№ 16
16–31 августа 2010

Тема номера

Основы криптографии

Ваш баланс

В конце июня, после визита Президента в Кремниевую долину, Twitter в России сильно прибавил в популярности. Я прямо представляю себе, как чиновники разных уровней — те, кому “положено” держать на рабочем месте портрет Президента, суматошно теребят своих айтишников, чтобы те “наладили им Twitter”. Ничего, ничего, пусть посуетятся ☺.

Я, кстати, тоже подписался на @KremlinRussia — интересно же. Но этот факт, пока не более чем забавный, заставил меня лишний раз задуматься о том, что я ведь практически постоянно подключен к различным информационным каналам. Фактически я никогда не бываю “один”. Это произошло как-то само собой. Ну, во-первых, конечно, “обычный” мобильный — голосовые звонки и sms. Это же устройство (на данный момент — смартфон под управлением Android) обеспечивает мобильную электронную почту, Skype и ICQ. Основным каналом общения, безусловно, является электронная почта, поэтому для надежности мобильная почта дублируется BlackBerry. В России я, наверное, был одним из первых частных пользователей этого замечательного устройства. Также “для общего развития” я подключен, в том числе и мобильными клиентами, к некоторым социальным сетям — Facebook, Twitter и т.д.

На пользу ли такая информационная включенность 24 х 7 х 365? Не знаю. Честное слово — не знаю. С одной стороны, вроде бы удобно — ты всегда в курсе событий; если что-то произойдет — хорошее ли, плохое, ты узнаешь об этом первым. Ты никого не “тормозишь” — когда на тебе замкнуто большое количество дел, в которые включено много народу, это важно. Это плюсы. Минус же я фактически уже обозначил выше — ты никогда не бываешь один. А ведь известная формула “думать некогда, делать надо” далеко не всегда приводит к наилучшим результатам. Часто как раз надо подумать — спокойно, отстраненно, уединенно. Побыть в тишине, пойти в парк, посидеть у костра. Где баланс между пользой (несомненной!) от постоянной информационной включенности и вредом от необдуманных (ну, скажем — недодуманных) решений, принятых “на ходу”? Безусловно, общего ответа на этот вопрос нет, но сам по себе он для меня интересен, как модельный. Ведь нет никакого сомнения, что дальше — больше. Вопрос скорости коммуникации, передачи информации между людьми скоро вообще перестанет кого-либо волновать. Все будут исходить из того, что всем и все можно сообщить мгновенно.

Вот решит Президент, например, на второй срок баллотироваться, и через секунду все об этом уже знают...

Пока я писал эту колонку, Дмитрий Анатольевич написал в Twitter “Погода испортилась. Вместо вертолетов едем в Торонто машинами. Природа — как у нас в Карелии”.

Сергей Островский, гл. редактор,
so@1september.ru

Содержание

Прилагается CD

ИНФОРМАТИКА
Издательский дом «ПЕРВОЕ СЕНТЯБРЯ»
Методическая газета для учителей информатики
32291 — индекс по каталогу «Газеты. Журналы» агентства «Роспечать»
79066 — индекс по каталогу «Почта России»

Избранные разделы разрабатываемого учебника

Информационная безопасность: основы криптографии

К.Ю. Поляков,
А.П. Шестаков,
Е.А. Еремин

Что такое шифрование?

Один из методов защиты информации от неправомерного доступа — это *шифрование*, то есть кодирование специального вида.

Шифрование — это преобразование (кодирование) открытой информации в зашифрованную, недоступную для понимания посторонних.

Шифрование применяется в первую очередь для передачи секретной информации по незащищенным каналам связи. Шифровать можно любую информацию — тексты, рисунки, звук, базы данных и т.д.

Человечество применяет шифрование с того момента, как появилась секретная информация, которую нужно было скрыть от врагов. Первое известное науке шифрованное сообщение — египетский текст, в котором вместо принятых тогда иероглифов были использованы другие знаки.

Методы шифрования и расшифровывания сообщения изучает наука *криптология*, история которой насчитывает около четырех тысяч лет. Она состоит из двух ветвей: *криптографии* и *криптоанализа*.

Криптография — это наука о способах шифрования информации.

Криптоанализ — это наука о методах и способах вскрытия шифров.

Обычно предполагается, что сам алгоритм шифрования известен всем, но неизвестен его *ключ*, без которого сообщение невозможно расшифровать. В этом заключается отличие шифрования от простого кодирования, при котором для восстановления сообщения достаточно знать только алгоритм кодирования.

Ключ — это параметр алгоритма шифрования (шифра), позволяющий выбрать одно конкретное преобразование из всех вариантов, предусмотренных алгоритмом. Знание ключа позволяет свободно зашифровывать и расшифровывать сообщения.

Все шифры (системы шифрования) делятся на две группы — *симметричные* и *несимметричные* (с открытым ключом).

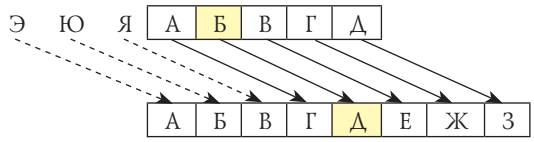
Симметричный шифр означает, что и для шифрования, и для расшифровывания сообщений используется один и тот же ключ. В системах **с открытым ключом** используются два ключа — открытый и закрытый, которые связаны друг с другом с помощью некоторых ма-

тематических зависимостей. Информация шифруется с помощью открытого ключа, который доступен всем желающим, а расшифровывается с помощью закрытого ключа, известного только получателю сообщения.

Криптостойкость шифра — это устойчивость шифра к расшифровке без знания ключа.

Стойким считается алгоритм, который для успешного раскрытия требует от противника недостижимых вычислительных ресурсов, недостижимого объема перехваченных сообщений или такого времени, что по его истечении защищенная информация будет уже неактуальна.

Шифр Цезаря[1] — один из самых известных и самых древних шифров. В этом шифре каждая буква заменяется на другую, расположенную в алфавите на заданное число позиций k вправо от нее. Алфавит замыкается в кольцо, так что последние символы заменяются на первые. Вот пример шифра Цезаря (со сдвигом 3):



Знаменитая фраза "ПРИШЕЛ УВИДЕЛ ПОБЕДИЛ" при использовании шифра Цезаря со сдвигом 3 будет закодирована так:

ТУЛЫИО ЦЕЛЗИО ТСДИЗЛО

Если первая буква алфавита имеет код 0, вторая — код 1 и т.д., алгоритм шифрования может быть выражен формулой

$$y = (x + k) \bmod n,$$

где x — код исходного символа, k — величина сдвига, y — код символа-замены, n — количество символов в алфавите, а запись $(x + k) \bmod n$ обозначает остаток от деления $x + k$ на n. Операция взятия остатка от деления необходима для того, чтобы "замкнуть" алфавит в кольцо. Например, при использовании русского алфавита (32 буквы[2]) для буквы "Я" (код 31) получаем код заменяющего символа $(31 + 3) \bmod 32 = 2$, это буква "В".

Ключом для шифра Цезаря служит сдвиг k; если его знать, то сообщение легко расшифровать. Для этого используется формула

$$x = (y - k + n) \bmod n.$$

Шифр Цезаря относится к шифрам *простой подстановки*, так как каждый символ исходного сообщения заменяется на другой символ из того же алфавита. Такие шифры легко раскрываются с помощью частотного анализа, потому что в каждом языке частоты встречаемости букв примерно постоянны для любого достаточно большого текста.

Значительно сложнее сломать *шифр Виженера*[3], который стал естественным развитием шифра Цезаря. Для использования шифра Виженера используется ключевое слово, которое задает переменную величину сдвига. Например, пусть ключевое слово — "ЗАБЕГ". По таблице определяем коды букв:

0	1	2	3	4	5	6	7	8	9	
А	Б	В	Г	Д	Е	Ж	З	И	К	...

Получаем: "З" — 7, "А" — 0, "Б" — 1, "Е" — 5, "Г" — 3. Это значит, что для кодирования первой буквы используется сдвиг 7, для кодирования второй — 0 (символ не меняется) и т.д. Для пятой буквы используется сдвиг 3, а для шестой — снова 7 (начали "проходить" кодовое слово сначала). Фраза "ПРИШЕЛ УВИДЕЛ ПОБЕДИЛ" при использовании шифра Виженера с ключом "ЗАБЕГ" будет закодирована в виде "ЦРЙЭИТ ФЗЛЛЕМ ТХБЖЙЛТ".

Шифр Виженера обладает значительно более высокой криптостойкостью, чем шифр Цезаря. Это значит, что его труднее раскрыть — подобрать нужное ключевое слово. Теоретически, если длина ключа равна длине сообщения и каждый ключ используется только один раз, шифр Виженера взломать невозможно.

Контрольные вопросы

1. Чем отличаются понятия "шифрование" и "кодирование"?
2. Что такое ключ?
3. Как называется наука, изучающая методы шифрования?
4. Что такое симметричный шифр? Какая проблема возникает при использовании симметричного шифра, если участники переписки находятся в разных странах?
5. Что такое несимметричные шифры? На чем основана их надежность?
6. Что такое криптостойкость алгоритма? Какой алгоритм считается криптостойким?

Задачи

1. Зашифруйте с помощью шифра Цезаря со сдвигом 6 высказывание "ЛЮДИ ОХОТНО ВЕРЯТ ТОМУ, ЧЕМУ ЖЕЛАЮТ ВЕРИТЬ".
(Ответ: "СДКО ФЫФШУФ ИЛЦЕШ ШФТЩ, ЭЛТЩ МЛСЖДШ ИЛЦОШВ".)
2. Напишите программу, которая выполняет шифрование строки с помощью шифра Цезаря.
3. *Попытайтесь расшифровать сообщение, закодированное шифром Цезаря с неизвестным сдвигом: "ХШЖНПУТ ФКХКОЙКТ". Для этого можно написать программу. (Ответ: "РУБИКОН ПЕРЕЙДЕН", сдвиг 5.)
4. Используя шифр Виженера с ключом "ЛЕНА", зашифруйте сообщение "НЕЛЬЗЯ ОБИЖАТЬ ГОСТЯ".
(Ответ: "ШКШЬТД ОМНУАЭБ ГЩЦЯЯ".)
5. *Измените программу для шифрования Цезаря так, чтобы учитывать букву Ё.
6. *Измените программу для шифрования Цезаря так, чтобы учитывать и букву Ё, и пробел.

[1] Назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки.

[2] Будем считать, что буквы Е и Ё совпадают.

[3] Назван по имени Блеза Виженера, швейцарского дипломата XVI века.

Хэширование и пароли

В современных информационных системах часто используется вход по паролю. Если при этом где-то хранить пароли всех пользователей, система становится очень ненадежной, потому что "утечка" паролей позволит сразу получить доступ к данным. С другой стороны, кажется, что пароли обязательно где-то нужно хранить, иначе пользователи не смогут войти в систему. Однако это не совсем так. Можно хранить не пароли, а некоторые числа, полученные в результате обработки паролей. Простейший вариант — сумма кодов символов, входящих в пароль. Для пароля "A123" такая сумма равна

$$215 = 65\ (\text{код "A"}) + 49\ (\text{код "1"}) + 50\ (\text{код "2"}) + 51\ (\text{код "3"}).$$

Фактически мы определили функцию $H(M)$, которая сообщение M *любой* длины превращает в короткий код m *заданной* длины. Такая функция называется *хэш-функцией* (от англ. *hash* — "мешанина", "крошить"), а само полученное число — *хэш-кодом*, *хэш-суммой* или просто *хэшем* исходной строки. Важно, что, зная хэш-код, невозможно восстановить исходный пароль! В этом смысле хэширование — это *необратимое* шифрование.

Итак, вместо пароля "A123" мы храним число 215. Когда пользователь вводит пароль, мы считаем сумму кодов символов этого пароля и разрешаем вход в систему только тогда, когда она равна 215. И вот здесь возникает проблема: существует очень много паролей, для которых наша хэш-функция дает значение 215, например, "B023". Такая ситуация — совпадение хэш-кодов различных исходных строк — называется *коллизией* (англ. *collision* — "столкновение"). Коллизии будут всегда — ведь мы "сжимаем" длинную цепочку байт до числа. Казалось бы, ничего хорошего не получилось: если взломщик узнает хэш-код, то, зная алгоритм его получения, он сможет легко подобрать пароль с таким же хэшем и получить доступ к данным. Однако это произошло потому, что мы выбрали плохую хэш-функцию.

Математики разработали надежные (но очень сложные) хэш-функции, обладающие особыми свойствами:

1) хэш-код очень сильно меняется при малейшем изменении исходных данных;

2) при известном хэш-коде m невозможно за приемлемое время найти сообщение M с таким хэш-кодом ($H(M) = m$);

3) при известном сообщении M невозможно за приемлемое время найти сообщение M_1 с таким же хэш-кодом ($H(M) = H(M_1)$).

Здесь выражение "невозможно за приемлемое время" (или "вычислительно невозможно") означает, что эта задача решается только перебором вариантов (других алгоритмов не существует), а количество вариантов настолько велико, что на решение уйдут сотни и тысячи лет. Поэтому даже если взломщик получил хэш-код пароля, он не сможет за приемлемое время получить сам пароль (или пароль, дающий такой же хэш-код).

Чем длиннее пароль, тем больше количество вариантов. Кроме длины, для надежности пароля важен используемый набор символов. Например, очень легко подбираются пароли, состоящие только из цифр. Если же пароль состоит из 10 символов и содержит латинские буквы (заглавные и строчные) и цифры, перебор вариантов (англ. *brute force* — метод "грубой силы") со скоростью 10 млн. паролей в секунду займет более 2000 лет.

Надежные пароли должны состоять не менее чем из 7–8 символов; пароли, состоящие из 15 символов и более, взломать методом "грубой силы" практически невозможно. Нельзя использовать пароли типа "12345", "qwerty", свой день рождения, номер телефона. Плохо, если пароль представляет собой известное слово, для этих случаев взломщики используют подбор по словарю. Сложнее всего подобрать пароль, который представляет собой случайный набор заглавных и строчных букв, цифр и других знаков[4].

Сегодня для хэширования в большинстве случаев применяют алгоритмы MD5, SHA1 и российский алгоритм, изложенный в ГОСТ Р 34.11 94 (он считается одним из самых надежных). В криптографии хэш-коды чаще всего имеют длину 128, 160 и 256 бит.

Хэширование используется также для проверки правильности передачи данных. Различные контрольные суммы, используемые для проверки правильности передачи данных, — это не что иное, как хэш-коды.

Контрольные вопросы

1. Что такое хэширование? хэш-функция? хэш-код?
2. Какую хэш-функцию вы используете, когда начинаете искать слово в словаре?
3. Что такое коллизии? Почему их должно быть как можно меньше?
4. Какие требования предъявляются к хэш-функциям, которые используются при хранении паролей?
5. Что значит "вычислительно невозможно"?
6. Взломщик узнал хэш-код пароля администратора сервера. Сможет ли он получить доступ к секретным данным на сервере?
7. Какие свойства пароля влияют на его надежность?
8. Как выбрать надежный пароль?
9. Какие алгоритмы хэширования сейчас чаще всего применяются?
10. *Предложите какой-нибудь свой метод хэширования. Подумайте, как часто при его использовании могут происходить коллизии.

Современные алгоритмы шифрования

Государственным стандартом шифрования в России является алгоритм, зарегистрированный как ГОСТ 28147-89. Он является *блочным* шифром, то есть шифрует не отдельные символы, а 64-битные блоки. В алгоритме предусмотрено 32 цикла преобразования

[4] Однако такой пароль сложно запомнить.

данных с 256-битным ключом, за счет этого он очень надежен (обладает высокой криптостойкостью). На современных компьютерах раскрытие этого шифра путем перебора ключей ("методом грубой силы") займет не менее сотен лет, что делает такую атаку бессмысленной. В США используется аналогичный блочный шифр AES.

В Интернете популярен алгоритм RSA, названный так по начальным буквам фамилий его авторов — Р.Райвеста (R.Rivest), А.Шамира (A.Shamir) и Л.Адлемана (L.Adleman). Это алгоритм с *открытым* ключом, стойкость которого основана на использовании свойств простых чисел. Для его взлома нужно разложить очень большое число на простые сомножители. Эту задачу сейчас умеют решать только перебором вариантов. Поскольку количество вариантов огромно, для раскрытия шифра требуется много лет работы современных компьютеров.

Для применения алгоритма RSA требуется построить открытый и секретный ключи следующим образом.

1. Выбрать два больших простых числа, p и q.

2. Найти их произведение $n = p \cdot q$ и значение $\varphi = (p - 1) \cdot (q - 1)$.

3. Выбрать число e ($1 < e < \varphi$), которое не имеет общих делителей с φ.

4. Найти число d, которое удовлетворяет условию $d \cdot e = k \cdot \varphi + 1$ для некоторого целого k.

5. Пара значений (e, n) — это открытый ключ RSA (его можно свободно публиковать), а пара (d, n) — это секретный ключ.

Передаваемое сообщение нужно сначала представить в виде последовательности чисел в интервале от 0 до $n - 1$. Для шифрования используют формулу

$$y = x^e \bmod n,$$

где x — число исходного сообщения, (e, n) — открытый ключ, y — число закодированного сообщения, а запись $x^e \bmod n$ обозначает остаток от деления x^e на n. Расшифровка сообщения выполняется по формуле

$$x = y^d \bmod n.$$

Это значит, что зашифровать сообщение может каждый (открытый ключ общеизвестен), а прочитать его — только тот, кто знает секретный показатель степени d.

Для лучшего понимания мы покажем работу алгоритма RSA на простом примере. Возьмем $p = 3$ и $q = 7$, тогда находим $n = p \cdot q = 21$ и $\varphi = (p - 1) \cdot (q - 1) = 12$. Выберем $e = 5$, тогда равенство $d \cdot e = k\varphi + 1$ выполняется, например, при $d = 17$ (и $k = 7$). Таким образом, мы получили открытый ключ (5, 21) и секретный ключ (17, 21).

