна, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**

#### 4. Числовой ребус со словом-числом КУБ

Напомним, что необходимо было решить числовой ребус **КУБ = (Б + У + К)<sup>3</sup>**

*Решение*

Прежде всего ясно, что сумма **Б + У + К** больше 4, так как  $4^3$  — двузначное число, и меньше 10, поскольку  $10^3 = 1000$ . Проверив соответствующие значения этой суммы, можно установить, что решением ребуса является следующее: **К = 5, У = 1, Б = 2** ( $512 = (2 + 1 + 5)^3$ ).

*Правильный ответ прислали:*

— Абелькалина Анастасия, Вахрушев Иван, Горбунов Роман, Довгалева Кристина, Дранишников Артем, Надёжин Максим, Николаева Анна, Пороцкий Никита, Филимонов Сергей, Фролов Сергей и Югов Дмитрий, Красноярский край, г. Канск, школа № 5, учитель **Павлова Н.Н.**;

— Абубакирова Аэлита, Хайбрахманова Регина, Хайбрахманова Эльвира и Хазиев Наиль, средняя школа села Сулеево им. Р.Г. Галеева, Республика Татарстан, Альметьевский р-н, учитель **Валиева Д.И.**;

— Андрющенко Александр и Свистунов Николай, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

— Базылев Юрий и Галушкова Карина, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**;

— Бергер Юлия, Свердловская обл., г. Нижняя Салда, школа № 7, учитель **Зорихина Н.Ю.**;

— Бобылев Павел, Москва, гимназия № 1540, учитель **Савенкова Л.С.**;

— Габлевский Эдуард и Савинов Сергей, г. Белово Кемеровской обл., поселок Краснобродский, школа № 31, учитель **Зайцева Л.А.**;

— Глазкова Екатерина и Семяшкин Иван, Республика Коми, г. Сыктывкар, МОУ “Лицей народной дипломатии”, учитель **Гранаткина О.М.**;

— Григоренко Василий, Григоренко Дмитрий, Есипова Мария, Круглякова Мария и Яснова Дарья, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Катышева Елизавета и Тананаева Ксения, г. Струнино Владимирской обл., школа № 11, учитель **Волков Ю.П.**;

— Мехнина Наталья, Суминова Марина и Ядзевичюс Стас, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Печерский Александр, Москва, кадетская школа-интернат № 5 “Преображенский кадетский корпус”, учитель **Хлопков Г.К.**;

— Фокеева Нина, Республика Башкортостан, г. Уфа, гимназия № 3 им. А.М. Горького, учитель **Болдырева С.В.**

#### 5. Статья “Календарь из часов”

Ответы на вопросы и другие материалы, связанные с заданиями для самостоятельной работы, предложенными в статье, прислали:

— Базылев Юрий, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**;

— Гусева Варвара, Казанкова Екатерина и Киселев Алексей, Санкт-Петербург, г. Зеленогорск, лицей № 445, учитель **Зорина Е.М.**;

— Изотова Алла и Пономарева Анастасия, г. Струнино Владимирской обл., школа № 11, учитель **Волков Ю.П.**;

— Яновский Виталий, Москва, гимназия № 1530, учитель **Шамшев М.В.**

Все представленные программы написаны на языке Паскаль (Юрий Базылев, Алла Изотова и Анастасия Пономарева подготовили также лист электронной таблицы Microsoft Excel, с помощью которого можно решать задачи). В качестве “небольшой ложки дегтя в бочке меда” заметим, что в программах использовались многократно вложенные друг в друга условные операторы (лучше применить оператор варианта). Тем не менее редакция решила наградить всех перечисленных читателей дипломами. Поздравляем!

#### 6. Головоломка “Как получить три квадрата?”

*Правильные ответы представили:*

— Базылев Юрий и Галушкова Карина, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**;

— Бергер Юлия, Свердловская обл., г. Нижняя Салда, школа № 7, учитель **Зорихина Н.Ю.**;

— Бобылев Павел, Москва, гимназия № 1540, учитель **Савенкова Л.С.**;

— Глазкова Екатерина и Семяшкин Иван, Республика Коми, г. Сыктывкар, МОУ “Лицей народной дипломатии”, учитель **Гранаткина О.М.**;

— Довгалёва Кристина, Крамков Сергей, Мерзляков Никита, Мусина Вероника, Мутовина Александра, Пороцкий Никита, Старкова Анастасия, Танкова Оксана, Туркменов Роман и Фролов Сергей, Красноярский край, г. Канск, школа № 5, учитель **Павлова Н.Н.**;

— Домашина Анастасия и Чумакова Елена, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н, учитель **Краснёнкова Л.А.**;

— Мехнина Наталья и Полюхович Максим, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Печерский Александр, Москва, кадетская школа-интернат № 5 “Преображенский кадетский корпус”, учитель **Хлопков Г.К.**;

— Стульников Дмитрий и Яснова Наталья, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Хазиев Наиль, средняя школа села Сулеево им. Р.Г. Галеева, Республика Татарстан, Альметьевский р-н, учитель **Валиева Д.И.**

Отметим ответ Карины Галушковой из школы № 1 поселка Надвоицы, Республика Карелия, разработавшей презентацию Microsoft PowerPoint, иллюстрирующую решение задачи.

