

ИНФОРМАТИКА

- 4 **Скоро лето**
И ЕГЭ-2015
- 12 **Роботы наступают**
И отступают. И ходят,
и рисуют, и слышат, и
видят, и... вообще это
так увлекательно!
- 26 — **Мне надоели ва-
ши фокусы!**
— Не проблема, у
нас есть новые!

электронная версия журнала
дополнительные
материалы
в Личном кабинете
на сайте
www.1september.ru



► 16 ноября 1922 г. — день рождения Джина Амдала. Это имя и эта дата — “для тех, кто понимает”. Понимает, сколько “гениальный создатель мейн-фреймов” сделал для истории компьютерной науки, в которую его имя останется вписано навсегда. Амдал был генеральным конструктором и главным разработчиком таких легендарных ЭВМ, как IBM 704, 709, 7030, 7090. Вершиной его творчества в IBM стала разработка архитектуры машин семейства IBM 360 — компьютеров, сделавших историю.

3 ПАРА СЛОВ

► У вас пароль не 1234567890? Значит, вам есть что скрывать?!

4 ЕГЭ

► Что год грядущий нам готовит: ЕГЭ-2015

12 РОБОТОТЕХНИКА

► Пять уроков по робототехнике

26 ВНЕКЛАСНАЯ РАБОТА

► Фокусы с “информатическим” секретом
 ► Интеллектуальные алгоритмы и искусственный интеллект

48 ЗАНИМАТЕЛЬНЫЕ МАТЕРИАЛЫ ДЛЯ ПЫТЛИВЫХ УЧЕНИКОВ И ИХ ТАЛАНТЛИВЫХ УЧИТЕЛЕЙ

► “В мир информатики” № 202

Облачные технологии от Издательского дома “Первое сентября”

Уважаемые подписчики бумажной версии журнала!

Дополнительные материалы к номеру и электронная версия журнала находятся в вашем Личном кабинете на сайте www.1september.ru.

Для доступа к материалам воспользуйтесь, пожалуйста, кодом доступа, вложенным в №7–8/2014.

Срок действия кода: с 1 июля по 31 декабря 2014 года.

Для активации кода:

- зайдите на сайт www.1september.ru;
- откройте Личный кабинет (создайте, если у вас его еще нет);
- введите код доступа и выберите свое издание.

Справки: podpiska@1september.ru или через службу поддержки на портале “Первое сентября”.



ЭЛЕКТРОННЫЕ МАТЕРИАЛЫ

► Презентации и исходные файлы к статьям номера

ИНФОРМАТИКА

ПОДПИСНЫЕ ИНДЕКСЫ по каталогу “Почта России”: 79066 — бумажная версия, 12684 — электронная версия

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

Учебно-методический журнал для учителей информатики
 Основан в 1995 г.
 Выходит один раз в месяц

РЕДАКЦИЯ:
 гл. редактор С.Л. Островский
 редакторы

Е.В. Андреева,
 Д.М. Златопольский
 (редактор вкладки
 “В мир информатики”)

Дизайн макета И.Е. Лукьянов
 верстка Н.И. Пронская
 корректор Е.Л. Володина
 секретарь Н.П. Медведева
 Фото: фотобанк Shutterstock
 Журнал распространяется по подписке
 Цена свободная
 Тираж 27 000 экз.
 Тел. редакции: (499) 249-48-96
 E-mail: inf@1september.ru
<http://inf.1september.ru>

ИЗДАТЕЛЬСКИЙ ДОМ
 “ПЕРВОЕ СЕНТЯБРЯ”

Главный редактор:
 Артем Соловейчик
 (генеральный директор)

Коммерческая деятельность:
 Константин Шмарковский
 (финансовый директор)

Развитие, IT
 и координация проектов:
 Сергей Островский
 (исполнительный директор)

Реклама, конференции
 и техническое обеспечение
 Издательского дома:
 Павел Кузнецов

Производство:
 Станислав Савельев

Административно-
 хозяйственное обеспечение:
 Андрей Ушков

Педагогический университет:
 Валерия Арсланьян (ректор)