Зашифруем сообщение "123" с помощью открытого ключа (5,21). Получаем

$$1 \Rightarrow 1^5 \bmod 21 = 1,\ 2 \Rightarrow 2^5 \bmod 21 = 11,$$
$$3 \Rightarrow 3^5 \bmod 21 = 12,$$

то есть зашифрованное сообщение состоит из чисел 1, 11 и 12. Зная секретный ключ (17, 21), можно его расшифровать:

$$1 \Rightarrow 1^{17} \bmod 21 = 1,\ 11 \Rightarrow 11^{17} \bmod 21 = 2,$$
$$12 \Rightarrow 12^{17} \bmod 21 = 3.$$

Мы получили исходное сообщение.

Конечно, вы заметили, что при шифровании и расшифровке приходится вычислять остаток от деления очень больших чисел (например, 12^{17}) на n. Оказывается, само число 12^{17} в этом случае находить не нужно. Достаточно записать в обычную целочисленную переменную, например x, единицу, а потом 17 раз выполнить преобразование $x = 12 \cdot x \bmod 21$. После этого в переменной x будет значение $12^{17} \bmod 21 = 3$. Попробуйте доказать правильность этого алгоритма.

Для того чтобы расшифровать сообщение, нужно знать секретный показатель степени d. А для этого, в свою очередь, нужно найти сомножители p и q, такие что $n = p \cdot q$. Если n велико, это очень сложная задача, ее решение перебором вариантов на современном компьютере займет сотни лет. В 2009 году группа ученых из разных стран в результате многомесячных расчетов на сотнях компьютеров смогла расшифровать сообщение, зашифрованное алгоритмом RSA с 768-битным ключом. Поэтому сейчас надежными считаются ключи с длиной 1024 бита и более. Если будет построен работающий квантовый компьютер, взлом алгоритма RSA будет возможен за очень небольшое время.

При использовании симметричных шифров всегда возникает проблема: как передать ключ, если канал связи ненадежный? Ведь, получив ключ, противник сможет расшифровать все дальнейшие сообщения. Для алгоритма RSA этой проблемы нет, сторонам достаточно обменяться открытыми ключами, которые можно показывать всем желающим.

У алгоритма RSA есть еще одно достоинство: его можно использовать для цифровой подписи сообщений. Она служит для доказательства авторства документов, защиты сообщений от подделки и умышленных изменений.

Цифровая подпись — это набор символов, который получен в результате шифрования сообщения с помощью личного секретного кода отправителя.

Отправитель может передать вместе с исходным сообщением такое же сообщение, зашифрованное с помощью своего секретного ключа (это и есть цифровая подпись). Получатель расшифровывает цифровую подпись с помощью открытого ключа. Если она совпала с незашифрованным сообщением, можно быть уверенным, что его отправил тот человек, который знает секретный код. Если сообщение было изменено при передаче, оно не совпадет с расшифрованной цифровой подписью. Так как сообщение может быть очень длинным, для сокращения объема передаваемых данных чаще всего шифруется не все сообщение, а только его хэш-код.

Во многих современных программах есть возможность шифровать данные с паролем. Например, офисные пакеты *OpenOffice.org* и *Microsoft Office* позволяют шифровать все создаваемые документы (для их просмотра и/или изменения нужно ввести пароль). При создании архива (например, в архиваторах 7ZIP,  *WinRAR*,  *WinZip*) также можно установить пароль, без которого извлечь файлы невозможно.

В простейших задачах для шифрования файлов можно использовать бесплатную программу *Шифровальщик* (*www.familytree.ru/ru/cipher.htm*), версии которой существуют для *Linux* и *Windows*. Программы *TrueCrypt* (*www.truecrypt.org*), *BestCrypt* (*www.jetico.com*) и *FreeOTFE* (*freeotfe.org*) создают логические диски-контейнеры, информация на которых шифруется. Свободно распространяемая программа *DiskCryptor* (*diskcryptor.net*) позволяет шифровать разделы жестких дисков и даже создавать шифрованные флэш-диски и CD/DVD-диски.

Программа *GnuPG* (*gnupg.org*) также относится к свободному программному обеспечению. В ней поддерживаются симметричные и несимметричные шифры, а также различные алгоритмы электронной цифровой подписи.

Контрольные вопросы

1. Какой алгоритм шифрования принят в России в качестве государственного стандарта?
2. Что такое блочный алгоритм шифрования?
3. К какому типу относится алгоритм RSA? На чем основана его криптостойкость?
4. Что такое цифровая подпись?
5. Как можно использовать алгоритм RSA для цифровой подписи?

Задачи

1. *Напишите программу, которая строит открытый и секретный ключи RSA для небольших множителей p и q.
2. *Напишите программу, которая шифрует и расшифровывает сообщения с помощью алгоритма при небольших значениях открытого и секретного ключей.

Стеганография

При передаче сообщений можно не только применять шифрование, но и скрывать сам факт передачи сообщения.

> **Стеганография** — это наука о скрытой передаче информации путем скрытия самого факта передачи информации.

Древнегреческий историк Геродот описывал, например, такой метод: на бритую голову раба записывалось сообщение, а когда его волосы отрастали, он отправлялся к получателю, который брил его голову и читал сообщение.

Классический метод стеганографии — *симпатические* (невидимые) *чернила*, которые проявляются только при определенных условиях (нагрев, освещение, химический проявитель). Например, текст, написанный молоком, можно прочитать при нагреве.

Сейчас стеганография занимается скрытием информации в текстовых, графических, звуковых и видеофайлах с помощью программного "внедрения" в них нужных сообщений.

Простейший способ — заменять младшие биты файла, в котором закодировано изображение. Причем это нужно сделать так, чтобы разница между исходным и полученным рисунками была неощутима для человека. Например, если в черно-белом рисунке (256 оттенков серого) яркость каждого пикселя кодируется 8 битами. Если поменять 1–2 младших бита этого кода, "встроив" туда текстовое сообщение, фотография, в которой нет четких границ, почти не изменится. При замене 1 бита каждый байт исходного текстового сообщения хранится в младших битах кодов 8 пикселей. Например, пусть первые 8 пикселей рисунка имеют такие коды:

10101101	10010100	00101010	01010010
10101010	10101010	10101011	10101111

Чтобы закодировать в них код буквы "И" (11001000_2), нужно изменить младшие биты кодов:

1010110**1**	1001010**1**	0010101**0**	0101001**0**
1	1	0	0

1010101**1**	1010101**0**	1010101**0**	1010111**0**
1	0	0	0

Получателю нужно взять эти младшие биты и "собрать" их вместе в один байт.

Для звуков используются другие методы стеганографии, основанные на добавлении в запись коротких условных сигналов, которые обозначают 1 и 0 и не воспринимаются человеком на слух. Возможна также замена одного фрагмента звука на другой.

Для подтверждения авторства и охраны авторских прав на изображения, видео и звуковые файлы применяют *цифровые водяные знаки* — внедренную в файл информацию об авторе. Они получили свое название от старых водяных знаков на деньгах и документах. Для того чтобы установить авторство фотографии, достаточно расшифровать скрытую информацию, записанную с помощью водяного знака.

Иногда цифровые водяные знаки делают видимыми (текст или логотип компании на фотографии или на каждом кадре видеофильма). На многих сайтах, занимающихся продажей цифровых фотографий, видимые водяные знаки размещены на фотографиях, предназначенных для предварительного просмотра.

Контрольные вопросы

1. Что такое стеганография?
2. Какие методы стеганографии существовали до изобретения компьютеров?
3. Как можно добавить текст в закодированное изображение?
4. На чем основаны методы стеганографии для звуковых и видеоданных?
5. Что такое цифровые водяные знаки? Зачем они используются?

Турнир по программированию “История криптографии”

А.В. Зязин,
С.Ю. Катышев

С 1991/92 учебного года в Москве, а с 2007 года и в ряде других городов России проводится Межрегиональная олимпиада школьников по математике и криптографии. Значительное число участников этих олимпиад показывает интерес школьников к задачам данной тематики, находящейся на стыке математики и информатики. На решение задач олимпиады отводится ограниченное время и не предполагается использование электронно-вычислительных средств. В то же время существует класс интересных задач, основанных на исторических примерах шифров, в которых наряду с математическим содержанием присутствует алгоритмическая составляющая. Если под решением таких задач подразумевать нетривиальную программную реализацию, то они по формату подходят для турниров по программированию.

Для проведения турнира по программированию среди школьников по направлению “История криптографии” был подготовлен комплект заданий. При их составлении подразумевалось, что учащиеся должны владеть следующим материалом:

- Основы программирования на языках высокого уровня (C++ или Pascal). Переменные, выражения, логические и условные выражения, циклы, функции;
- Одномерные и двумерные массивы. Определение, инициализация, обращение к элементам массива;
- Файлы. Создание файла, считывание, запись и добавление данных;
- Строки. Определение длины, конца строки. Сохранение и считывание строки из файла;
- Динамическое выделение и освобождение памяти, динамическое приведение типов, указатели.

Каждое из подготовленных для турнира заданий структурно состоит из:

- **Условия задачи**, написанного в игровой форме, призванной заинтересовать школьника и мотивировать к решению;
- **Комментария к задаче**, в котором приводится более строгое математическое описание используемой модели шифра, а также рассматривается пример применения данной модели на конкретных данных;
- **Примеров** тестовых входных и выходных данных, для того чтобы школьники могли самостоятельно оценить правильность разработанного алгоритма и его программной реализации;
- **Указания к решению**. В указании дается наводящее соображение или подсказка;
- **Решения**. В данном пункте описывается подход к решению данной задачи. Приводится алгоритм решения, а также программный код на языке C++;
- **Тестов**. Составлен ряд тестов, позволяющих выявить недостатки в решении, а также его полноту.

Школьникам предоставляются условия задачи, комментарии к задаче, а также пример тестовых входных и выходных данных. В исключительных случаях учащимся может быть предоставлено указание к решению. Ожидаемые затраты по времени на одно задание — 1–1,5 часа.

В апреле 2010 года турнир по материалам приведенных далее заданий был проведен в московской гимназии № 1514. Участники разбились на команды по три-четыре человека. Школьниками были представлены варианты решения всех предложенных задач. Некоторые задачи были решены отличными от авторского решения способами. По окончании турнира был проведен разбор заданий.

Памятник А.Тьюрингу в Университете Суррея, Великобритания

Шифр “Поворотная решетка”

Условия задачи

Ключом шифра “Поворотная решетка” является трафарет, изготовленный из квадратного листа клетчатой бумаги размера $n*n$ (n — четно). Некоторые из

клеток вырезаются. Одна из сторон трафарета помечена. При наложении этого трафарета на чистый лист бумаги четырьмя возможными способами (помеченной стороной вверх, вправо, вниз, влево) его вырезы полностью покрывают всю площадь квадрата, причем каждая клетка оказывается под вырезом ровно один раз. Буквы сообщения, имеющего длину $n*n$, последовательно вписываются в вырезы трафарета, сначала наложенного на чистый лист бумаги помеченной стороной вверх. После заполнения всех вырезов трафарета буквами сообщения трафарет располагается в следующем положении и т.д. После снятия трафарета на листе бумаги оказывается зашифрованное сообщение.

Необходимо по известному трафарету найти исходное сообщение.

Формат входного файла input.txt

В первой строке записано число n. В следующих двух строках записаны координаты "дырочек" в трафарете: в первой строке — номер строки дырочки, во второй — номер столбца. Нумерация строк и столбцов начинается с нуля. В четвертой строке написан зашифрованный текст.

Информация во входном файле задана корректно, то есть длина текста ровно $n*n$.

Формат выходного файла output.txt

В файл необходимо вывести расшифрованный текст.

Пример 1

input.txt

```
12
0 0 0 0 0 1 1 1 1 2 2 2 2 2 3 3 3 3 3
3 4 4 4 4 4 4 5 5 5 5 5 5 6
0 1 2 4 10 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3
4 5 11 0 1 2 3 4 5 0 1 2 3 4 5 0
ключичы, явол м шифяется ра,
оттрафарправлеет, инной
тезготовлеграмленныимразмез квадра
n*nратного (n - листа четно клетч).
Некатой йоботрыеу маг
```

output.txt

```
ключом шифра, отправленной телеграммы,
является трафарет, изготовленный из
квадратного листа клетчатой бумаги
размера n*n (n - четно). Некоторые
```

Ограничения

По времени: 1 секунда

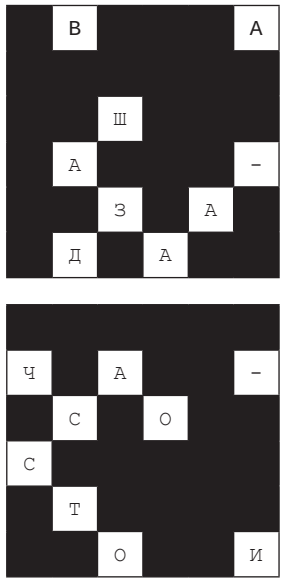
По памяти: 64 мегабайта

Комментарии

Покажем зашифрование на примере текста

ВАША_ЗАДАЧА_СОСТОИТ_В_НАХОЖДЕНИИ_...

При использовании приведенного выше трафарета зашифрование будет производиться следующим образом



после еще двух поворотов получим таблицу

Д	В	Т	Е	_	А
Ч	В	А	_	Н	_
Н	С	Ш	О	А	И
С	А	И	Х	_	_
.	Т	З	.	А	.
О	Д	О	А	Ж	И

Тогда зашифрованным сообщением будет:

ДВТЕ_АЧВА_Н_НСШОАИСАИХ__.ТЗ.А.ОДОАЖИ

Указания

Первое, что необходимо заметить: число дырочек в трафарете есть $n*n/4$, так как всего клеточек в таблице $n*n$, а за четыре поворота трафарета все клетки таблицы будут заполнены. Трафарет удобно хранить в массиве из 0 и 1 размерами $n \times n$.

Далее необходимо заметить, что расшифрование производится следующим образом: накладывается трафарет и выписываются те символы, которые "видны в дырочки".

Решение

Считываем из файла трафарет следующим образом: в массив $n \times n$ на месте дырочек поставим единицы, а на остальных поставим нули. Текст запишем в массив $n \times n$.

Теперь, пробегая по массиву текста, выписываем те символы, для которых в массиве, задающем трафарет, на соответствующем месте стоит единица.

Поворачиваем трафарет на 90 градусов, заменяя координаты по правилу $(i, j) \to (j, (n-1)-i)$, выписываем символы и так далее.

Шифр Цезаря

Условия задачи

Галльским воинам посчастливилось перехватить римского гонца, везшего свиток с важным посланием римскому дивизиону от самого Юлия Цезаря.

Развернув свиток, галлы обнаружили не имеющую на первый взгляд смысла последовательность символов и потребовали объяснений от гонца. Гонец же, будучи уверенным в том, что все равно галлам не удастся ничего прочитать, рассказал им, что все послания Цезарь шифрует особым образом. Каждый символ он заменяет другим, отстоящим от него в алфавите на фиксированное число позиций вправо, считая, что после "Я" алфавит начинается с начала. Также гонец назвал галлам слово, присутствующее в послании. Помогите галлам прочитать важное послание Юлия Цезаря.

Формат входного файла input.txt

В первой строке записано известное слово. Во второй строке — текст перехваченного послания, состоящий из символов алфавита *А* = (*А*, *Б*, *В*, ..., *Э*, *Ю*, *Я*) и пробела. Длина текста менее 1000. Длина известной последовательности символов менее 50.

Формат выходного файла output.txt

В файл необходимо сохранить текст исходного послания.

Пример файлов input.txt и output.txt

input.txt

```
ЧЕРЕМУХА
ДЗУЗКЮВХСТСОБВЪЗУЗПЦШГВУГФТЦФНГОЛВФЕСЛВН
ОЗМНЛЗВЛВТГШЩЪЛЗВОЛФХЯБВОЛТЮВРГЖЦЕГОЛВОСТ
ГЕЫЛЗФБВТСЪНЛВЁГОНЛВЕСУСДЯЛВЛВЁСОЦДЛВТСВЕ
ЗФЗРРЗПЦВУГЖСФХРСВЁСХСЕЛОЛВЦЙЗВЁРЗКЖГВЛВП
ЦШЛВЙЦЙЙГОЛВЦВФХЗРВТУЛЁУЗХЮЗВФСОРЩЗПВЕЗФЗ
ОЮВДЮОЛВЛВУГФХЗРЛБВЛВТХЛЩЮВЛВРГФЗНСПЮЗВЛВ
ЖЗХЛ
```

output.txt

```
БЕРЕЗЫ ТОПОЛЯ ЧЕРЕМУХА РАСПУСКАЛИ СВОИ
КЛЕЙКИЕ И ПАХУЧИЕ ЛИСТЬЯ ЛИПЫ НАДУВАЛИ
ЛОПАВШИЕСЯ ПОЧКИ ГАЛКИ ВОРОБЬИ И ГОЛУБИ
ПО ВЕСЕННЕМУ РАДОСТНО ГОТОВИЛИ УЖЕ ГНЕЗДА
И МУХИ ЖУЖЖАЛИ У СТЕН ПРИГРЕТЫЕ СОЛНЦЕМ
ВЕСЕЛЫ БЫЛИ И РАСТЕНИЯ И ПТИЦЫ И НАСЕКОМЫЕ
И ДЕТИ
```

Ограничения

По времени: 1 секунда
По памяти: 1 мегабайт

Комментарии

Опишем более строго математическую модель шифра Цезаря. Зададим упорядоченный алфавит $A=(a_0,a_1,a_2,...,a_{n-1})$, содержащий n символов, и выберем ключ — число k, большее нуля и меньшее n. Именно на это число позиций символ шифрованного текста отстоит от соответствующего ему символа исходного (открытого) текста. Пусть имеется текст, составленный из символов алфавита A, который мы хотим зашифровать на ключе k. Тогда мы выбираем первый символ этого текста, находим его позицию (индекс) i в алфавите A, вычисляем сумму $i + k$ и, если эта сумма меньше n, то заменяем первый символ с индексом $i+k$, если же сумма больше или равна n, то на символ с индексом $i + k - n$. В общем виде символ с индексом i мы заменяем на символ с индексом $(i + k) \bmod n$, где mod n — операция нахождения остатка от деления на n. Аналогично мы поступаем со вторым символом открытого текста, с третьим и так далее.