## 7. Задача “День рождения”

Правильное решение задачи прислали Пыров Егор, Телегин Дмитрий и Шоршин Кирилл, г. Струнино Владимирской обл., школа № 11, учитель **Волков Ю.П.** Скорее всего небольшое количество ответов связано с тем, что условие задачи было оформлено в виде красочной “врезки”, на которую многие читатели не обратили внимания ☹. Благодаря Егора, Дмитрия и Кирилла, редакция решила еще раз предложить задачу для решения.

*“Алина и Алена были очень правдивыми девушками и говорили неправду в виде исключения только в день своего рождения. 26 октября кто-то спросил каждую из них: “Когда твой день рождения?” Алина сказала: “Он был вчера”, Алена ответила: “Он будет завтра”. На следующий день их спросили то же самое, и, как ни странно, девушки ответили точно так же. Когда родилась каждая из них?”*

Мы ждем ответы до 1 октября.

## 8. Задача “120 тетрадей”

Напомним условие задачи: “Имеются 120 тетрадей. Сколько времени понадобится, чтобы взять 96 тетрадей, если на подсчет трех тетрадей уходит две секунды?”

Ответы представили:

— Анисимова Диана, Владимиров Виталий, Владимирова Снежана, Емелькова Виктория, Петрова Алена, Семенова Екатерина и Яковлева Анастасия,

основная школа села Именевое, Республика Чувашия, Красноармейский р-н, учитель **Тимофеева И.А.**;

— Борисов Валентин, средняя школа села Бабино, Удмуртская Республика, Завьяловский р-н, учитель **Мерзлякова Р.В.**;

— Кренгель Евгений и Харламов Виталий, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Неофитова Елена, средняя школа села Янтиково, Чувашская Республика, учитель **Неофитова Н.Н.**;

— Товмасын Арсен, средняя школа села Урман, Республика Башкортостан, Иглинский р-н, учитель **Товмасын М.Г.**;

— Чудинов Роман и Юзифович Кристина, Москва, гимназия № 1530, учитель **Козырева О.В.**

Большинство приславших ответы правильно решили, что отсчитывать 96 тетрадей — нерационально. Лучше (быстрее) отсчитать  $120 - 96 = 24$  тетради и отложить их — останутся нужные 96 тетрадей. На подсчет 24 тетрадей уйдет  $24/3 * 2 = 16$  сек.

Решения головоломок “судоку”, опубликованных в 155-м и 156-м номерах нашей газеты (“Информатика” № 2–3/2011), представили<sup>1</sup>:

— Абубакирова Аэлига, Хайбрахманова Регина, Хайбрахманова Эльвира и Хазиев Наиль, средняя школа села Сулеево им. Р.Г. Галеева, Республика Татарстан, Альметьевский р-н, учитель **Валиева Д.И.**;

— Адамюк Анастасия, Грибанов Владлен, Дукач Светлана, Дюбарова Анастасия, Клименко Надежда, Кирсанова Алеся, Пирогов Егор, Романова Надежда, Сагитова Зильда, Смолярова Наталья, Шошина Екатерина, г. Лесосибирск Красноярского края, поселок Стрелка, школа № 8 им. Константина Филиппова, учитель **Лопатин М.А.**;

— Базылев Юрий и Галушкова Карина, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**;

— Васильева Юлия, Евченко Мария, Кольтякова Анна, Ноздрин Данила, Семенова Наталья и Харитоновна Елена, Республика Башкортостан, г. Стерлитамак, школа № 17, учитель **Орлова Е.В.**;

— Галушкова Карина, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**;

— Глазкова Екатерина и Семяшкин Иван, Республика Коми, г. Сыктывкар, МОУ “Лицей народной дипломатии”, учитель **Гранаткина О.М.**;

— Домашина Анастасия и Чумакова Елена, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н, учитель **Краснёнкова Л.А.**;

— Дробизов Олег, Колпаков Александр и Сайбель Кристина, Красноярский край, г. Канск, школа № 5, учитель **Павлова Н.Н.**;

— Зуева Дарья, Савинов Сергей, Сидорова Виктория и Хорева Евгения, г. Белово Кемеровской обл., поселок Краснобродский, школа № 31, учитель **Зайцева Л.А.**;

— Казанкова Екатерина, Санкт-Петербург, г. Зеленогорск, лицей № 445, учитель **Зорина Е.М.**;

— Катышева Елизавета, Тананаева Ксения и Телегин Дмитрий, г. Струнино Владимирской обл., школа № 11, учитель **Волков Ю.П.**;

<sup>1</sup> Список будет дополнен в следующем выпуске газеты.