ЖУРНАЛЫ ИЗДАТЕЛЬСКОГО ДОМА
 “ПЕРВОЕ СЕНТЯБРЯ”
 Английский язык – А.Громушкина
 Библиотека в школе – О.Громова
 Биология – Н.Иванова
 География – и.о. А.Митрофанов
 Дошкольное образование – Д.Тюттерин
 Здоровье детей – Н.Сёмина
 Информатика – С.Островский
 Искусство – О.Волкова
 История – А.Савельев
 Классное руководство и воспитание школьников – М.Битянова

Литература – С.Волков
 Математика – Л.Рослова
 Начальная школа – М.Соловейчик
 Немецкий язык – М.Бузоева
 ОБЖ – А.Митрофанов
 Русский язык – Л.Гончар
 Спорт в школе – О.Леонтьева
 Технология – А.Митрофанов
 Управление школой – Е.Рачевский
 Физика – Н.Козлова
 Французский язык – Г.Чесновицкая
 Химия – О.Блохина
 Школа для родителей – Л.Печатникова
 Школьный психолог – М.Чибисова

УЧРЕДИТЕЛЬ:
 ООО “ЧИСТЫЕ ПРУДЫ”

Зарегистрировано
 ПИ № ФС77-44341
 от 22.03.2011
 в Министерстве РФ
 по делам печати
 Подписано в печать:
 по графику 23.09.2014,
 фактически 23.09.2014
 Заказ №
 Отпечатано в ОАО “Первая
 Образцовая типография”
 Филиал “Чеховский Печатный Двор”
 ул. Полиграфистов, д. 1,
 Московская область,
 г. Чехов, 142300
 Сайт: www.chpd.ru
 E-mail: sales@chpk.ru
 Факс: 8 (495) 988-63-76

АДРЕС ИЗДАТЕЛЯ:
 ул. Киевская, д. 24,
 Москва, 121165
 Тел./факс: (499) 249-31-38

Отдел рекламы:
 (499) 249-98-70
<http://1september.ru>

ИЗДАТЕЛЬСКАЯ ПОДПИСКА:
 Телефон: (499) 249-47-58
 E-mail: podpiska@1september.ru

BANK
XXY274SECURITY
ZZZ4H3GE-MAIL
H24407WORK
247742CNETWORK
J7-42QE

PASS

У вас пароль не 1234567890? Значит, вам есть что скрывать?!

► Для немалого количества российских пользователей событием осени 2014 года стала вовсе не презентация новых гаджетов Apple, а крупный скандал, связанный с публикацией нескольких миллионов (!) логинов и паролей электронной почты.

Первой “ласточкой” стала публикация 5 сентября в Интернете базы адресов с “взломанными” паролями более миллиона ящиков e-mail на Яндексе. Однако не успели пользователи другого популярного в России почтового сервиса — Mail.Ru — порадоваться, что их почтовые ящики не на Яндексе, как 8 сентября в Интернете появился аналогичный список из почти 4,5 миллиона “взломанных” адресов на Мейле. А уже 10 сентября аналогичным образом в сеть “утекли” и пароли более 4,5 миллиона пользователей Гугловского Gmail (правда, в этом случае пострадавшими стали в основном иностранцы).

Что это было — утечка ли конфиденциальных данных непосредственно с серверов почтовых сервисов или массовая кампания по взлому паролей, — эксперты все еще спорят. Специалисты же из технических служб почтовых сервисов утверждают, что указанные базы адресов — лишь скомпилированные кем-то из старой, разрозненной информации об отдельных паролях почтовых ящиков, “взломанных” за многие годы до этого события при помощи троянских программ или “фишинга”. Так, представители Mail.Ru заявили, что из опубликованной базы “их” адресов e-mail более 95% уже считались “подозрительными” (например, с них была замечена рассыл-

ка спама) или “забытыми” пользователями, а потому эти ящики или были уже заблокированы, или их владельцам высылались предупреждения о необходимости сменить пароль.

Что бы то ни было, — какие выводы нужно сделать из всего этого?

Во-первых, владельцам почтовых ящиков на Mail.Ru, Yandex.Ru и Gmail.com рекомендуется лишний раз поменять пароль (что, кстати, полезно периодически делать в любом случае). Можно также проверить свой почтовый адрес на его нахождение в опубликованных базах паролей — для этого окажется полезным специальный сайт с весьма “говорящим” названием <http://yaslit.ru>. На этом ресурсе хранятся списки “скомпрометированных” логинов электронной почты на сервисах Яндекс.Почта, Gmail и Mail.Ru (включая все возможные вариации последнего — Inbox, List и Bk). Достаточно ввести свой логин в соответствующую графу, и компьютер вынесет свой вердикт: нужно ли менять пароль *срочно*, или можно не так спешить.