Рассмотрим следующий пример. Пусть задан алфавит *А* = (, *А*, *Б*, *В*, ..., *Э*, *Ю*, *Я*) и выбран ключ $k = 3$. Для каждого символа алфавита определим, в какой символ он будет переходить при шифровании, и отобразим это в таблице:

А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О
В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С

П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я		А	Б

Тогда легко понять, что фраза "МАЛЕНЬКОЕ ДЕЛО ЛУЧШЕ БОЛЬШОГО БЕЗДЕЛЬЯ" после зашифрования станет такой: "ПГОЗРЯНСЗВЖЗОСВОЦЪЫЗВДСОЯЫСЁСВДЗКЖЗОЯБ"

Указания

Сначала необходимо понять, как производится расшифрование. Пусть задан алфавит $A=(a_0,a_1,a_2,...,a_{n-1})$, содержащий n символов, и известен ключ k. Тогда мы выбираем первый символ шифрованного текста, находим его индекс i в алфавите A, вычисляем разность $i - k$ и, если она неотрицательна, заменяем первый символ шифрованного текста на символ алфавита A с индексом $i - k$, если же разность отрицательна, то на символ с индексом $i - k + n$. В общем виде, при расшифровании мы заменяем символ с индексом i на символ с индексом $(i - k + n) \bmod n$. Аналогично поступаем со вторым символом шифрованного текста, третьим, четвертым и так далее.

Теперь предположим, что нам известно слово или в более общем случае последовательность символов длины l, точно присутствующая в открытом тексте. Как это может пригодиться, если мы знаем лишь шифрованный текст и хотим найти открытый? Допустим, что нам также известна позиция, начиная с которой эта последовательность входит в текст. Тогда рассмотрим последовательность символов, входящую в шифрованный текст с той же позиции и той же длины l. Затем составим соответствующие обеим последовательностям символов последовательности индексов. Пусть они имеют вид: $\{i_1,i_2,...,i_l\}$ — для последовательности открытого текста, и $\{j_1,j_2,...,j_l\}$ — для последовательности шифрованного. Затем составим из них новую последовательность $\{(i_1-j_1+n)\bmod n,(i_2-j_2+n)\bmod n,...,(i_l-j_l+n)\bmod n\}$, и поскольку $j_1=(i_1+k)\bmod n$, $j_2=(i_2+k)\bmod n$ и т.д., то ясно, что все члены этой последовательности равны между собой и каждый из них равен k. Это соображение и является основным при написании программы дешифрования.

Решение

Для начала сохраним используемый алфавит в массиве и напишем функцию, на вход которой подается символ, а возвращаемое значение равно индексу входного символа в алфавите.

Затем реализуем функцию расшифрования. Входные параметры этой функции: указатель на строку, содержащую шифрованный текст, ключ k, указатель на строку, в которой нужно сохранить расшифрованный текст. Процедура расшифрования проста и описана в указании. Проходим последовательно по всему шифрованному тексту, для каждой буквы находим индекс, вычитаем из него k по модулю n, и буква алфавита с таким индексом и есть буква открытого текста.

Немного сложнее нахождение открытого текста в случае, когда ключ неизвестен. Входные параметры этой функции: указатель на строку, содержащую шифрованный текст, указатель на строку, содержащую известную последовательность символов открытого текста, указатель на строку, в которой нужно сохранить полученный текст. В основе процедуры дешифрования лежит соображение, описанное в указании. Сначала мы выбираем из шифрованного текста последовательность первых l символов и находим последовательность их индексов, затем находим последовательность индексов известной последовательности открытого текста и рассматриваем особую почленную разность последовательностей индексов. Тогда если все разности равны, то известное слово входило в открытый текст, начиная с первой позиции. Если же нет, то выбираем вторые l символов шифрованного текста и повторяем процедуру. Так до тех пор, пока на некотором шаге все разности не будут равны. Как только это произойдет, сразу станет известен ключ и останется лишь применить процедуру расшифрования.

Шифр "Сцитала"

Условия задачи

Проводя раскопки на территории Древнего Рима, археологи обнаружили тонкий кожаный ремешок с нанесенными на него символами. Попытка прочесть их оказалась неудачной. Там не было известных древних слов. Тогда археологи обратились к истории криптографии и выяснили, что это послание зашифровано особым образом, который в наше время называют "шифр "Сцитала". Помогите археологам прочесть исторически важное сообщение.

Комментарии

Рассмотрим подробнее, как применялся шифр "Сцитала". На жезл наматывалась тонкая кожаная лента, затем на этой ленте вдоль жезла писали строки сообщения, после чего ленту снимали с жезла и отдавали гонцу. Чтобы прочитать сообщение, ленту необходимо снова намотать на жезл такого же диаметра, поэтому ясно, что ключом в данном методе зашифрования является диаметр жезла. По одной из легенд, Архимед изобрел метод дешифрования, или антисциталу, — конусообразный жезл, на который наматывалась лента, и проводился поиск диаметра, при котором возникает осмысленный текст. В наше время этот метод возможно применить следующим образом. Пусть длина исходного текста N символов и задан ключ $k < N$. Рассчитывают число $m = \left\lceil \frac{N}{k} \right\rceil$, где $\left\lceil \frac{N}{k} \right\rceil_N$ — наименьшее целое число, большее или равное числу $\frac{N}{k}$, и затем вписывают текст по строчкам слева направо в таблицу размером k строк на m столбцов. Оставшиеся пустыми клетки заполняют случайными символами из алфавита. После этого текст считывают по столбцам сверху вниз. То, что получилось, и есть шифрованный текст. Рассмотрим пример.

Пусть надо зашифровать текст длины 30 символов "СОВЕРШЕННО СЕКРЕТНОЕ СООБЩЕНИЕ" на ключе $k = 4$. Тогда $m = \left\lceil \frac{30}{4} \right\rceil = 8$. Нарисуем таблицу и впишем в нее текст:

С	О	В	Е	Р	Ш	Е	Н
Н	О		С	Е	К	Р	Е
Т	Н	О	Е		С	О	О
Б	Щ	Е	Н	И	Е	К	Т

Теперь считаем текст по столбцам и получим шифрованное сообщение: "СНТБООНЩВ ОЕЕСЕНРЕ ИШКСЕЕРОКНЕОТ".

Формат входного файла input.txt

Файл содержит текст зашифрованного собщения, состоящий из символов алфавита $A = (А, Б, В, ..., Э, Ю, Я)$ и пробелов. Длина сообщения менее 1000.

Формат выходного файла output.txt

В файл необходимо сохранить текст дешифрованного сообщения.

Пример файлов input.txt и output.txt

input.txt

```
СОВЕРШЕННО СЕКРЕТНОЕ СООБЩЕНИЕ
```

output.txt

```
СНТБООНЩВ ОЕЕСЕНРЕ ИШКСЕЕРОКНЕОТ
```

Ограничения

По времени: 1 секунда
По памяти: 1 мегабайт

Указания

Сначала необходимо понять, как производится расшифрование. Пусть длина шифрованного текста составляет M символов и известен ключ k. Тогда нужно вписать шифрованный текст по столбцам сверху вниз в таблицу из k строк и $m = \frac{M}{k}$ столбцов, а затем считать по строкам слева направо. Тот текст, который при этом получится, и будет исходным (открытым) текстом.

Если же ключ неизвестен, то заметим, что в любом случае должны быть выполнены неравенства $1 < k < M$, поскольку в противном случае шифрованный текст отличается от открытого лишь наличием нескольких случайных символов в конце. Таким образом, набор всевозможных различных ключей равен $K = \{2, 3, 4, ..., M–1\}$, и если расшифровать зашифрованный текст последовательно на каждом из этих ключей, то получим $M–2$ различных текстов, один из которых является открытым текстом. Остается лишь определить, какой именно. Как это возможно определить? Тексты, полученные при расшифровании на неверном ключе, будут бессмысленны и нечитаемы. Но

бессмысленность и нечитаемость — те категории, которыми, скорее, мыслит человек, а не компьютер. И тут нам поможет одно из самых простых необходимых условий на осмысленность текста. Суть этого условия заключается в том, что в русских текстах никогда не встречаются определенные биграммы — пары символов алфавита, стоящие на соседних местах в тексте. Например, ни одно слово не начинается с твердого знака, и, следовательно, биграмма " Ъ" (пробел + твердый знак) запретная и никогда не встретится в осмысленном тексте.

Решение

Для начала необходимо сохранить используемый алфавит в массиве и написать функцию, на вход которой подается символ, а возвращаемое значение равно позиции входного символа в алфавите (индексу).

Затем нужно реализовать функцию расшифрования. Входные параметры этой функции: указатель на строку, содержащую шифрованный текст, ключ, указатель на строку, в которой нужно сохранить расшифрованный текст. Процедура расшифрования достаточно проста и описана в указании. Вычисляем число столбцов таблицы (число строк равно ключу), вписываем по столбцам сверху вниз в получившуюся таблицу шифрованный текст, затем считываем из нее текст по строкам слева направо, который и будет открытым текстом.

Для отсева неосмысленных текстов необходимо создать двумерный массив размера n на n, где n — количество символов алфавита, заполнить его единицами, а далее, для каждой запретной пары, определить индексы первого i и второго j символов в алфавите A и записать в ij-й элемент массива ноль.

Затем остается реализовать функцию нахождения открытого текста в случае, когда ключ неизвестен. Входные параметры этой функции: указатель на строку, содержащую шифрованный текст, указатель на двумерный массив запретных пар, указатель на строку, в которой нужно сохранить найденный текст. Процедура его получения строится следующим образом: выбираем первый ключ из набора всевозможных различных ключей K, применяем функцию расшифрования шифрованного текста на выбранном ключе, к расшифрованному тексту применяем необходимое условие. Если текст содержит запретные пары, значит, выбранный ключ не является истинным ключом зашифрования, и мы выбираем следующий ключ из набора и повторяем описанную выше процедуру. Если же текст не содержит запретных пар, то считаем, что выбранный ключ истинный и что результат расшифрования является открытым текстом.

Диск Альберти

Условия задачи

Агент отправил сообщение, которое было перехвачено. При обыске у него было изъято приспособление, похожее на диск Альберти. Сообщить начальное положение дисков агент наотрез отказался. Требуется восстановить открытый текст, зная, что в нем содержится кодовое имя агента — некоторая последовательность символов алфавита A.

Комментарии

Известный философ, живописец и архитектор Леон Альберти в 1466 году написал труд о шифрах. В этой работе был предложен шифр, основанный на использовании шифровального диска. Сам Альберти назвал его "шифром, достойным королей". Вариант приспособления для шифрования, о котором идет речь в условии, представляет собой пару соосных дисков разного диаметра. Больший из них — неподвижный, его окружность разделена на 34 равных сектора, в которые вписаны символы алфавита $A = \{А, Б, В, Г, Д, Е, Ё, ..., Э, Ю, Я\}$ — прописные буквы русского алфавита в их естественном порядке и знак пробела. Меньший диск — подвижный, по его окружности, также разбитой на 34 сектора, вписаны переставленные в некотором известном порядке символы алфавита A. Таким образом, каждое положение меньшего диска задает однозначное соответствие между символами алфавита на внешнем и на внутреннем дисках. При зашифровании очередная буква открытого текста отыскивается на внешнем диске и заменяется соответствующей ей буквой на внутреннем диске, после чего внутренний диск поворачивается на один сектор по часовой стрелке.

Формат входного файла input.txt

Первая строка — некоторая перестановка символов алфавита A, которая расположена на внутреннем диске.

Вторая строка — шифрованный текст, состоящий из символов алфавита A.

Третья строка — кодовое имя агента.

Формат выходного файла output.txt

Текст исходного сообщения.

Пример файлов input.txt и output.txt

input.txt

```
ЗИЙКОПХАФЮЯБВГНЕЖ ЦЧДЕЛМШЬЭРСЩЪЫТУ
ТККНЯЯУЭАФЯЪЫЭЧКУЧЫЭЗЩЭЭССКХОЮЮДЖЩЛТУВИББЙФЬФО
ШДОЙМЕЗЗЖВЕУЮГМЬЬПХНБЗБТЕЙФОРЙЬЮАСЭППЕ КТТНЕШЖШ
ЦХХЕШЩЦДЯЧ ЪЪАСЮЯВТУДЕЗДКЙТЦЪ СБРЬЭФНБЧГИИЖИБТИ
ЪЗГЭЩФПЙЛЬЗЗСЖУЕЛЪТЙЧСГЧИЧЧГИЧЮБГАЩБЮЭПХЖККППС
ЬЖТВ РГЮЦАБХ У ЧЧФЫИИПВВСЗЛЛДЬЬЙТТЕНЩЪТЦЧНАББ
ГЛДУУГТЪЕЩПВИЕЩАУЙЬЦУУЕТНТЕГШШАЭХЖМВЭХЦЯГЙЩОРББ
ФШФРШЩЮМИРЪЧВССЧЛГЕБЧГФИУООЕББТЮЭАФЯЪЫЧЧЕ ЧЫЬСЦ
БНЧЯГЧИЧЧЙХГТТЗЕИТЮЭЖЪЙОШЭЯШЙЙЦЕГШВРЧСФЕПДСФОЯО
ГГЕЕШ
КАПИТАН ОЧЕВИДНОСТЬ
```

output.txt

```
ЯДЕРНОЕ ОРУЖИЕ СОВОКУПНОСТЬ ЯДЕРНЫХ БОЕПРИПАСОВ
СРЕДСТВ ИХ ДОСТАВКИ К ЦЕЛИ И СРЕДСТВ УПРАВЛЕНИЯ
ОТНОСИТСЯ К ОРУЖИЮ МАССОВОГО ПОРАЖЕНИЯ ОБЛАДАЕТ
ГРОМАДНОЙ РАЗРУШИТЕЛЬНОЙ СИЛОЙ ПО МОЩНОСТИ
ЗАРЯДОВ И ДАЛЬНОСТИ ДЕЙСТВИЯ ЯДЕРНОЕ ОРУЖИЕ
ДЕЛИТСЯ НА ТАКТИЧЕСКОЕ ОПЕРАТИВНО ТАКТИЧЕСКОЕ
И СТРАТЕГИЧЕСКОЕ ПРИМЕНЕНИЕ ЯДЕРНОГО ОРУЖИЯ В
ВОЙНЕ ГИБЕЛЬНО ДЛЯ ВСЕГО ЧЕЛОВЕЧЕСТВА КАПИТАН
ОЧЕВИДНОСТЬ
```

Указания

Очевидно, ключом при этом методе шифрования является перестановка символов на внутреннем диске и его начальное положение относительно внешнего диска. Так как перестановка символов задается во входном

файле, то задача состоит в том, чтобы перебрать все возможные значения начального положения внутреннего диска. Пусть $T' = a_{i_1} a_{i_2} a_{i_3} \ldots a_{i_k}$, $i_j \in \{0, 1, 2, ..., N-1\}$ — шифрованный текст, где k — его длина, N — мощность алфавита A, $A' = (a'_0, a'_1, ..., a'_{33})$ — строка символов, задающая порядок символов на внутреннем диске (ее мы считываем из входного файла). Пусть $f(a_{ij}) = t$, где t такое, что $a_{ij} = a_t$. Тогда расшифрованный текст на ключе $K \in \{0, 1, ..., 33\}$ будет выглядеть следующим образом: $T = a_{(f(i_1)+K) \bmod N} a_{(f(i_2)+K+1) \bmod N} \ldots a_{(f(i_k)+K+(k-1)) \bmod N}$, $i_j \in \{0, 1, 2, ..., N-1\}$, где mod N — операция нахождения остатка от деления на N.

Решение

Предлагается простая, в силу своей наглядности, последовательность действий: осуществляем расшифрование на K-м ключе, проверяем полученный текст на наличие в нем кодового слова, если находим, то прекращаем перебор и записываем данный текст в выходной файл, если нет, то повторяем процедуру на $(K+1)$-м ключе, и так далее для всех $K \in \{0, 1, ..., 33\}$. Таким образом, в худшем случае проверим все 34 ключа.

К примеру, можно сделать так: реализовать функцию расшифрования, параметрами которой будут являться указатель на строку, содержащую шифрованный текст, текущий ключ — целое число, а также указатель на строку, где требуется сохранить расшифрованный текст. В функции расшифрования мы будем использовать функцию `int position(char c)`, которая в качестве своего параметра принимает символ, а возвращаемое значение — номер позиции этого символа на внутреннем диске. Из вида открытого текста T в указании видно, как будет выглядеть выражение для расшифрования i-го символа. Далее производится поиск в строке расшифрованного на данном ключе текста кодового имени агента, для этого можно использовать любой из известных алгоритмов поиска подстроки в строке, в частности, ниже, в листинге программы, используется тривиальный способ.

Упрощенный RSA

Условия задачи

Для обмена шифрованными сообщениями Аня и Маша решили придумать и использовать систему шифрования с открытым ключом. Но придумывали они ее не "с нуля", а взяли за основу известную и широко распространенную систему RSA (применяется, например, в банковских кредитных картах) и, сделав в ней несколько упрощающих изменений, стали использовать. Поразмышляв над алгоритмом зашифрования, хакер Вася понял, что упрощения, сделанные девочками, делают шифр нестойким, и через час он уже читал их сообщения. А сможете ли вы прочитать сообщения?