— Кондаков Роман, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Манохина Татьяна, основная школа поселка Михинский, Воронежская обл., Таловский р-н, учитель **Удалова А.А.**;

— Мехнина Наталья и Суминова Марина, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Мухин Борис, Олёнкин Олег и Плахонина Виктория, основная школа поселка Михинский, Воронежская обл., Таловский р-н, учитель **Удалова А.А.**;

— Николаев Даниил и Сметанин Илья, Республика Башкортостан, г. Стерлитамак, гимназия № 5, учитель **Пучкина С.А.**;

— Рожкова Вера, средняя школа села Кипцы, Саратовская обл., Екатериновский р-н, учитель **Омельченко С.Ю.**;

— Фокеева Нина, Республика Башкортостан, г. Уфа, гимназия № 3 им. А.М. Горького, учитель **Болдырева С.В.**;

— Шестакова Олеся, Куминская средняя школа, Тюменская область, Ханты-Мансийский автономный округ — Югра, Кондинский р-н, учитель **Шишгина О.В.**

Программы решения задач, предложенных для самостоятельной работы в статье “Обмены, обме-

ны...”, представил Мартынов Андрей, Воронежская обл., поселок Каменка, средняя школа № 1 им. Героя Советского Союза В.П. Захарченко, учитель **Старикова М.Е.**

Редакция решила наградить Андрея дипломом. Поздравляем!

Ответ на задачу “120 тетрадей” прислали также:

— Гараева Регина, Республика Татарстан, г. Бавлы, гимназия № 4, учитель **Матянина О.М.**;

— Гурьянова Анастасия, Кириченко Анастасия, Севастьянова Дарья и Ходюк Екатерина, Московская обл., г. Краснознаменск, Московский кадетский корпус “Пансион воспитанниц МО РФ”, учитель **Федорова Л.А.**;

— Джанхотов Альбек, г. Волгоград, поселок Горьковский, школа № 8, учитель **Брусенская М.С.**;

— Каюмов Адель и Каюмов Ансель, Республика Татарстан, г. Бавлы, гимназия № 4, учитель **Шафиков Н.Р.**;

— Юхтенко Илья, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**

Правильное решение ребусов по информатике, опубликованных в 155-м номере нашей газеты, представила также Исаева Анжелика, средняя школа села Средний Васюган, Томская обл., Каргасокский р-н, учитель **Вторушина Н.А.**

## GAMES.EXE

### Еще раз игра “Доминошки на клетчатой доске”

► В газете “В мир информатики” № 155 (“Информатика” № 21/2011) была описана такая игра: “Двое играют в такую игру на клетчатой доске размером  $10 \times 10$  клеток. За ход разрешается накрыть любые две соседние клетки доминошкой (прямоугольником  $1 \times 2$  клетки) так, чтобы доминошки не перекрывались. Проигрывает тот, кто не сможет сделать ход. Кто выиграет — начинающий ее или делающий ход вторым?”

Благодаря Юрия Базылева из школы № 1 поселка Надвоицы, Республика Карелия (учитель **Богданова Л.М.**), Екатерину Глазкову и Ивана Семяшкина из МОУ “Лицей народной дипломатии”, Республика Коми, г. Сыктывкар (учитель **Гранаткина О.М.**) и Сергея Хотеева из московской гимназии № 1530 (учитель **Шамшев М.В.**), приславших ответ, редакция решила предложить читателям еще раз подумать над этой задачей. Предлагаем также при решении вспомнить задачу “Монеты на столе”, опубликованную в газете “В мир информатики” № 148 (“Информатика” № 18/2010). В ней выигрывает тот, кто начи-



нает игру независимо от размеров стола. Первым ходом он должен положить монету так, чтобы центры монеты и стола совпали. После этого на каждый ход второго игрока начинающий должен класть монету симметрично относительно центра стола. При такой стратегии после каждого хода первого игрока позиция будет симметрична. Поэтому если возможен очередной ход второго игрока, то возможен и симметричный ему ответный ход первого. Когда-то делающий ход вторым разместить монету на столе не сможет. Следовательно, побеждает первый игрок. Предлагаем также вспомнить игру “Короли на клетчатой доске”, рассмотренную в 159-м номере нашей газеты (“Информатика” № 5/2011). Во всех этих задачах использовалась симметрия размещения предметов.