Во-вторых, можно подключить “двухфакторную” авторизацию: например, в Gmail она реализована еще с 2010 года. В этом случае к аккаунту привязывается номер сотового телефона, и, кроме логина и пароля, для входа на ресурс требуется также ввести одноразовый числовой код, который будет прислан через SMS (на Mail.Ru и Яндекс.Почте такая процедура предлагается при смене пароля).

В-третьих же — это еще один повод задуматься о принципиальной ненадежности самих электронных технологий (вспомнив известный тезис: “к любому замку можно подобрать отмычку”) и не доверять электронной переписке действительно конфиденциальные сведения. А также надеяться, что IT-специалисты все же сумеют найти более надежные способы авторизации, вместе с тем удобные для пользователя и не заставляющие его запоминать множество различных “неудобочитаемых” паролей...

Дмитрий Усенков



Что год грядущий нам готовит: ЕГЭ-2015

К.Ю. Поляков,
д. т. н., Санкт-Петербург

► Как и всегда, к началу нового учебного года в сети появился демовариант контрольно-измерительных материалов (КИМ) будущего ЕГЭ, пока в статусе проекта [1]. Цель настоящей заметки — сравнить этот проект с прошлогодним демовариантом [2] и проанализировать внесенные изменения.

Прежде всего отметим, что демовариант составлен в классическом стиле для “бумажного” ЕГЭ. Таким образом, введение компьютерного варианта экзамена снова откладывается.

Согласно традиции последних лет, в демовариант включены задачи, которые входили в реальные варианты КИМ, предлагавшиеся на последнем проведенном ЕГЭ.

Изменение формы

Сразу заметно, что изменилась нумерация (она стала сквозной, без букв А, В и С). Это коснулось не только информатики, но и всех предметов. Скорее всего это новшество упрощает автоматическую обработку материалов, однако усложняет запоминание для человека (убрано деление на категории). Это все равно что из файловой системы убрать каталоги и вернуться к единому списку файлов. Таким образом, компьютеры наступают на людей ☺.

Всего в проекте демоварианта 27 заданий (вместо 32 в ЕГЭ-2014), и только три из них — с выбором ответа. Используя “старую” терминологию, все больше заданий переводится из группы А в группу В, что можно только приветствовать, — игра “Угадайка” постепенно уходит в прошлое. В следующей таблице показано соответствие заданий ЕГЭ-2015 и ЕГЭ-2014:

	ЕГЭ-2015	ЕГЭ-2014	Материал
А	1	A9	кодирование и декодирование данных
	2	A3	таблицы истинности логической функции
	3	A4 A6	файловая система сортировка и поиск в базах данных
В	4	A1	кодирование чисел в разных системах счисления
	5	A2	анализ информационных моделей
	6	A5 B1	выполнение и анализ простых алгоритмов анализ и построение алгоритмов для исполнителей
	7	A7 B3	адресация в электронных таблицах анализ диаграмм в электронных таблицах
	8	B5	анализ программ с циклами
	9	A8 B10	кодирование звуковой информации скорость передачи информации
	10	B4	кодирование, комбинаторика, системы счисления
	11	B6	рекурсивные алгоритмы
	12	B11	адресация в Интернете
	13	A11	вычисление количества информации
	14	A13	анализ и выполнение алгоритмов для исполнителя
	15	B9	поиск путей в графе
	16	B7	кодирование чисел, системы счисления
	17	B12	сложные запросы для поисковых систем
	18	A10	проверка истинности логического выражения
	19	A12	обработка массивов и матриц
	20	B8	анализ программы с циклами и условными операторами
	21	B14	анализ программ с циклами и подпрограммами
	22	B13	перебор вариантов, динамическое программирование
	23	B15	системы логических уравнений
С	24	C1	поиск ошибок в программе
	25	C2	алгоритмы обработки массивов
	26	C3	теория игр
	27	C4	обработка массивов, символьных строк и последовательностей