Комментарии

Маша и Аня выбрали простое число p, занимающее в памяти не более слова (2 байта) и не менее байта, и ключи, такие что $e \cdot d \equiv 1 \pmod{p-1}$. Алгоритм зашифрования разбивает открытый текст на байты $x = x_1 x_2 \ldots x_n$ и для каждого i-го байта вычисляет величину $x_i^e \pmod{p}$, которая и есть i-е число зашифрованного текста, т.е. $y_i = x_i^e \pmod{p}$. Аналогично алгоритм расшифрования считывает числа шифрованного текста $y_1 y_2 \ldots y_n$ и для каждого i-го числа вычисляет величину $y_i^d \pmod{p}$, которая и есть i-й байт открытого текста. Отметим, что так получается в силу малой теоремы Ферма, которая гласит, что $x^{p-1} \equiv 1 \pmod{p}$, поскольку, если $ed \equiv 1 \pmod{p-1}$, то $ed = k(p-1)+1$ для некоторого k, и, следовательно, $y_i^d = (x_i^e)^d = x_i^{ed} = x_i^{k(p-1)+1} \equiv x_i \pmod{p}$.

Пусть p = 37 813, буква открытого текста "а", а открытый ключ 5. По таблице ASCII "а — 49", возводим 49 в пятую степень и берем остаток при делении на 37 813:

$$49^5 \pmod{37\,813} = 12\,139$$

Тогда число шифрованного текста 12 139.

Формат входного файла input.txt

В первой строке файла содержится значение числа p, во второй — значение ключа зашифрования e, в третьей — шифрованный текст, длина которого менее 1000 символов.

Формат выходного файла output.txt

В файл необходимо вывести открытый текст.

Пример файлов input.txt и output.txt

input.txt

```
37813
5
24636 39194 4667 65047 56115 4667 31000 56631
57364 55863 43267 43289 48921 31000 30272 41781
56631 49990 31000 55863 56631 64842 41870 34906
9054 56631 24636 32882 32882 56631 43267 55863
19526 43289 55863 56631 48921 17172 57364 55863
39194 30272 41870 55863 44110 18244 31000 30272
56631 65047 39194 64842 56631 43289 18244 57364
48921 17172 18244 43289 48921 64842 56631 57364
42090 55863 9863 42090 18244 43267 43267 56631
39194 47662 14682 55863 9863 55863 56631 31000
48921 57364 18244 25218
```

output.txt

```
Следует помнить, что язык C++ можно
использовать для написания программ любого
типа.
```

Ограничения

По времени: 1 секунда
По памяти: 1 мегабайт

Указания

Основной вопрос, возникающий при решении этой задачи, как найти d, зная p и e. Поскольку $ed \equiv 1 \pmod{p-1}$, то наибольший общий делитель $(e, p-1) = 1$, и, значит, применяя к этим числам алгоритм Евклида, получим цепочку следующего вида:

$$p-1=q_1e+r_1$$
$$e=q_2r_1+r_2$$
$$r_1=q_3r_2+r_3$$
$$\ldots$$
$$r_{n-2}=q_nr_{n-1}+1.$$

Затем заметим, что r_1 можно представить в виде суммы $r_1=k_1(p-1)+l_1e$, где $k_1=1, l_1=-q_1$. r_2 также можно представить в виде суммы $r_2=k_2(p-1)+l_2e$, где $k_2=-q_2, l_2=1+q_1q_2$, поскольку $r_2=e-q_2r_1=e-q_2\big((p-1)-q_1e\big)=(-q_2)(p-1)+$ $+(1+q_1q_2)e$.

Продолжая дальше эти рассуждения, можно прийти к выводу, что если положить $k_0=0, l_0=1$, то для любого $i\geq 2$ верно $k_i=k_{i-2}-q_ik_{i-1}$, $l_i=l_{i-2}-q_il_{i-1}$, а так как $r_n=1=k_n(p-1)+l_ne$, то, положив $d=l_n \bmod (p-1)$, получим, что верно сравнение $ed\equiv 1-k_n(p-1)\equiv 1(\bmod p-1)$.

Также может возникнуть вопрос о возведении числа в степень по модулю. Чтобы избежать переполнения памяти, выделенной под переменную, возводимую в степень, просто нужно на каждой итерации цикла возведения в степень вычислять остаток от деления этого числа на p.

Решение

Для начала необходимо написать функции нахождения d по известным p и e и возведения в степень по модулю. Это делается достаточно просто с учетом того, что написано в указании.

Затем напишем функцию расшифрования. Пока не достигнем конца шифрованного текста, будем считывать числа четырехбайтовой переменной, возводить ее в степень, равную ключу расшифрования, и младший байт того, что получилось, будем записывать в расшифрованный текст. Отметим, что после возведения в степень лишь младший байт четырехбайтовой переменной будет отличен от нуля, в силу алгоритмов зашифрования и расшифрования.

Остается реализовать лишь функцию получения открытого текста. На вход подаются шифрованный текст, ключ зашифрования, параметр p и указатель на строку, в которую необходимо сохранить открытый текст. Сначала вызываем функцию нахождения d по известным p и e, а после этого — функцию расшифрования с ключом расшифрования d.

Фрактальные собратья

А.И. Азевич,
Москва

Непостижимые, таинственные, захватывающие. Эти яркие эпитеты вряд ли могут передать чувство восхищения фракталами — сложными самоподобными структурами. Глядя на них, поражаешься и гармонии форм, и неистощимой фантазии виртуального художника. Современный компьютер, вооруженный программой-генератором, творит настоящие чудеса.

Условно все фракталы можно разделить на два вида: плоские и пространственные[1]. Некоторые 3D-фракталы уместно было бы назвать трехмерными аналогами своих плоских собратьев. Например, губка Мюнгера — "объемный двойник" коврика Серпинского. А у салфетки Серпинского есть пространственный аналог — пирамида, носящая имя того же ученого.

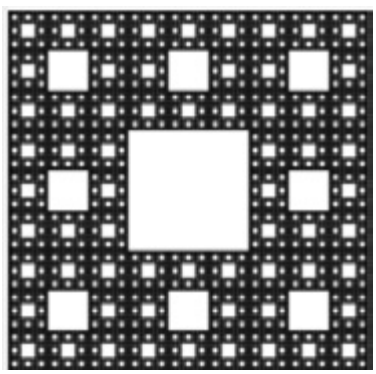


Рис. 1

Чтобы построить коврик Серпинского, надо взять квадрат. Затем разделить каждую его сторону на три равные части и через каждую пару точек деления провести прямую, параллельную соответствующей стороне квадрата. Прямые разобьют исходный квадрат на девять равных малых квадратов. Удалим центральный квадрат, а с оставшимися восьмью повторим ту же процедуру. Продолжая этот процесс неограниченно, получим салфетку Серпинского (*рис.* 1). Чтобы построить геометрический пространственный фрактал — губку Мюнгера, в кубе разделим на три равные части два соседних ребра и через полученные точки деления проведем плоскости, соответственно, параллельные боковой грани и основанию куба. Эти плоскости "разрежут" куб на 27 равных кубов. Удалим центральный куб, лежащий в средней полосе, и с каждым из оставшихся кубов проделаем такие же построения, что и с исходным. Повторяя деления и удаления, замечаешь, что "полых" кубиков будет все больше и больше. А их размеры — все меньше и меньше. При неограниченном построении куб превращается в губку с бесконечным количеством пустот (*рис.* 2).

Нетрудно установить ход создания еще одной пары фрактальных собратьев — салфетки и пирамиды Серпинского. В первом случае на

[1] *Азевич А.И.* Симфония фракталов. / Газета "Информатика" № 23–24/2009.

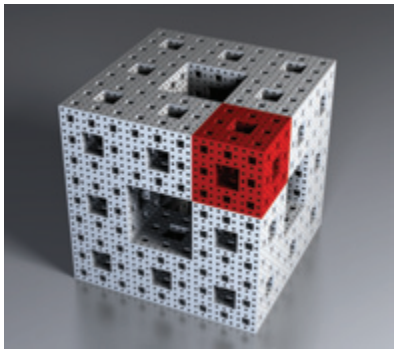


Рис. 2

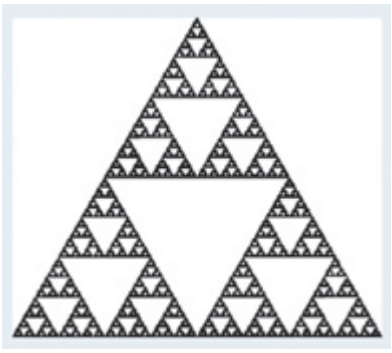


Рис. 3

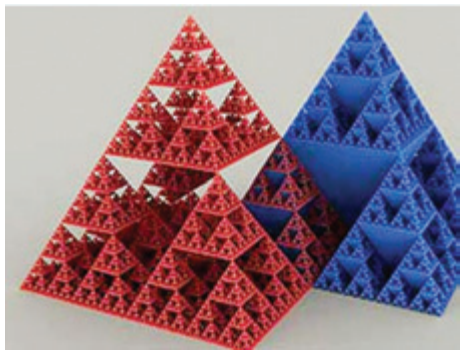


Рис. 4

начальном шаге разбиваем правильный треугольник на четыре равных равносторонних треугольника и удаляем центральный треугольник. На втором шаге повторяем аналогичные построения для оставшихся трех треугольников. Продолжая этот процесс бесконечно, получим "дырявую" салфетку Серпинского (*рис.* 3). Во втором случае, чтобы получить пирамиду Серпинского, в качестве исходного тела берут правильный тетраэдр. Затем в каждой его грани проводят средние линии. Эти отрезки — ребра правильного октаэдра, а его вершины — середины ребер тетраэдра. Удалив вписанный октаэдр, проделаем с оставшимися четырьмя правильными пирамидами аналогичные действия. В ходе бесконечного построения у тетраэдра будет все больше и больше пустот. А сама фигура обретет новую форму, называемую пирамидой Серпинского (*рис.* 4).

Эти фигуры — простейшие геометрические фракталы, хотя слово "простейший" вряд ли к ним подходит, ведь построения выполняются бесконечно. Описанный выше алгоритм позволяет "вручную" с точностью до определенного шага построить один из перечисленных геометрических фракталов.

Уяснив особенности формирования структуры фракталов, отвлечемся от строгой геометрии и точных алгоритмов. Окунемся в бесконечные вариации форм, которые генерируют компьютерные программы. Начнем с галереи фрактальных 3D-картин, коих в Интернете превеликое множество. На сайте http://fractals.nsu.ru/gallery.htm встречаем "старых знакомых" — губку Мюнгера и пирамиду Серпинского. С ними соседствуют другие фракталы: ель, сосна, крест и даже противотанковый еж.

А как построить такие фракталы? Например, какое-нибудь дерево? В идеале оно тоже является фракталом. В дереве отчетливо прослеживаются самоподобные пространственные формы. Сконструировать древовидный фрактал поможет компьютерная программа *Easy Tree Generation*.

Скачаем программу, проинсталлируем и займемся построением дерева. Интерфейс программы весьма прост. Большую часть главного окна занимает поле для рисования. Справа — панель с многочисленными инструментами. Начнем с построения дерева без листьев. Нам надо понять структуру фрактала. Листья отвлекают, мешая сосредоточиться на главном. Чтобы на дереве не было кроны, достаточно в нижней части панели инструментов снять крестик в чекбоксе Leaves (листья). После того как "зеленая шапка" покинет своего хозяина, дерево будет почти готово к осмотру (*рис.* 5 на с. 16). Правда, оно неудобно повернуто к зрителю: не видно фрактального ветвления. Это нетрудно исправить: достаточно воспользоваться командой Twist. Перемещая синий шарик-движок вправо-влево, подберем наиболее подходящее положение дерева в пространстве. Его можно вращать по и против часовой стрелки и увеличивать длину ствола. Кстати, чтобы дерево выглядело реалистично, стоит воспользоваться операцией Balance. Вновь двигая очередной синий шарик, растение превращается то в длинный вытянутый ствол, то в "оленьи рога". Отрегулировав необходимый баланс компьютерной картины, перейдем к изучению других функций. Под рисунком есть еще несколько кнопок. Первая позволяет покрывать дерево листьями, вторая окрашивает ствол в разные цвета, третья задает размеры листьев (*рис.* 6). Изучив все функции программы, можно легко создавать фрактальные деревья самого различного вида: с кроной и без, с падающей тенью, в виде каркасной модели. А двигая колесико мыши в ту или другую сторону, дерево можно приближать или удалять, предлагая наблюдателю во всей красе насладиться результатами компьютерного моделирования.

Если *Easy Tree Generation* служит для генерации деревьев, то программа *Apophysis 2.02* создает совершенно невероятные формы, напоминающие космические галактики. Рассмотрим главное окно программы, пытаясь разобраться в ее многочисленных функциях. Слева виден целый ряд названий уже сгенерированных фракталов (*рис.* 7 на с. 17). Выбираем один из них и нажимаем кнопку F2 на клавиатуре компьютера. Тут же появляется заявленный фрактал. Список тем довольно обширен. Каких только фрактальных рисунков здесь нет: причудливые изгибы разнообразных кривых, замысловатые туманности, воздушные пространственные слои. В компьютерной галерее один шедевр соревнуется с другим. Цветовую гамму "космического рисунка" можно менять по собственному усмотрению. Кроме того, с фрактальными объектами несложно производить всевозможные мутации. Для этого на панели задач надо зайти во вкладку Flame. Затем перейти к функции Mutation. Откроется новое окно, в котором выберем