Итак, мы ждем ответы на вопрос об игре “Доминошки на клетчатой доске”. Срок представления ответов — 1 октября.

### Отгадывание задуманного двузначного числа

► Предложите своему товарищу задумать двузначное число. Потом пусть он умножит сумму цифр этого числа на 11, вычтет задуманное число и сообщит вам результат. Узнав его, вы сможете определить задуманное число. Как?

## Итоги конкурса № 84 “Гривенники и пятиалтынные”

▶ Напомним, что необходимо было, используя программу Microsoft Excel, решить старинную задачу: “Сколько существует способов составить сумму в 6 рублей из одних только гривенников (10 коп.) и пятиалтынных (15 коп.)?”

### Решение

Если обозначить число гривенников в некоей комбинации, соответствующей условию задачи, буквой  $x$ , а количество пятиалтынных — буквой  $y$ , то можем записать:

$$10x + 15y = 600,$$

откуда  $y = (600 - 10x)/15$ .

Используя полученную зависимость, можем получить на листе следующие данные:

	A	B	C	D
1	Гривенников	Пятиалтынных		Подходит?
2	$x$	$y$		
3	0	40		1
4	1	39,33333333		0
...				

Ясно, что так как количество пятиалтынных не может быть нецелым, то искомое число способов составления суммы в 6 рублей равно числу целых значений в столбце B. Но как подсчитать это число? Один из вариантов решения такой:

1) в столбце D (например) получить 0 или 1 в зависимости от того, является ли значение в соответствующей строке столбца B целым (для этого можно использовать функцию ЕСЛИ, а также функции ЦЕЛОЕ, ОКРУГЛ или ОТБР);

2) искомое число в ячейке C65 получить, используя формулу =СУММ(D1:D62). Оно равно 21.

### Ответы представили:

— Андрющенко Александр и Свистунов Николай, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

— Диков Андрей и Филимонова Галина, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Арчеков Роман и Орозбаев Алексей, г. Волгоград, лицей № 9, учитель **Широкова Л.В.**;

— Базылев Юрий и Галушкова Карина, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**;

— Горюнова Галина и Чуйков Андрей, средняя школа поселка Озеры Красноярского края, учитель **Филипченко И.С.**;

— Дернов Антон и Хотеев Даниил, Москва, гимназия № 1530, учитель **Козырева О.В.**;

— Исмаилова Эльнара, Республика Башкортостан, село Стерлибашево, школа № 2, учитель **Загидуллин Н.Р.**;

— Неофитова Елена, средняя школа села Янтиково, Чувашская Республика, учитель **Неофитова Н.Н.**

В ряде писем правильное значение ответа приведено без обоснования или получено подсчетом “вручную”, а не с использованием возможностей программы Microsoft Excel.

Победителями конкурса признаны: Александр Андрющенко, Роман Арчеков, Карина Галушкова, Антон Дернов, Алексей Орозбаев, Николай Свистунов и Даниил Хотеев, решившие задачу с использованием стандартных функций Microsoft Excel, как предлагалось в условии. Все они будут награждены дипломами. Поздравляем!

Решение двух японских головоломок “судоку”, опубликованных в 156-м номере газеты, представили также:

— Балахонова Анастасия, Бочкарев Николай, Патрушева Валерия, Финк Андрей и Фролова Екатерина, средняя школа села Средний Васюган, Томская обл., Каргасокский р-н, учитель **Вторушина Н.А.**;

— Ефимов Петр и Никитин Василий, Караклинская средняя школа, Чувашская Республика, Канашский р-н, учитель **Макарова Л.Ф.**

Василий Никитин прислал также решение трех “судоку”, опубликованных в одном из осенних номеров нашей газеты.

Гурьянова Анастасия и Кириченко Анастасия, Московская обл., г. Краснознаменск, Московский кадетский корпус “Пансион воспитанниц МО РФ” (учитель **Федорова Л.А.**), представили правильное решение задач “День рождения”, “Кто выше?” и “В одном купе”.

Луцинова Ирина, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н, учитель **Краснёнкова Л.А.**, прислала правильное решение задач “В одном купе” и “День рождения”.