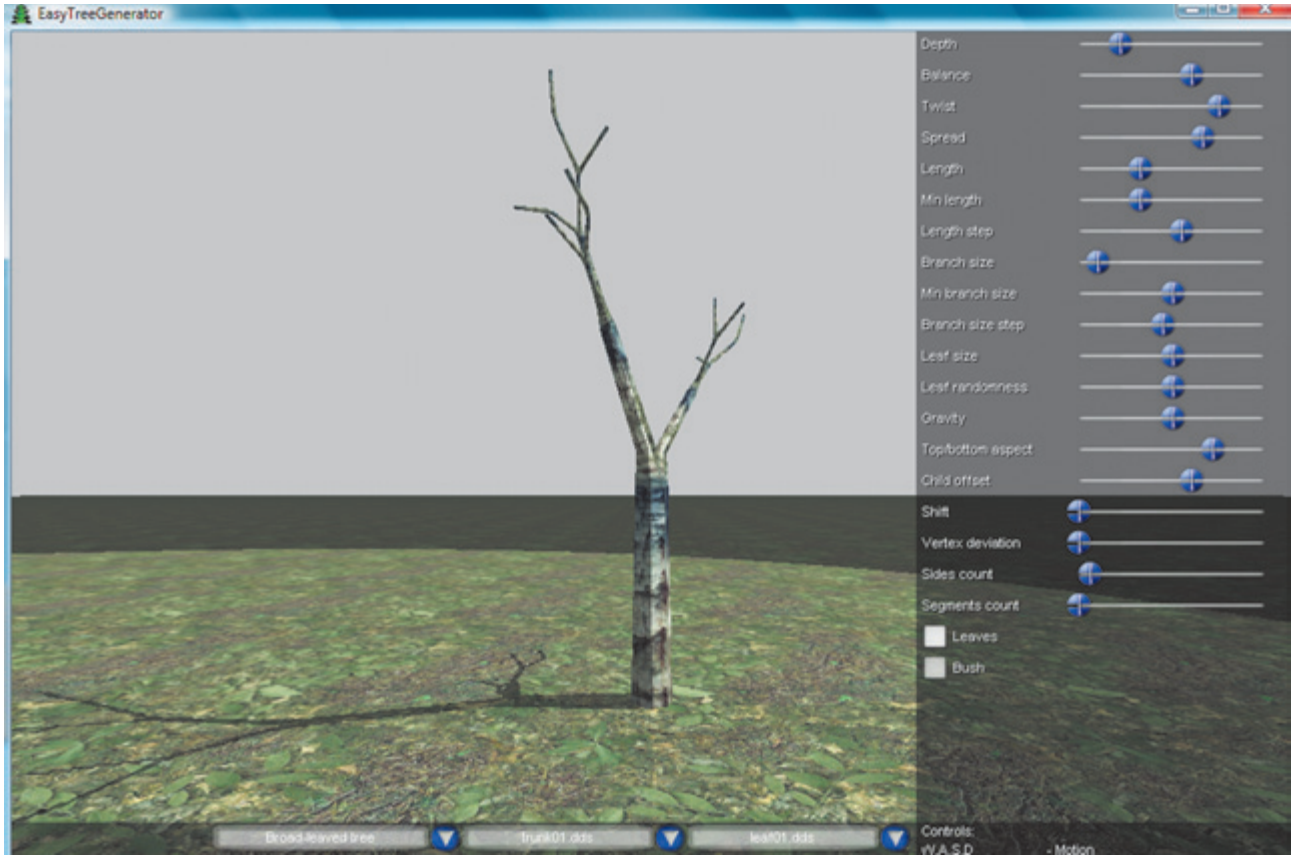


Рис. 5

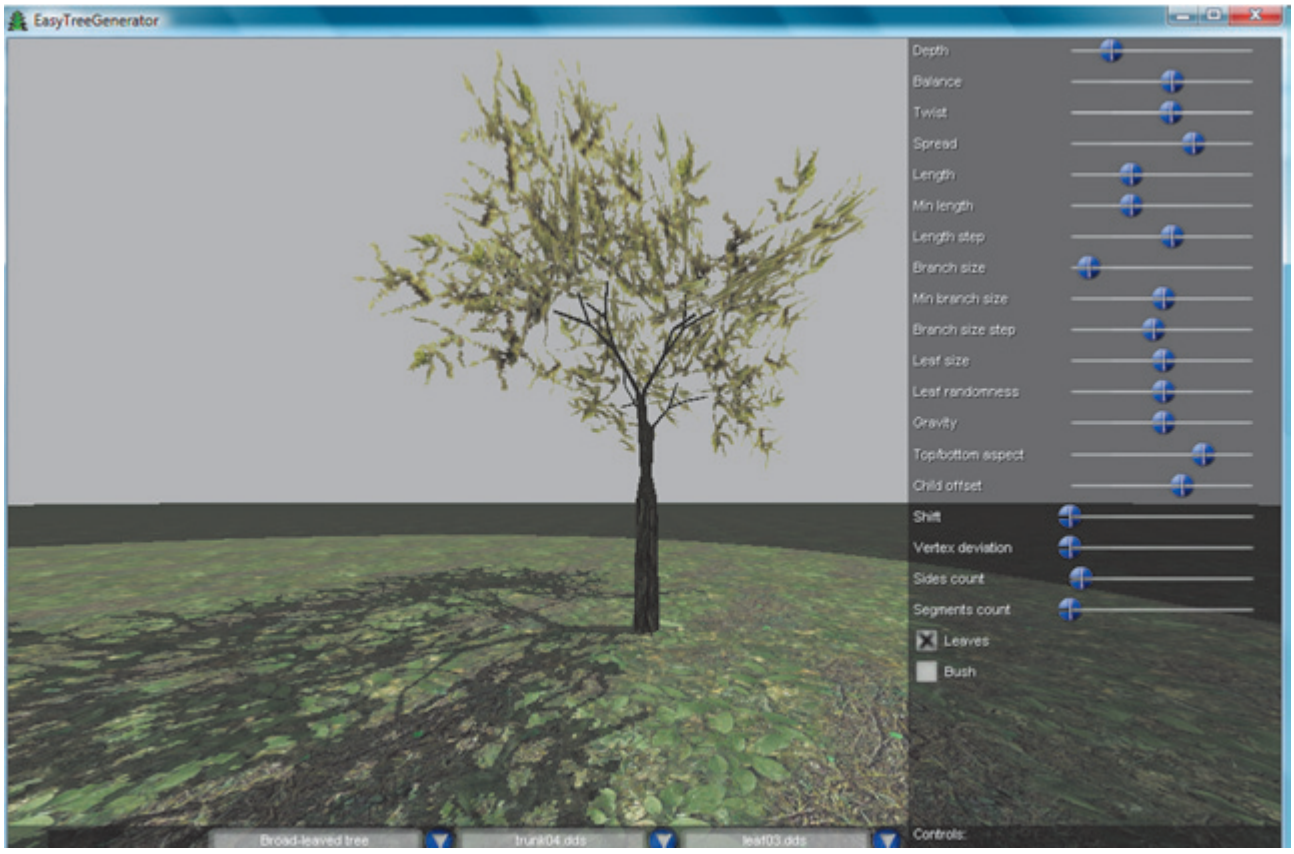


Рис. 6

понравившуюся конфигурацию (*рис.* 8). Это сделать непросто: космических рисунков немало. Если хочется чего-то необычного, программа не заставит себя долго ждать. Например, требуется создать собственный фрактал. Что ж, вновь войдем в Flame. Откроем редактор Editor (*рис.* 9). Он выполнит любые трансформации с формой, цветом и положением в пространстве. Достаточно изменить соответствую-

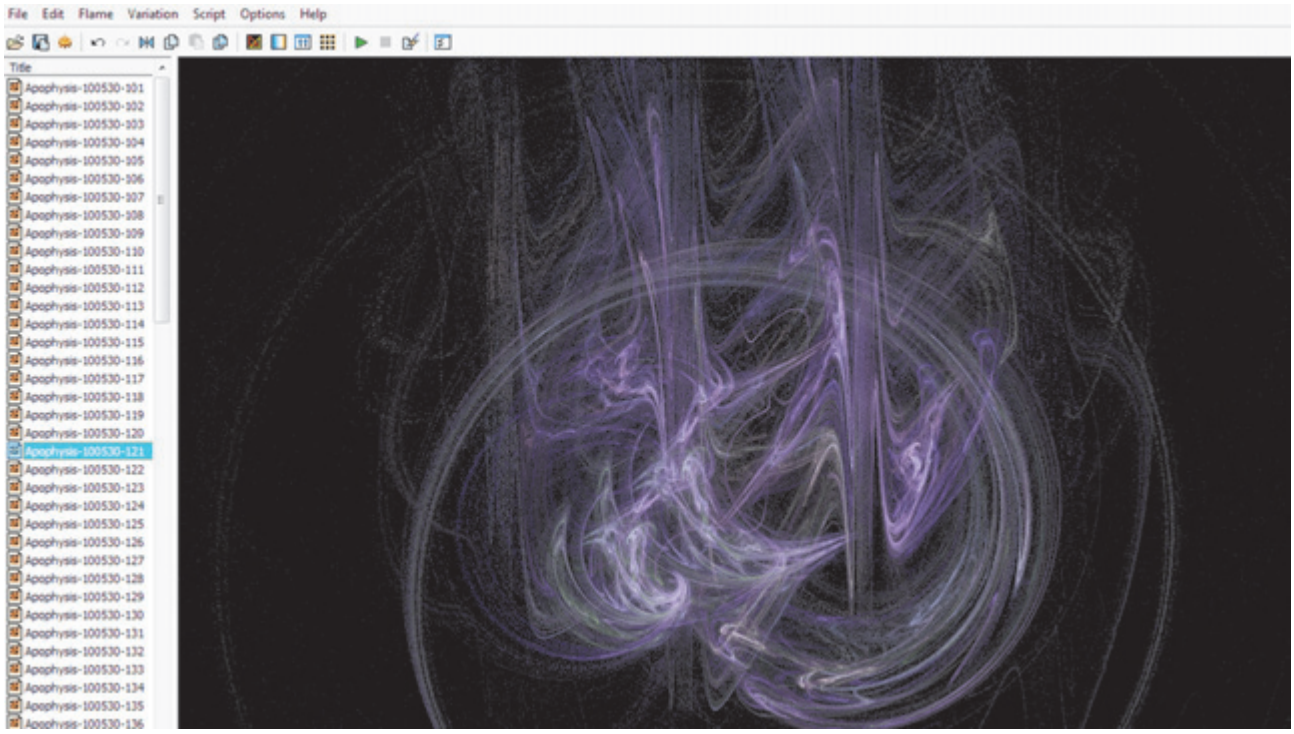


Рис. 7

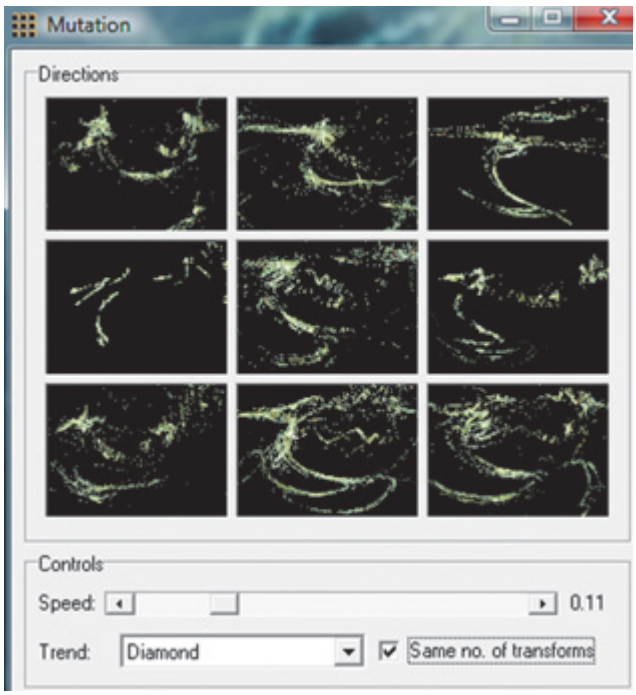


Рис. 8

щие параметры команд и новый фрактал появится на экране компьютера.

Итак, мы рассмотрели простейшие алгоритмы построения фракталов и некоторые программы, генерирующие самоподобные множества. А теперь обратимся к истокам фрактальной математики — множеству Мандельброта[2]. Ученые научились строить его пространственный аналог. Для этого они разработали сложные компьютерные программы. Чтобы узнать, как плоские формы преобразуются в пространственные, достаточно на одном из поисковых серверов Интернета набрать фразу “2D fractal to 3D fractal”. А затем перейти к первой ссылке из огромного числа найденных адресов. На странице по адресу http://www.skytopia.com/project/fractal/mandelbulb.html можно найти много интересной информации, посвященной 3D-фракталам.

Одним из первых, кто представил классическое фрактальное множество Мандельброта в виде пространственного тела, был американский математик, программист и большой любитель фантастики Руди Ракер. Он создал компьютерную программу, в которой была заложена сферическая система координат, позволяющая создавать в пространстве различные фрактальные множества. Кроме того, система переводит плоские фрактальные множества в 3D-объекты (*рис.* 10). На этом рисунке показан пространственный аналог множества Мандельброта.

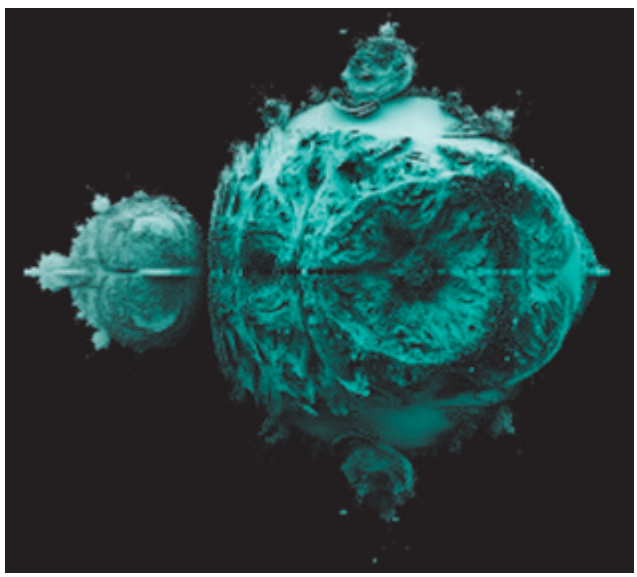


Рис. 9

[2] *Азевич А.И.* Симфония фракталов. / Газета “Информатика” № 23–24/2009.

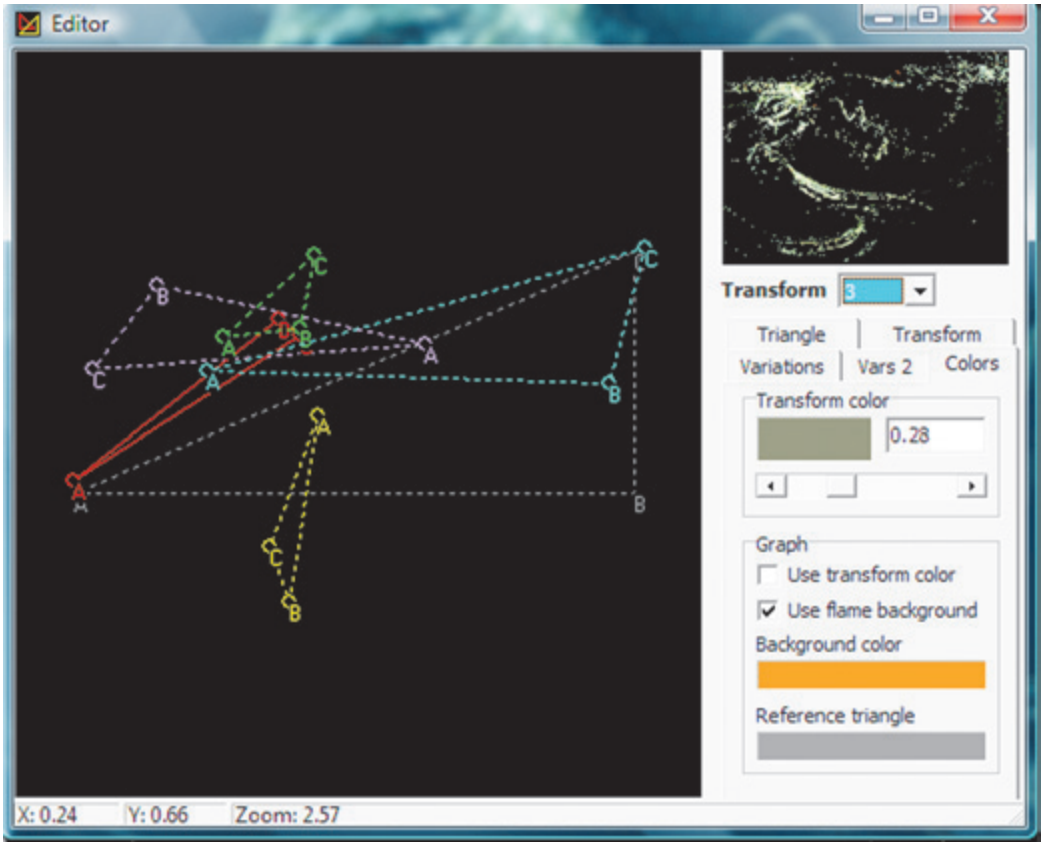


Рис. 10

Пространственные формы получили название фракталов Мандельбульба. Используя гиперкомплексную алгебру и современные языки программирования, ученые создали компьютерные программы, которые генерируют удивительно красивые 3D-фракталы. Один из них можно увидеть на *рис.* 11. Это так называемый "Мандельбульб 8-го порядка". Глядя на него сверху, нетрудно заметить поворотную симметрию 8-го порядка. Если условно провести перпендикулярную прямую через отверстие на части поверхности, напоминающей правильный восьмиугольник, а затем повернуть фрактал на угол $360^0/8$, то он перейдет сам в себя. Вот она, красота самоподобной структуры: многократное повторение, симметрия и совершенство!

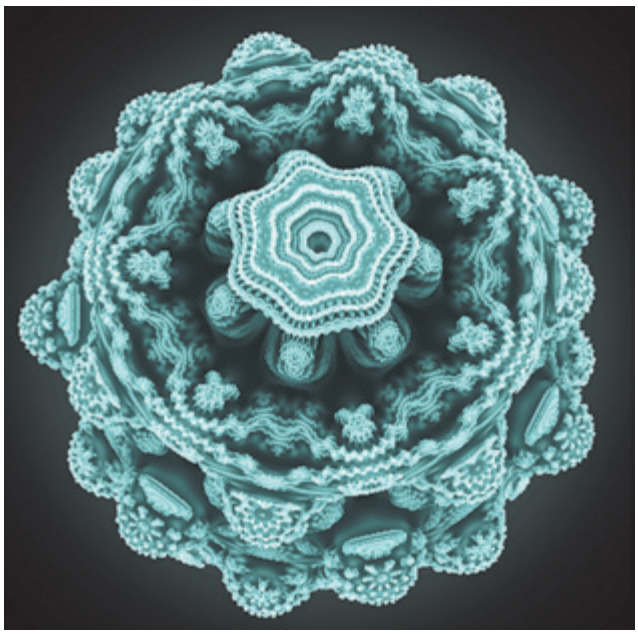


Рис. 11

Еще более таинственными и привлекательными выглядят другие 3D-фракталы из семейства Мандельбульбов. Потрясающее достижение человеческого разума! Форма фракталов настолько естественна и колоритна, будто их создала сама природа. Она, как самобытный и неповторимый художник, потрудилась на славу, предаваясь самым невероятным фантазиям. Так думаешь, рассматривая картины, рожденные современными компьютерами. Сколько увлеченного человеческого труда, глубоких знаний и творческих фантазий несут в себе эти красивейшие формы. Смотрите, удивляйтесь и изучайте! Впереди еще так много интересного!

Источники сети Интернет

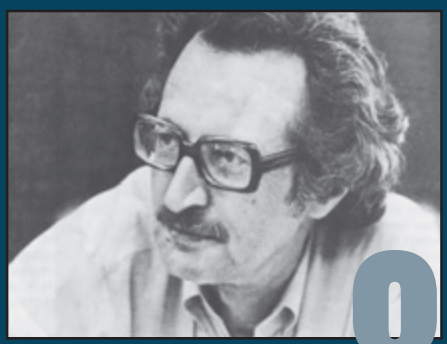
1. http://fractbifur.narod.ru/html/index1.html
2. http://apotut.blogspot.com//
3. http://www.incendia.net/download/index.html
4. http://fractals.narod.ru/descript.htm
5. http://314159.ru/belov/frakyy.htm
6. http://www.skytopia.com/project/fractal/mandelbulb.html
7. http://fraktals.ucoz.ru/publ/0-3
8. http://gorod.tomsk.ru/index-1206035689.php
9. http://fraktalz.narod.ru/programs.html
10. http://ru.wikipedia.org/wiki

XIV

ЧЕТЫРНАДЦАТЫЕ ЮБИЛЕЙНЫЕ СОЛОВЕЙЧИКОВСКИЕ ЧТЕНИЯ

Московский городской дом учителя

1 и 2 октября 2010 года



1 октября – 80 лет

со дня рождения Симона Соловейчика

Сегодняшнее время в школе – жесткое и во многом прагматическое. Общее настроение: не говорите о высоком, не призывайте к идеалам – дайте методику, научите, подскажите, объясните. Этому есть свои причины: к школе предъявляют другие, чем прежде, требования и дети стали другими, и родители – школа не может оставаться в стороне от общего духа времени.

Но тем более важно понимать, что в основании воспитания и образования лежат ценности, которые – непреходящи. И дело школы во все времена свершалось именно тем, насколько эти ценности определяли педагогическое кредо учителя и становились действенной силой школьной жизни. Свобода, идеалы, внутренний мир человека – не отвлеченные понятия, а та духовная действительность, с которой имеет дело учитель. И на Чтениях мы попытаемся понять: как совместить требования времени – и неотменимые законы педагогики, законы становления человека, которые исследовал и о которых писал Симон Соловейчик.

Дорогие учителя!
Приходите на чтения!

Вход, как всегда, свободный.

Открытие чтений – 1 октября в 10 часов утра.
Программа чтений будет опубликована в газете «Первое сентября», а также на сайте www.1september.ru
Справки по телефону: (499) 249-31-38.

Адрес Московского городского дома учителя:
Москва, ул. Пушечная, дом 4, строение 2.
Проезд: центр, станция метро «Кузнецкий мост».

.RU РОБОТЛАНДСКИЙ УНИВЕРСИТЕТ

Роботландский университет (руководитель — Дуванов Александр Александрович) открывает набор студентов на 2010/2011 учебный год.

В университет принимаются коллективные, а на курс “42. Web-конструирование” коллективные и индивидуальные ученики.

Коллективный студент — это группа детей, работающая под руководством одного или нескольких наставников (наставник, как правило, — школьный учитель).

Индивидуальный студент — это учитель, желающий пройти обучение индивидуально, без группы детей.

В конце годичного курса при успешном обучении школьный учитель получает удостоверение от негосударственного образовательного учреждения “Роботландия” и удостоверение государственного образца от Российской академии повышения квалификации и переподготовки работников образования. Школьники получают удостоверение от “Роботландии”.

Занятия в университете платные (цены — на сайте www.botik.ru/~robot). Они начинаются 10 октября текущего года и продолжаются в течение двух семестров до мая следующего года. Для подписчиков “Информатики” предусмотрена 10%-ная скидка за обучение.

Характерные черты Роботландской школы:

1. Совместное обучение учителя и школьников в рамках одной команды.
2. Турнирный цикл обучения.
3. Моделирование коллективной деятельности.
4. Реальная практическая польза детских проектов.
5. Перекрестные проверки работ.
6. Развитые горизонтальные связи.

Образец заявки

ЗАЯВКА НА ОБУЧЕНИЕ В РОБОТЛАНДСКОМ УНИВЕРСИТЕТЕ	
Предполагаемый курс (отделение)	42. Web-конструирование
Форма обучения	Коллективный ученик
Руководитель (обучаемый)	Иванов Петр Николаевич
Профессия	Учитель информатики средней школы
Стаж работы в школе	3 года
Директор школы	Петров Семен Сергеевич
Электронный адрес	ivanov@sch62.nsk.ru
Название учреждения и его почтовый адрес (с индексом)	630090, г. Новосибирск, ул. Жемчужная, 6, средняя школа № 62
Почтовый адрес для посылок (лучше домашний руководителя)	630091, г. Новосибирск, ул. Строителей, д. 45, кв. 124, Иванову Петру Николаевичу
Предполагаемое число детей в группе и их возраст (для коллективного ученика)	20 школьников, 10–11-е классы
Первый раз в Роботландском университете	Да
Руководитель (обучаемый) — подписчик газеты “Информатика”	Да
Откуда получена информация об университете	Газета “Информатика”

Заявки принимаются по адресу: kurs@robotland.pereslavl.ru до 1 октября.

Для получения более подробной информации можно написать по адресу kurs@robotland.pereslavl.ru (администрация университета) или заглянуть на сайт www.botik.ru/~robot (здесь можно отправить заявку, заполнив электронную форму).

КРАТКОЕ ОПИСАНИЕ КУРСОВ

Номер и название курса	Возраст детей	Куратор курса	Описание курса
11. Зимние вечера	3–4-е классы	Первин Юрий Абрамович	Знакомство с алгоритмами и исполнителями, электронная почта; детские конкурсы
12. Азы информатики-I. Знакомимся с компьютером. Работаем с информацией	3–5-е классы	Кацай Ирина Ивановна	Начала информатики для малышей. Коллективная работа на Wiki-сайте и других сервисах Web 2.0
14. Азы информатики-II. Пишем на компьютере	4–6-е классы	Кацай Ирина Ивановна	В рамках обозначенной темы курс связывает пять контентов: познавательный, инструментальный, концептуальный, дизайнерский и творческий. Коллективная работа на Wiki-сайте и других сервисах Web 2.0
15. Азы информатики-III. Рисуем на компьютере	5–7-е классы	Каюшкина Лариса Александровна	Основы информатики на базе конструирования компьютерных изображений. Коллективная работа на Wiki-сайте и других сервисах Web 2.0
16. Азы информатики-IV. Выходим в Интернет	6–8-е классы	Чмелева Елена Николаевна	Компьютерные сети, электронная почта, сетевые технологии. Коллективная работа на Wiki-сайте (и других сервисах Web 2.0)
31. Азы программирования-I. Плюсик и Кукарача	6–8-е классы	Садовая Ирина Владимировна	Для детей, знакомых с понятиями “исполнитель”, “алгоритм”, “программа” и желающих поближе познакомиться с программированием
32. Азы программирования-II. Корректор	7–9-е классы	Садовая Ирина Владимировна	Продолжение курса 31 на базе исполнителя Корректор
42. Web-конструирование	старшие классы	Дуванов Александр Александрович	Создание сайтов на базе HTML+CSS+JavaScript (последнее факультативно). Основы проектирования, web-дизайна и юзабилити. Планируется кружок по теме “Принципы и шаблоны современного web-интерфейса”

ЧТО ДЕЛАТЬ, ЕСЛИ ВЫ НЕ ПОЛУЧИЛИ ГАЗЕТУ ВОВРЕМЯ

Дорогие друзья!

Вот уже много лет номера всех наших газет сдаются в печать точно в срок, своевременно печатаются в типографии и немедленно рассылаются подписчикам. Мы можем ответственно утверждать, что такая практика сохранится и в будущем.

Тем не менее некоторые подписчики иногда не получают газеты вовремя. Мы тщательно разбираем каждый такой случай и всегда обнаруживаем, что сбой происходит в системе доставки. Это действительно редкие происшествия, но мы все равно озабочены ими и хотим помочь каждому подписчику получать газету регулярно.

Если вы не получили тот или иной номер газеты, вам нужно в самое ближайшее время обратиться в ваше почтовое отделение.

Почтовые работники обязаны проверить, почему указанный номер газеты не был вам доставлен. У них есть возможность связаться с центральным отделом подписки, куда стекается вся информация по подписке в регионе, с сортировочным узлом вашего региона, куда приходят газеты из типографии, и выяснить, где и в какой момент произошел сбой в доставке. Конечно, им проще переложить ответственность на редакцию, но вам не стоит верить такому объяснению. Лучше напомните сотрудникам почтового отделения об их возможностях.

Если же результат не достигнут, то вам нужно в письменном виде в двух экземплярах **оформить требование на имя начальника отдела доставки**. В требовании укажите свое имя, адрес, наименование издания с подписным индексом и номер, который вам не был доставлен. Завершите документ просьбой решить возникшую проблему и назначьте разумный срок. Один экземпляр передайте в почтовое отделение, на другом попросите расписаться того работника почты, который принял у вас претензию.

Ваше право обращаться с такими требованиями по своему усмотрению к редакции, издателю или распространителю установлено «Правилами распространения периодических печатных изданий по подписке», которые Правительство Российской Федерации утвердило 1 ноября 2001 года. В п. 17 этих Правил записано, что **«требования подписчика подлежат обязательной регистрации в пункте приема подписки»**. А в п. 25 сказано, что **«не доставленный в срок экземпляр… должен быть доставлен в срок, назначенный подписчиком»**, правда, «с учетом времени получения распространителем периодического печатного издания от редакции, издателя».

Практика показывает, что обычно такой официальный способ снимает все трудности. Сотрудники почтового отделения разбираются в проблеме, находят потерянные номера и доставляют их подписчику. Но если проблема с получением газеты не решена и после подачи требования, сразу сообщите об этом нам, и мы постараемся помочь. Если ваш запрос поступит своевременно, то мы наверняка сможем решить проблему.

Поверьте, для нас очень важно, чтобы наша газета служила помощником в вашем нелегком труде. Мы со своей стороны делаем все, чтобы газета оправдывала ваши ожидания, была интересным и полезным подспорьем в работе и в жизни.

Отдел распространения
Издательского дома «Первое сентября»
ул. Киевская, д. 24, Москва, 121165
Тел.: (499) 249-47-58 Факс: (499) 249-31-38
E-mail: podpiska@1september.ru

GAMES.EXE

Пасьянс “Турецкий платок”

Д.М. Златопольский

В этой статье мы опишем методику разработки программы, моделирующей раскладывание старинного пасьянса “Турецкий платок”. Правила такие. Из одной колоды карт в 52 листа выкладываются картинкой вверх пять рядов по 10 карт в каждом. Последние две карты кладут в шестой неполный ряд, как правило, к первому и второму столбцам (*рис.* 1).

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
3♠	В♦	5♣	Х♠	6♠	7♣	Т♠	Д♣	Д♠	Х♦
2♥	9♣	К♦	В♠	2♦	Т♣	К♠	8♥	4♣	8♣
5♠	В♥	5♦	8♠	5♥	7♥	В♣	7♦	6♦	6♥
2♠	4♠	Х♣	Д♦	Т♥	8♦	7♠	3♣	4♦	9♥
К♣	4♥	К♥	9♦	Т♦	Х♥	3♥	3♦	2♣	6♣
9♠	Д♥								

Рис. 1

Требуется “распустить” этот турецкий платок, снимая за один ход по две нижние карты одного достоинства (тройки, дамы и т.д.) из различных столбцов.

В программе полное название карты будем обозначать двумя символами, первый из которых соответствует достоинству карты (“2” — “двойка”, “3” — “тройка”, ..., “9” — “девятка”, “Х” — “десятка”, “В” — “валет”, “Д” — “дама”, “К” — “король”, “Т” — “туз”), а второй — названию масти (“П” — “пики”, “К” — “крести”, “Б” — “бубны”, “Ч” — “червы”), например, “8Т” — восьмерка треф, “ДБ” — “дама бубен”, “ХП” — “десятка пик”. Полный набор карт (52 шт.) будем хранить в массиве из 52 элементов, значениями которого являются 2-символьные величины. В приведенной ниже программе имя этого массива — *карты*.

Расклад карт (исходный и последующие) будем моделировать в виде двумерного массива с именем *расклад* из 6 строк и 10 столбцов; значения его элементов — элементы массива *карты*.

Общая схема программы, моделирующей рассматриваемый пасьянс, такая:

```
Формирование и вывод на экран исходного
расклада
цикл
  Очередной ход играющего
  если играющий не сдался
    то
     ОчисткаЭкрана
     Изменение расклада и его вывод на экран
  все
конец_цикла в случае, если играющий сдался
            или при снятии всех карт
Объявление результатов
```

— где `ОчисткаЭкрана` — стандартная процедура очистки экрана.

Поскольку в программе понадобится многократный вывод на экран текущего расклада, целесообразно создать соответствующую процедуру. На школьном алгоритмическом языке она выглядит следующим образом:

```
алг Печать_Расклада(арг лит таб
                    расклад[1:6, 1:10])
нач цел i, j
  вывод нс, "ПАСЬЯНС 'Т У Р Е Ц К И Й
                      П Л А Т О К'"
  вывод нс
  нц для i от 1 до 6
    вывод нс
    нц для j от 1 до 10
      вывод расклад[i,j]
    кц
  кц
  |Вывод номеров столбцов
  вывод нс, "1  2  3  4  5  6  7  8  9  10"
кон
```

Формирование исходного расклада может быть проведено в три этапа:

1. Получить одномерный массив из 52 элементов, значениями которого являются неповторяющиеся числа от 1 до 52. Имя этого массива в программе — *набор1*.

2. Получить двумерный массив (с именем *набор2*), в котором первые пять строк и первые два элемента шестой строки представляют собой значения элементов массива *набор1*.

3. Получить двумерный массив *расклад*, в котором первые пять строк и первые два элемента шестой строки представляют собой обозначения карт из массива *карты* (по их номерам в массиве *набор2*).

Этап 1 можно выполнить несколькими способами ([1–2]). Один из них заключается в том, что

1) генерируется случайное число (с помощью функции `rnd`);

2) проверяется, имеется ли уже такое число в заполненной части массива; если не имеется, то число записывалось в массив, в противном случае генерировалось новое случайное число и т.д. до генерации числа, которого еще нет в массиве.

Описанные действия повторяются до заполнения всего массива.

В приведенном ниже фрагменте программы, реализующей этот метод, используются следующие основные величины:

набор1 — заполняемый массив;

случ_число — очередное случайное число, получаемое с помощью функции `rnd`;

всего_в_наборе1 — количество уже заполненных значениями элементов массива *набор1*;

есть_уже — величина логического типа, фиксирующая факт наличия очередного случайного числа в заполненной части массива.

```
|Заполнение массива набор1
набор1[1] := 1 + rnd(52)
всего_в_наборе1 := 1
нц
  |Получаем случайное число
  случ_число := 1 + rnd(52)
  |Проверяем, есть ли уже такое
  i := 1; есть_уже := нет
  нц
    если набор1[i] = случ_число
     то
       есть_уже := да
     иначе
       i := i + 1
    все
  кц_при i > всего_в_наборе1 или есть_уже
  если не есть_уже
   то       |Числа случ_число в массиве нет
    |Записываем его в массив
    всего_в_наборе1 := всего_в_наборе1 + 1
    набор1[всего_в_наборе1] := случ_число
  все
кц_при всего_в_наборе1 = 52
```

На этапе 2 надо переписать значения элементов массива *набор1* в двумерный массив *набор2*. Для этого необходимо установить зависимость индексов элемента двумерного массива (номер строки и номер столбца) от индекса соответствующего элемента одномерного массива. Анализ показывает, что эти зависимости следующие:

номер строки = (*индекс* + 9) div 10;
если *индекс* кратен 10
то *номер столбца* = 10
иначе *номер столбца* = *индекс* mod 10,

где div — операция целочисленного деления, mod — операция определения остатка от деления первого операнда на второй, *индекс* — индекс элемента одномерного массива.

Соответствующий фрагмент программы:

```
|Заполнение массива набор2
нц для i от 1 до 52
  номер_строки := div(i + 9, 10)
  если mod(i,10) <> 0
   то
    номер_столбца := mod(i,10)
   иначе
    номер_столбца := 10
  все
  набор2[номер_строки,
         номер_столбца] := набор1[i]
кц
```

После этого этап 3 (формирование расклада карт) может быть выполнен так:

```
|Заполняем массив расклад
|Первые 5 строк
нц для i от 1 до 5
  нц для j от 1 до 10
    расклад[i, j] := карты[набор2[i, j]]
  кц
кц
|Первые два элемента 6-й строки
нц для j от 1 до 2
  расклад[6, j] := карты[набор2[6, j]]
кц
|Остальные элементы 6-й строки "пустые"
нц для j от 3 до 10
  расклад[6, j] := "  "
кц
```

Очередной ход играющего будем моделировать вводом им двух чисел, обозначающих номера столбцов, с которых снимаются карты. Эти номера обозначим *выбор1* и *выбор2*. Для контроля правильности введенных значений нам понадобится информация о:

— количестве оставшихся карт в каждом столбце (ее будем в программе хранить в массиве *осталось* из десяти элементов);

— обозначении выбранных карт (имена этих величин — *карта1* и *карта2*);

— достоинстве этих двух карт (имена этих величин — *достоинство1* и *достоинство2*).

Заполнение массива *осталось* происходит следующим образом:

```
|Первые два столбца
осталось[1] := 6; осталось[2] := 6
 |Остальные столбцы
нц для i от 3 до 10
  осталось[i] := 5
кц
```

Фрагмент программы, относящийся к вводу значения *выбор1* и проверке его правильности, имеет вид:

```
нц
  вывод нс, "Укажите номер одного из столбцов"
  ввод выбор1
  если выбор1 > 10 или выбор1 < 0
   то
    вывод нс, "Неправильный номер!"
    Пауза
    ОчисткаЭкрана
    Печать_Расклада(расклад)
    иначе
    |Проверяем наличие карт в выбранном
    |столбце
    если выбор1 > 0   |причем делаем это
            |только если играющий не сдался
     то
      если осталось[выбор1] = 0
       то
        вывод нс, "В", выбор1, "-м столбце
                   нет карт!"
        Пауза
        ОчисткаЭкрана
        Печать_Расклада(расклад)
      все
    все
  все
кц_при выбор1 = 0 или выбор1 > 0
и выбор1 < 11 и осталось[выбор1] > 0
|Обратите внимание на условие окончания
|цикла
```

Применительно к вводу значения *выбор2* аналогичный фрагмент:

```
если выбор1 <> 0 |Если играющий не сдался
 то
  |Цикл ввода значения выбор2
  нц
    вывод нс, "Укажите номер второго
               из столбцов"
    ввод выбор2
    если выбор2 > 10 или выбор2 < 0
     то
      вывод нс, "Неправильный номер!"
      Пауза
      ОчисткаЭкрана
      Печать_Расклада(расклад)
     иначе
      |Проверяем, не равны ли номера
      если выбор1 = выбор2
        то
         вывод нс, "Номера столбцов
                    не должны быть
                    одинаковыми!"
         Пауза
         ОчисткаЭкрана
         Печать_Расклада(расклад)
        иначе
         |Проверяем наличие карт
         |в выбранном столбце
         если осталось[выбор2] = 0
          то
           вывод нс, "В", выбор2,
                     "-м столбце нет карт!"
           Пауза
           ОчисткаЭкрана
           Печать_Расклада(расклад)
        все
      все
    все
  кц_при выбор2 = 0 или выбор2 > 0
       и выбор2 < 11 и осталось[выбор2] > 0
         и выбор1 <> выбор2
все
```

Примечание. Пауза — стандартная процедура (при наличии ее в языке программирования), обеспечивающая некоторую паузу в выполнении программы. Вместо нее можно использовать также так называемый "пустой цикл" [3].