## Конкурс № 85

▶ Несмотря на каникулярное настроение читателей, предлагаем поучаствовать в конкурсе, условие которого связано с задачей “Кто выше?” (см. рубрику “Задачник” в этом выпуске). Проведите описанный эксперимент и пришлите результаты в редакцию. Срок представления ответов (в виде листов электронной таблицы Microsoft Excel) — 1 октября.

Напомним также, что были объявлены два конкурса “на лето” (см. 159-й и 160-й номера нашей газеты).

Уважаемые коллеги!

Для поощрения самых активных участников конкурсов, проводимых газетой-вкладкой “В мир информатики”, редакция может направить вам электронный вариант диплома.

Заявку на диплом просьба прислать в редакцию электронной (адрес: [vmi@1september.ru](mailto:vmi@1september.ru)) или обычной почтой в мае-июне. Оформление дипломов будет проводиться в учебном заведении.



Издательский дом

# ПЕРВОЕ СЕНТЯБРЯ

## НОВЫЙ ЭТАП РАЗВИТИЯ

Уважаемые коллеги! Напоминаем, что со II полугодия 2011 года все наши предметно-методические газеты становятся журналами: цветными, 64-страничными, в каждом номере CD-диск с материалами к урокам (для непредметных изданий с дополнительными материалами). **ЖУРНАЛЫ ВЫХОДЯТ В БУМАЖНОЙ И ЭЛЕКТРОННОЙ ВЕРСИЯХ.**



## ПРОДОЛЖАЕТСЯ ПОДПИСКА НА ЭЛЕКТРОННЫЕ ВЕРСИИ ПРЕДМЕТНО-МЕТОДИЧЕСКИХ ЖУРНАЛОВ!

### ЭЛЕКТРОННАЯ ВЕРСИЯ

- Полностью соответствует бумажной
- Выходит гарантированно в срок
- Легко распечатывается на принтере
- Стоит существенно дешевле
- Доставляется по Интернету



На электронные версии журналов можно подписаться

**НА САЙТЕ [www.1september.ru](http://www.1september.ru)**

**И ПОЛУЧИТЬ МЕСЯЦ ПОДПИСКИ БЕСПЛАТНО**



**699  
рублей**

– цена подписки  
для индивидуальных  
подписчиков  
и организаций  
за полгода

БЕСПЛАТНЫЕ ОЗНАКОМИТЕЛЬНЫЕ ВЫПУСКИ ЖУРНАЛОВ МОЖНО СКАЧАТЬ  
НА САЙТЕ [www.1september.ru](http://www.1september.ru)

## Джон Соча: известный неизвестный



► Более десятилетия Norton Commander был непременным атрибутом любого персонального компьютера. Это потом были Volkov, Total, Windows, FAR. Первенство панелей Norton'a никто не оспаривает. А вот авторство... Авторство тоже никто не оспаривает. Питер Нортон не является автором знаменитой программы. И он это никогда не оспаривал. Первые версии Norton Commander, в которых и были реализованы все ключевые интерфейсные решения и знаменитые панели, были написаны Джоном Соча (John Socha). Имя этого программиста со временем погружалось все глубже в недра справочного файла программы, а сам Джон ушел из компании Norton Computing и в последнее время посвящает свое время преимущественно... созданию моделей железных дорог.

Сам Джон так описывает свою работу над Norton Commander:

"Я начал работать над программой, которая теперь известна как Norton Commander, в конце 1984 г., еще будучи студентом кафедры прикладной физики Корнельского университета. Первая версия была целиком написана на ассемблере. Но программирование на нем занимало довольно много времени,

и скоро я переключился на некую смесь ассемблера и Си. Впрочем, большая часть критичного кода была по-прежнему написана на ассемблере.

В то время еще, конечно, не было названия "Norton Commander", и я не работал в Norton Computing. Мы называли программу VDOS – Visual DOS. Я продолжил работу над VDOS, когда занял должность руководителя отдела разработки Norton Computing. В результате в 1986 г. была закончена разработка первой коммерческой версии, которая и была названа Norton Commander".

Некоторое время Джон также возглавлял работу над версиями утилит Нортона для компьютеров Macintosh. Он написал несколько книг, в частности, "Язык ассемблер" в соавторстве с Питером Нортоном.

В Сети практически нет фотографий Джона, которые можно было бы воспроизвести в полиграфическом качестве. Поэтому тем, кто хочет посмотреть, как выглядит истинный автор одной из самых знаменитых программ, можно порекомендовать блог Джона <http://blogs.socha.com/>. Небольшая фотография там имеется. ◀

Help

View

RenMov

Copy

Delete

Edit

UserMn

MkFold

Quit

Screen

# ИНФОРМАТИКА