Если введенные значения номеров столбцов *выбор1* и *выбор2* правильные, необходимо еще проверить, одинакового ли достоинства нижние карты в выбранных столбцах. Делается это так[1]:

```
|Определяем, не сдался ли играющий
если выбор1 <> 0 и выбор2 <> 0
 то
  |Определяем достоинства нижних карт
  |в указанных столбцах
  |Карта в первом выбранном столбце
  карта1 := расклад[осталось[выбор1],
                    выбор1]
  |Карта во втором выбранном столбце
  карта2 := расклад[осталось[выбор2],
                    выбор2]
  |Достоинства карт
  достоинство1 := карта1[1]
  достоинство2 := карта2[1]
  |И сравниваем их
  если достоинство1 <> достоинство2
   то
    вывод нс, "Достоинства выбранных
               карт разные!"
    Пауза
    ОчисткаЭкрана
    Печать_Расклада(расклад)
  все
все
```

В результате мы рассмотрели процесс ввода играющим значений *выбор1* и *выбор2* (на общей схеме программы, приведенной в начале, этот этап называется Очередной ход играющего). Условие окончания цикла ввода этих значений: *выбор1* = 0 или *выбор2* = 0 или *достоинство1* = *достоинство2*.

И только после окончания этого цикла можно убрать нижнюю карту в выбранных столбцах (если играющий не сдался):

```
если выбор1 <> 0 и выбор2 <> 0
 то
  |Убираем соответствующие карты
  расклад[осталось[выбор1], выбор1] := "  "
  расклад[осталось[выбор2], выбор2] := "  "
  |Изменяем значения осталось[выбор1]
  |и осталось[выбор2],
  осталось[выбор1] := осталось[выбор1] — 1
  осталось[выбор2] := осталось[выбор2] — 1
  |а также величину всего_осталось_карт
  всего_осталось_карт := всего_осталось_
                         карт — 2
```

[1] Можно объединить приведенный фрагмент с фрагментами проверки правильности ввода значений номеров столбцов *выбор1* и *выбор* 2, однако это усложнит программу и затруднит ее чтение.

```
    |Печатаем новый расклад
    ОчисткаЭкрана
    Печать_Расклада(расклад)
  все
```

— где *всего_осталось_карт* — общее количество оставшихся карт (по значению этой величины можно зафиксировать факт раскладки пасьянса).

Итак, вся программа, моделирующая раскладывание пасьянса “Турецкий платок”, имеет вид:

```
|Набор всех карт
лит таб карты[1:52]
карты[1]  := "2П"; карты[2]  := "3П"; …
карты[14] := "2Т"; карты[15] := "3Т"; …
карты[27] := "2Б"; карты[28] := "3Б"; …
карты[40] := "2Ч"; карты[41] := "3Ч"; …
карты[51] := "КЧ"; карты[52] := "ТЧ"
алг Пасьянс_Турецкий_платок
нач … (описание всех используемых величин)
  |Заполнение массива набор1
  … (см. выше)
  |Заполнение массива набор2
  … (см. выше)
  |Заполнение массива расклад
  … (см. выше)
  |Заполнение массива осталось
  … (см. выше)
  |Вывод исходного расклада
  Печать_Расклада(расклад)
  всего_осталось_карт := 52
  нц
    нц
      нц
        |Ввод значения выбор1
        вывод нс, "Укажите номер одного
                   из столбцов"
        ввод выбор1
        … (см. выше)
      кц_при выбор1 = 0 или выбор1 > 0
      и выбор1 < 11 и осталось[выбор1] <> 0
      если выбор1 <> 0
       то
        |Ввод значения выбор2
        нц
          вывод нс, "Укажите номер второго
                     из столбцов"
          ввод выбор2
          … (см. выше)
        кц_при выбор2 = 0 или выбор2 < 11
                  и осталось[выбор2] <> 0
                  и выбор1 <> выбор2
      все
      |Проверяем, одинакового ли достоинства
      |нижние карты в выбранных столбцах
      |Определяем, не сдался ли играющий
      если выбор1 <> 0 и выбор2 <> 0
       то
        |Определяем достоинства нижних карт
        |указанных столбцов
        … (см. выше)
      все
    кц_при выбор1 = 0 или выбор2 = 0 или
                достоинство1 = достоинство2
    если выбор1 <> 0 и выбор2 <> 0
     то
      |Убираем соответствующие карты
      расклад[осталось[выбор1], выбор1] := "  "
      … (см. выше)
      |Печатаем новый расклад
      ОчисткаЭкрана
      Печать_Расклада(расклад)
    все
  кц_при всего_осталось_карт = 0
              или выбор1 = 0 или выбор2 = 0
  |Определение результата
  если всего_осталось_карт = 0
   то
    вывод нс, "Поздравляю! Вы выиграли!"
   иначе   |во всех других случаях
    вывод нс, "Увы! Вы проиграли!"
  все
кон
```

Литература

1. Случайные числа в программах. / “В мир информатики” № 33 (“Информатика” № 34/2004).

2. Заполнение массива неповторяющимися случайными числами. / “В мир информатики” № 42 (“Информатика” № 43/2004).

3. Школа программирования: тематический выпуск газеты-вкладки “В мир информатики”. / “Информатика” № 11/2005.

Задание для самостоятельной работы

Разработайте описанную программу на известном вам языке программирования и пришлите ее в редакцию. Все приславшие правильную программу будут награждены дипломами. Срок представления программы не ограничен.

ЗАДАЧНИК

Ответы, решения, разъяснения

к заданиям, опубликованным в газете “В мир информатики” № 139 (“Информатика” № 4/2010)

1. Задача “Культурные развлечения”

Напомним, что необходимо было определить, на каком виде культурных развлечений были молодые люди, если известно, что:

1) Андрей отправился на концерт;

2) Борис провел все время с Ольгой;

3) Сергей так и не увиделся с Ритой;

4) Полина побывала в кино;

5) Рита посмотрела спектакль в театре.

Кроме тех, кто уже назван, постоянными членами компании были Дима и Света. Вместе с каждым из юношей на том же виде культурных развлечений побывала одна девушка. Известно также, что одна из пар посетила художественную выставку.

Решение

Составим таблицу и запишем в нее известную информацию:

	Ольга	Рита	Полина	Света	
Андрей	–				*Концерт*
Борис	+	–	–	–	
Сергей	–	–			
Дима	–				
		Театр	*Кино*		

Точно зная "состав" одной из пар, поставим знак "–" в других клетках соответствующей строки и столбца.

С Андреем на концерте могла быть Ольга или Света (Полина и Рита по условию были в кино и театре). Ольга не могла быть с Андреем, т.к. все время провела с Борисом. Значит, с Андреем вечер провела Света:

	Ольга	Рита	Полина	Света	
Андрей	–	–	–	+	*Концерт*
Борис	+	–	–	–	
Сергей	–	–		–	
Дима	–			–	
		Театр	*Кино*	*Концерт*	

Из последней таблицы видно, что Ольга была на художественной выставке вместе с Борисом, а Рита была в театре с Димой. Полина была в кино с Сергеем.

Правильные ответы представили:

— Ан Роман и Кузнецов Василий, Москва, гимназия № 1530, учитель **Шамшев М.В.**;

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

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

— Бартецкий Павел и Савинов Александр, Московская обл., г. Краснознаменск, школа № 4 им. Г.К. Жукова, учитель **Федорова Л.А.**;

— Богданов Дмитрий, Вяткина Марина, Емелина Анастасия, Майорова Екатерина, Новикова Анна и Одинцова Екатерина, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Дубинин Федор, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Ерош Татьяна, Климентьев Владимир, Кузьмин Евгений, Мельник Кристина и Ушакова Мария, г. Лесосибирск Красноярского края, поселок Стрелка, школа № 8 им. Константина Филиппова, учитель **Лопатин М.А.**;

— Есипов Антон, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Желтышева Елизавета, средняя школа поселка Суксун, Пермский край, учитель **Желтышева Л.А.**;

— Жуков Григорий, Евтушенко Илья, Корхов Александр, Лубенцов Дмитрий и Чередниченко Александр, Москва, Центр образования № 1406 (школа для детей с нарушениями слуха), учитель **Миронова А.А.**;

— Карасева Анастасия и Пучкина Елена, Республика Башкортостан, г. Стерлитамак, гимназия № 5, учитель **Пучкина С.А.**;

— Михайловский Даниил, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

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

— Слимов Никита и Чарикова Александра, Москва, школа № 1399, учитель **Шнейдер Е.В.**;

— Тамошина Наталья, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Яковенко Василий, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**

2. Задача "Сколько этажей в доме?"

Напомним, что необходимо было определить, сколько этажей в доме, в котором живут Наташа и Дима, если Наташа живет на пятом этаже, в квартире номер 83, а Дима — на третьем этаже, в квартире номер 169. Во всех подъездах число этажей одинаковое, а на каждой лестничной клетке — 4 квартиры.

Решение

Можно определить, сколько "полных" этажей (лестничных клеток) имеется до этажа, на котором живет Наташа. Это число равно целочисленному частному от деления 83 на 4, т.е. 20. Из них в ее подъезде — 4 этажа (так как Наташа живет на пятом этаже), а до ее подъезда — 20 – 4 = 16 этажей. Эти 16 этажей могут находиться в двух подъездах по 8 этажей в каждом или в одном подъезде из 16 этажей.

Аналогичные рассуждения для квартиры Димы приводят к выводу, что до его подъезда имеются 40 этажей. Они могут находиться в восьми подъездах по 5 этажей в каждом, в пяти подъездах по 8 в каждом, в четырех подъездах по 10 в каждом, в двух подъездах по 20 в каждом, или в одном подъезде из 40 этажей (если допустить, что такое возможно).

Сравнивая возможные варианты (с учетом того, что Наташа и Дима живут в одном доме), можно установить, что допустимым является только вариант из 8 этажей в каждом подъезде.

Правильные ответы прислали:

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

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

— Бартецкий Павел и Савинов Александр, Московская обл., г. Краснознаменск, школа № 4 им. Г.К. Жукова, учитель **Федорова Л.А.**;

— Дубинская Елена и Козырева Ольга, Москва, гимназия № 1530, учитель **Шамшев М.В.**;

— Емелина Анастасия и Новикова Анна, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Ерош Татьяна, Климентьев Владимир, Кузьмин Евгений, Мельник Кристина и Ушакова Мария, г. Лесосибирск Красноярского края, поселок Стрелка, школа № 8 им. Константина Филиппова, учитель **Лопатин М.А.**;

— Желтышева Елизавета, средняя школа поселка Суксун, Пермский край, учитель **Желтышева Л.А.**;

— Жуков Григорий, Евтушенко Илья, Корхов Александр, Лубенцов Дмитрий и Чередниченко Александр,

Москва, Центр образования № 1406 (школа для детей с нарушениями слуха), учитель **Миронова А.А.**;

— Карасева Анастасия и Пучкина Елена, Республика Башкортостан, г. Стерлитамак, гимназия № 5, учитель **Пучкина С.А.**;

— Михайловский Даниил, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Мнацаканян Ашот, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

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

— Слимов Никита и Чарикова Александра, Москва, школа № 1399, учитель **Шнейдер Е.В.**

3. Статья "Еще софизм про рубли и копейки"

Напомним суть софизма: 1 руб. = 100 коп. = 10 коп. × × 10 коп. = 0,1 руб. × 0,1 руб. = 0,01 руб. = 1 коп., т.е. получаем: 1 руб. = 1 коп. (?!).

Объяснение такому странному результату состоит в ошибочности записи: 100 коп. = 10 коп. × 10 коп.

Ответы представили:

— Ан Роман и Кузнецов Василий, Москва, гимназия № 1530, учитель **Шамшев М.В.**;

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

— Асмандиярова Полина и Романова Валерия, г. Смоленск, школа № 29, учитель **Родикова Р.Д.**;

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

— Вяткина Марина, Новикова Анна и Одинцова Екатерина, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

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

— Жуков Григорий, Евтушенко Илья, Корхов Александр, Лубенцов Дмитрий и Чередниченко Александр, Москва, Центр образования № 1406 (школа для детей с нарушениями слуха), учитель **Миронова А.А.**;

— Ерош Татьяна, Климентьев Владимир, Кузьмин Евгений, Мельник Кристина и Ушакова Мария, г. Лесосибирск Красноярского края, поселок Стрелка, школа № 8 им. Константина Филиппова, учитель **Лопатин М.А.**;

— Леопольд Елена, рабочий поселок Яя Кемеровской обл., школа № 2, учитель **Щербакова О.Б.**;

— Михайловский Даниил, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Мнацаканян Ашот, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

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

— Слимов Никита и Чарикова Александра, Москва, школа № 1399, учитель **Шнейдер Е.В.**;

— Тамошина Наталья, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**

4. Задача "Два раба и два хозяина" (новый вариант)

Напомним условие. Когда-то в древности к реке подошли два хозяина со своими рабами. Они должны были переправиться на другой берег в двухместной лодке. При этом хозяева могли чувствовать себя в безопасности только тогда, когда они находятся вдвоем. Могла ли быть устроена безопасная переправа при этих условиях?

Ответ — переправа могла быть устроена следующим образом:

1) на другой берег переправляются два раба;
2) один из рабов (любой) возвращается;
3) на другой берег переправляются оба хозяина;
4) раб с другого берега возвращается;
5) оба раба переправляются на другой берег.

Правильные ответы прислали:

— Ан Роман и Кузнецов Василий, Москва, гимназия № 1530, учитель **Шамшев М.В.**;

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

— Асмандиярова Полина и Романова Валерия, г. Смоленск, школа № 29, учитель **Родикова Р.Д.**;

— Аствацатурян Эдуард, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

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

— Богданов Дмитрий, Емелина Анастасия, Новикова Анна и Черных Дарья, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Жуков Григорий, Евтушенко Илья, Корхов Александр, Лубенцов Дмитрий и Чередниченко Александр, Москва, Центр образования № 1406 (школа для детей с нарушениями слуха), учитель **Миронова А.А.**;

— Михайловский Даниил, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

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

— Слимов Никита и Чарикова Александра, Москва, школа № 1399, учитель **Шнейдер Е.В.**;

— Тамошина Наталья, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Яковенко Василий, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**

5. Фокус "Отгадывание задуманного дня недели"

Напомним, что в статье был описан фокус, в котором после вычислений по объявленному результату определяется загаданный день недели, и было предложено привести обоснование "секрета" фокуса с помощью формул и(или) разработать компьютерную программу, демонстрирующую описанный фокус.

Компьютерную программу разработали:

— Анисимов Павел, Гафуров Зафар и Слимов Никита, Москва, школа № 1399, учитель **Шнейдер Е.В.**;

— Баженов Виктор, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

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

— Шагалов Антон, гимназия № 1530, учитель **Шамшев М.В.**

"Математическое" обоснование фокуса привели:

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

— Асмандиярова Полина, г. Смоленск, школа № 29, учитель **Родикова Р.Д.**;

— Деминцев Борис, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Михайловский Даниил, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

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

— Шагалов Антон, гимназия № 1530, учитель **Шамшев М.В.**

Правильное решение задачи "Послание будущим издателям" и числового ребуса "НИКЕЛЬ — ЕЛЬНИК" представили также Белослудцев Роман, Бергс Петр, Бывальцева Анастасия, Волков Степан, Ворончихин Роман, Генчик Анастасия, Дерендяева Вера, Жигалова Ксения, Касаткина Мария, Пинаева Дарья, Тронин Антон и Шулепова Алена, Кезская средняя школа № 1, Удмуртская Республика, учитель **Ветошкина Н.В.**, а правильное решение задачи "Лишний рубль" и головоломки "Переставить две спички" — Демидов Павел, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н, учитель **Краснёнкова Л.А.**

Задача о кроликах

Надеемся, что читателям известна последовательность чисел: 1, 1, 2, 3, 5, ..., очередными членами которой являются числа 8, 13, ... (каждый член последовательности, начиная с третьего, равен сумме двух предыдущих). Эту последовательность называют "последовательностью Фибоначчи", поскольку впервые она была представлена в книге "Liber abaci" ("Книга абака"), написанной в XIII веке итальянским математиком Фибоначчи, известным также как Леонардо Пизанский (Leonardo Pisano). В этой книге имеется такая задача: "Некто поместил пару кроликов в загоне, огороженном со всех сторон, дабы узнать, сколько пар кроликов родится в течение года. Природа кроликов такова, что через месяц пара кроликов производит на свет другую пару, а потомство дают они со второго месяца после своего рождения". Так сколько же пар кроликов будет через год?

Приведем решение Фибоначчи. В начале первого месяца была одна пара кроликов; в начале второго — две, причем одна из них зрелая, т.е. способная через месяц принести потомство, вторая — нет. Поэтому в начале третьего месяца будут три пары кроликов, две из них зрелые. В начале четвертого месяца станет пять пар (три пары были и две — новое потомство), из них только три зрелые, и т.д.

Эти рассуждения можно оформить в виде таблицы:

Номер месяца (его начало)	1	2	3	4	5	...
Общее число пар	1	2	3	5	8	
Число зрелых пар	1	1	2	3	5	
В этом месяце родятся, пар	1	1	2	3	5	
В этом месяце станут зрелыми, пар	0	1	1	2	3	

Можно также составить программу, которая на школьном языке программирования выглядит так:

```
алг Задача_о_кроликах
нач цел месяц, КЗрП, КНЗрП, КРП, ВсегоПар
. |1-й месяц
. КЗрП := 1; КНЗрП := 0; КРП := 1
. нц для месяц от 2 до ?
. . |На начало этого месяца:
. . |- общее число зрелых пар
. . КЗрП:=КЗрП+КНЗрП
. . |- общее число пар
. . ВсегоПар  :=  КЗрП + КРП
. . |В этом месяце:
. . |- в этом месяце станут зрелыми
. . |пары, родившиеся в предыдущем месяце
. . КНЗрП := КРП
. . |- родятся новые пары
. . КРП := КЗрП
. кц
. вывод нс, "Общее число пар кроликов "
. вывод "через год будет равно ", ВсегоПар
кон
```

Примечания

1. Смысл использованных переменных величин ясен из их имен и приведенных комментариев.

2. Конечное значение параметра цикла определите самостоятельно.

Задание для самостоятельной работы

Оформите лист электронной таблицы Microsoft Excel и/или составьте компьютерную программу (на изучаемом языке программирования), с помощью которых можно решить указанную задачу. Результаты присылайте в редакцию.

В небольшом городке

В небольшом городке Армении живут пятеро друзей: Аветисян, Петросян, Симонян, Григорян и Карапетян. Профессии у них разные: один из них маляр, другой — мельник, третий — плотник, четвертый — почтальон, а пятый — парикмахер.

Известно, что:

1) Петросян и Григорян никогда не держали в руках малярной кисти;

2) Аветисян и Григорян уже давно собираются посетить мельницу, на которой работает их товарищ;

3) Петросян и Карапетян живут в одном доме с почтальоном;

4) Симонян был недавно в ЗАГСе одним из свидетелей, когда Петросян и дочь парикмахера сочетались законным браком;

5) Аветисян и Петросян каждое воскресенье играют в нарды с плотником и маляром;

6) Григорян и Карапетян по субботам обязательно встречаются в парикмахерской, где работает их друг, а почтальон предпочитает бриться сам.

Кто есть кто?

Иностранные языки

Вадим, Сергей и Михаил изучают разные иностранные языки — китайский, японский и арабский. На вопрос, какие языки изучает каждый из них, один юноша ответил: “Вадим изучает китайский, Сергей не изучает китайский, а Михаил не изучает арабский”. Впоследствии выяснилось, что в этом ответе только одно утверждение верно, а два других ложны”.

Какой язык изучает каждый из юношей?

ТВОРЧЕСТВО НАШИХ ЧИТАТЕЛЕЙ

Приведенное ниже стихотворение прислала в редакцию Мария Пинаева, ученица Кезской средней школы № 1, поселок Кез Удмуртской Республики (учитель **Ветошкина Н.В.**).

Главное изобретение

Люди — умные создания,
И они изобрели
Много всяких разных штучек,
Чтоб им силы сберегли.

Колесо здесь и компьютер,
“стиралка”, телеобъектив,
Телефон, машина, скутер —
Это ж суперколлектив!

Этот список же, конечно,
Можно долго продолжать,
Что важней для человека,
Каждый волен сам решать.

Лично мне всего дороже
То, что в древние века
Кто-то изобрел компьютер —
Вот была где голова!

Я не мыслю, как сегодня
Люди могут жить без них?
Мне представить даже страшно...
И не буду — я ж не псих!

Существует и легенда
Как один старик копил,
А потом, собрав деньжищи,
Он компьютер закупил!

Нет, спасибо тому предку,
Что старался так для нас!
Как жилось бы без “Инета”
Человечеству сейчас?

“ЛОМАЕМ” ГОЛОВУ

Сколько будет ОГОГО + УГУГУ?[2]

Приведенный ниже числовой ребус прислал в редакцию Павел Димитриев, Чувашская Республика, г. Чебоксары, лицей № 3 (фамилия учителя в письме, к сожалению, не была сообщена). Решите его. Как принято в таких головоломках, одинаковыми буквами зашифрованы одинаковые цифры, разными буквами — разные цифры.

	О	Г	О	Г	О	
+	У	Г	У	Г	У	
У	Г	У	Г	У	Г	

Откуда взялся один квадратик?

В нашей газете публиковался ряд так называемых “софизмов” — рассуждений, в которых имеются скрытые ошибки, которые приводят к недопустимому результату (от греческого слова *sóphisma* — уловка, ухищрение, выдумка, головоломка). В них, как правило, рассматривались числа. А здесь мы рассмотрим, так сказать, “геометрический” софизм.

Возьмем квадрат размером 8 на 8 маленьких квадратиков (см. *рис.* 1) и разрежем его так, как показано на рисунке. Из полученных отдельных частей составим прямоугольник, представленный на *рис.* 2.

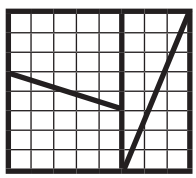


Рис. 1

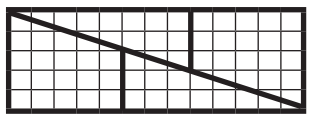


Рис. 2

Проведем расчеты. Площадь исходного квадрата равна 64 единицам. Ширина прямоугольника составляет 8 + 5 = 13 единиц, а высота — 5, т.е. его площадь равна 65 единицам. Откуда взялся еще один квадратик?

Ответы присылайте в редакцию.

Уточнение

В статье “Запись чисел в решетках” (“В мир информатики” № 142/“Информатика” № 8/2010) задача 2*в* должна выглядеть так: “Определите, как будет записана в решетке денежная сумма: три рубля двадцать четыре алтына четыре денги и пол ѣполушки”.

Предлагаем читателям еще раз ознакомиться с указанной статьей.

[2] Задание предназначено для учащихся начальной школы и учеников 5–7-х классов.

Умножение многозначных чисел в уме

Вы, наверное, слышали о людях, умевших умножать в уме многозначные числа (см., например, [1])? А не хотите ли научиться делать это сами? Если да, то эта статья — для вас.

Методику вычислений рассмотрим на примере умножения двух чисел: 7496 и 3852. Результат умножения представлен в таблице:

				7	4	9	6
				3	8	5	2
				14	8	18	12
			35	20	45	30	
		56	32	72	48		
	21	12	27	18			
2	8	8	7	4	5	9	2
	7	*10*	*13*	*10*	*4*	*1*	

В первой строке таблицы записаны произведения первой цифры множителя (3852) на каждую цифру множимого 7496, во второй — произведения второй цифры множителя на каждую цифру множимого и т.д. Но для вычислений легче запомнить следующее: в любом из столбцов таблицы стоит:

1) произведение цифры единиц множителя на соответствующую цифру (α) множимого;

2) произведения остальных цифр множителя, перебираемых справа налево, на цифры множимого, стоящие правее цифры (α) множимого (перебирая их одновременно слева направо).

Опишем это сложное на первый взгляд требование подробнее.

1. В крайнем справа разряде цифра единиц множителя равна 2, соответствующая цифра множимого — 6, их произведение — 12. Здесь $\alpha = 6$. Для других цифр множителя произведения отсутствуют, так как правее цифры 6 в множимом цифр нет.

2. В разряде десятков соответствующая цифра множимого — 9, произведение цифры единиц множителя (2) на нее равно 18. Кроме того, рассматриваем следующую цифру (при просмотре справа налево) множителя (5) и следующую (при просмотре слева направо) цифру множимого (6), поэтому в этом разряде записывается также число 30. Здесь $\alpha = 9$.

3. Аналогично, в разряде сотен $\alpha = 4$. В соответствующем столбце записаны следующие произведения:

1) $2 \times 4 = 8$;

2) 5 (цифра слева от 2) $\times$ 9 (цифра справа от 4) = 45;

3) 8 (вторая цифра слева от 2) $\times$ 6 (вторая цифра справа от 4) = 48.

4. В разряде тысяч $\alpha = 7$. В соответствующем столбце записано следующее:

1) $2 \times 7 = 14$;

2) 5 (цифра слева от 2) $\times$ 4 (цифра справа от 7) = 20;

3) 8 (вторая цифра слева от 2) $\times$ 9 (вторая цифра справа от 7) = 72;

4) 3 (третья цифра слева от 2) $\times$ 6 (третья цифра справа от 7) = 18.

5. В разряде десятков тысяч ($\alpha = 0$) записано[3]:

1) $0\ (2 \times 0)$;

2) $5 \times 7 = 35$;

3) $8 \times 4 = 32$;

4) $3 \times 9 = 27$.

6. В разряде сотен тысяч ($\alpha = 0$) записано:

1) $0\ (2 \times 0)$;

2) $5 \times 0 = 0$;

3) $8 \times 7 = 56$;

4) $3 \times 4 = 12$.

7. В разряде миллионов:

1) $0\ (2 \times 0)$;

2) $5 \times 0 = 0$;

3) $8 \times 0 = 0$;

4) $3 \times 7 = 21$.

Складывая затем числа, стоящие в каждом столбце, и прибавляя к ним отмеченные “в уме” десятки, полученные при сложении в соседнем справа разряде (числа “в уме” записаны в нижней строке и выделены курсивом), получим, идя справа налево, все цифры искомого произведения. Во время демонстрации “фокуса” числа, стоящие в строчках, расположенных между горизонтальными линиями, можно не выписывать, а выполнять сложение соответствующих двузначных чисел в уме — для этого надо, конечно, научиться быстро и безошибочно складывать двузначные числа.

Например, если вы уже дошли до разряда сотен (от разряда десятков у нас осталось “4 в уме”), то мысленно произнесите (глядя на 2 и 4, затем на 5 и 9, наконец, на 8 и 6): “восемь и четыре “в уме” — двенадцать; и сорок пять — пятьдесят семь; и сорок восемь — сто пять”; 5 запишите в разряде сотен, а 10 запомните “в уме”.

Попробуйте поупражняться сначала на двузначных числах, затем на трехзначных и т.д., и вы увидите, что при некотором навыке можно быстро и безошибочно находить произведение многозначных чисел.

Литература

1. *Азевич А.И.* Человек-арифмометр. / “В мир информатики” № 116 (“Информатика” № 22/2008).

2. *Доморяд А.П.* Математические игры и развлечения. М.: Физматгиз, 1961.

3. Возведение двузначных чисел в квадрат. / “В мир информатики” № 50 (“Информатика” № 3/2005).

4. Извлечение кубического корня в уме. / “В мир информатики” № 116 (“Информатика” № 22/2008).

5. Возведение в квадрат чисел пятого десятка. / “В мир информатики” № 134 (“Информатика” № 22/2009).

[3] Нули в приведенной выше таблице не показаны.

Издательский дом «Первое сентября»

предлагает уникальную программу скидок для подписчиков!

НОВОЕ В 2010/2011 УЧЕБНОМ ГОДУ!

ХОТИТЕ ПОЛУЧИТЬ

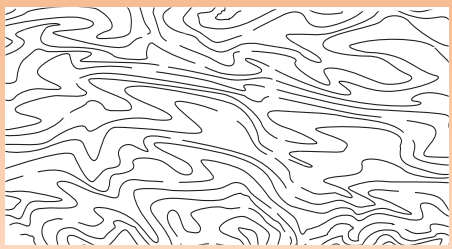
скидку 10% на подписку и бесплатную доставку газет

на первое полугодие 2011 года,

а также возможность получить скидку 20%

в следующую подписную кампанию?

1. **Сотрите защитный слой и прочитайте индивидуальный код** →
2. **Активируйте индивидуальный код на сайте www.1september.ru**
3. **Получите абонемент на скидки**
4. **Подпишитесь через сайт**



Стирая защитный слой, вы подтверждаете, что ознакомлены с условиями скидочной программы, размещенными на сайте www.1september.ru, и принимаете их

КРОМЕ СКИДОК НА ПОДПИСКУ, АБОНЕМЕНТ ДАСТ ВАМ

- **10% скидки** на обучение на дистанционных курсах Педагогического университета «Первое сентября» во II потоке 2010/2011 учебного года и право на 20% скидки при обучении в 2011/2012 году

- **возможность получить 20% скидки** на участие в фестивалях «Открытый урок» и «Портфолио» 2011/2012 года

- **возможность подарить** вашим коллегам сертификаты на скидки

Полностью условия скидочной программы читайте на сайте

www.1september.ru

На диске к этому номеру

Уважаемые коллеги! На диске, приложенном к этому номеру, как обычно, размещены электронные материалы к номерам месяца, в данном случае — августа. Прежде всего мы хотим обратить ваше внимание на очень (!) интересную и в определенном смысле слова уникальную публикацию, дополняющую № 15, — примерную программу по информатике и ИКТ для 7–9-х классов, разработанную С.А. Бешенковым (проф., д. п. н.), А.А. Кузнецовым (академик РАО, проф., д. п. н.), А.Г. Кушниренко (доц., к. ф.-м. н.), А.Л. Семеновым (чл.-корр. РАН и РАО, проф., д. ф.-м. н.). На данный момент эта программа фактически является авторской, но, учитывая коллектив авторов, к ней стоит отнестись более чем серьезно.

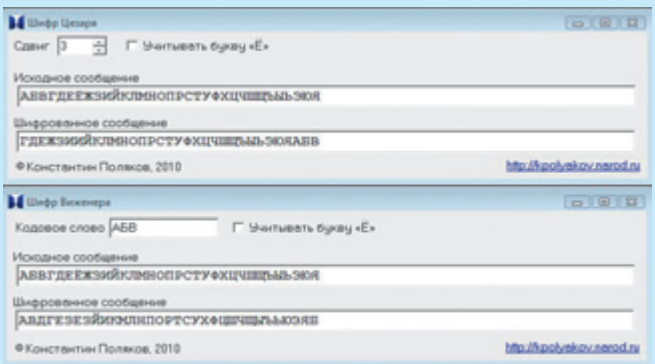
Предлагаемая вашему вниманию программа создана группой авторов, которые уже в течение четверти века участвуют в формировании российской школьной информатики. Роль ИКТ в нашей жизни все возрастает, необходимость фундаментальных знаний в области информатики на уровне школы становится все более очевидной. При этом с большой скоростью идет процесс информатизации школы. Одним из следствий этого процесса является, если можно так сказать, “диффузия” ИКТ из информатики в другие дисциплины. Пожалуй, именно это является основным мотивом для создания новых примерных программ.

Перед автором программы по информатике и ИКТ стоит неразрешимая дилемма — ориентировать программу на школу, где с точки зрения ИКТ “ничего нет” (а такие школы есть), или на школу, где “все есть” (а таких школ становится все больше). В данном случае авторы поступили просто — написали два варианта планирования для созданной программы. При этом во втором варианте планирования возможность передачи части задач по формированию умений и навыков работы с ИКТ в другие дисциплины и для более раннего освоения информатики и ИКТ используется в первую очередь для усиления фундаментально-математического компонента курса.

А.Л. Семенов

Из материалов к № 16 хотим отметить презентацию по теме “Шифры”, подготовленную К.Ю. Поляковым. Помимо презентации, Константин Юрьевич подготовил демонстрационные программы для знакомства с шифрами Цезаря, Вижинера и алгоритмом RSA. Также на диске для ознакомления размещены избранные главы разрабатываемого профильного учебника для 10-го класса.

К статье “Фрактальные собратья” на диске имеются ознакомительные версии программ Easy Tree Generator и Apophysis.



Гл. редактор С.Л. Островский
Редакция
Е.В. Андреева
Д.М. Златопольский (редактор вкладки “В мир информатики”)
Н.П. Медведева
Корректор Е.Л. Володина
Дизайн и верстка Н.И. Пронская
Фото с сайта shutterstock.com

Адрес редакции и издателя: Киевская, 24, Москва, 121165. Тел. 8 (499) 249-48-96
Отдел рекламы: Тел. 8 (499) 249-98-70
Тел.: 8 (499) 249-31-38, 249-33-86. Факс 8 (499) 249-31-38
inf@1september.ru, http://1september.ru

Учредитель: ООО “Чистые пруды”
Зарегистрировано в Министерстве РФ по делам печати.
ПИ № 777230 от 12.04.2001.
Отпечатано в ОАО “Чеховский полиграфический комбинат”, ул. Полиграфистов, д. 1, Московская область, г. Чехов, 142300

Тираж 3500 экз.
Срок подписания в печать по графику 01.07.2010.
Номер подписан 01.07.2010.
Заказ № Цена свободная

ИЗДАТЕЛЬСКИЙ ДОМ “ПЕРВОЕ СЕНТЯБРЯ”
главный редактор – А.С. Соловейчик

ГАЗЕТЫ ИЗДАТЕЛЬСКОГО ДОМА

Первое сентября
гл. ред. — Е.В. Бирюкова, индекс подписки — 32024;

Английский язык
гл. ред. — Е.В. Громушкина, индекс подписки — 32025;

Библиотека в школе
гл. ред. — О.К. Громова, индекс подписки — 33376;

Биология
гл. ред. — Н.Г. Иванова, индекс подписки — 32026;

География
гл. ред. — О.Н. Коротова, индекс подписки — 32027;

Дошкольное образование
гл. ред. — М.С. Аромштам, индекс подписки — 33373;

Здоровье детей
гл. ред. — Н.В. Семина, индекс подписки — 32033;

Информатика
гл. ред. — С.Л. Островский, индекс подписки — 32291;

Искусство
гл. ред. — М.Н. Сартан, индекс подписки — 32584;

История
гл. ред. — А.Л. Савельев, индекс подписки — 32028;

Классное руководство и воспитание школьников
гл. ред. — О.М. Леонтьева, индекс подписки — 19651;

Литература
гл. ред. — С.В. Волков, индекс подписки — 32029;

Математика
гл. ред. — Л.О. Рослова, индекс подписки — 32030;

Начальная школа
гл. ред. — М.В. Соловейчик, индекс подписки — 32031;

Немецкий язык
гл. ред. — М.Д. Бузоева, индекс подписки — 32292;

Русский язык
гл. ред. — Л.А. Гончар, индекс подписки — 32383;

Спорт в школе
гл. ред. — О.М. Леонтьева, индекс подписки — 32384;

Управление школой
гл. ред. — Я.А. Сартан, индекс подписки — 32652;

Физика
гл. ред. — Н.Д. Козлова, индекс подписки — 32032;

Французский язык
гл. ред. — Г.А. Чесновицкая, индекс подписки — 33371;

Химия
гл. ред. — О.Г. Блохина, индекс подписки — 32034;

Школьный психолог
гл. ред. — И.В. Вачков, индекс подписки — 32898.

Издательская подписка
Тел. (499) 249-47-58;
E-mail: podpiska@1september.ru

Индексы подписки
Каталог “Роспечать”:
32291 (инд.),
32591 (орг.),
32744 (комп.)
Каталог “Почта России”:
79006 (инд.),
79574 (орг.)