Из этой таблицы следует, что

1) принципиально новых задач, которые раньше не встречались в КИМах, добавлено не было; поэтому все “старые” ресурсы (например, [3]) и литературу можно использовать для подготовки к экзамену;

2) уменьшение количества заданий достигнуто за счет объединения некоторых тем из ЕГЭ-2014: в проекте демоварианта появились задания, в которых приведены две задачи, связанные словом ИЛИ (это задания 3, 6, 7 и 9);

3) убрано простое задание В2 из ЕГЭ-2014 (анализ простой программы с операторами присваивания и условными операторами);

4) среди заданий с развернутым ответом (в “бывшей” части С) практически ничего, кроме нумерации, не поменялось.

Сокращение количества задач — это палка о двух концах. С одной стороны, чем меньше задач, тем меньше затрачивается усилий. Заметим, что объединялись только простые задачи, решение которых легко алгоритмизируется, то есть ученика несложно на них “натаскать”. С другой стороны, возрастает влияние случайной ошибки, вес которой в первичной сумме баллов теперь уже $1/35$, а не $1/40$, как было в ЕГЭ-2014. Не секрет, что в последние несколько лет были работы с единственным потерянным

баллом в задаче А13 (задача с Роботом). Вряд ли можно предположить, что человек, написавший на полный балл всю часть С, недостаточно знает предмет. Поэтому уменьшение количества задач скорее всего немного усложнит сдачу экзамена: большую роль будет играть внимательность.

Пожалуй, наиболее интересно появление “двойных” заданий (3, 6, 7 и 9), каждое из которых представлено в демоварианте двумя задачами. Задание 6 включает простые задачи на алгоритмизацию, а остальные три “двойных” задания объединяют задачи по теоретической информатике (задание 9) и информационным технологиям (3 и 6). Таким образом, в будущем ЕГЭ повышается роль задач на программирование (эта тенденция наметилась уже в прошлые годы). Здесь есть и некоторый подвох: можно подумать, что в “двойных” заданиях реальных КИМ *точно* будет один из двух вариантов задач, которые представлены в демоварианте. Однако здравый смысл должен подсказывать ученику, что это не так. В “боевом” варианте он может получить аналог любой из двух приведенных в демоварианте задач но скорее всего не точную копию (формулировка может отличаться с сохранением сути проверяемых знаний).

Отметим и изменения в формулировке сложной задачи на программирование (задание 27, аналог

задания С4 в ЕГЭ-2014). Как и в реальных КИМ 2014 года, здесь уточняется, что для получения максимальной оценки 4 балла нужно написать программу, эффективную и по времени выполнения, и по используемой памяти; для получения трех баллов достаточно обеспечить эффективность по времени, а за правильное решение, неэффективное ни по времени, ни по памяти, можно получить 2 балла.

Изменение содержания

Что же поменялось по существу? Прежде всего во всех заданиях на программирование приводится вариант кода на языке Python, который уверенно пробивает себе дорогу в школу. Действительно, запись программ на Python выглядит привлекательно, хотя с точки зрения преподавания Python в школе есть еще довольно много методических вопросов [5].

В некоторых задачах вопросы поставлены немного по-другому. При этом попытка решения задачи “в лоб” требует значительных усилий, а применение знаний теории нередко позволяет довольно быстро получить правильный ответ.

Задание 2

Задача 2.1. Александра заполняла таблицу истинности для выражения F . Она успела заполнить лишь небольшой фрагмент таблицы:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	F
	0						1	0
1			0					1
			1				1	1

Каким выражением может быть F ?

- $x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge x_6 \wedge \neg x_7 \wedge \neg x_8$
- $x_1 \vee x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee \neg x_6 \vee \neg x_7 \vee \neg x_8$
- $\neg x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4 \wedge x_5 \wedge \neg x_6 \wedge \neg x_7 \wedge \neg x_8$
- $x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee \neg x_6 \vee \neg x_7 \vee \neg x_8$

Решение. Как и в заданиях прошлых лет, нам дана часть таблицы истинности. Однако значения переменных в трех заданных строчках известны только частично. Если в качестве возможных ответов задать произвольные функции, то для проверки, например, соответствует ли функция $F_1(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$ первой строчке таблицы, необходимо проверить все возможные значения функции $F_1(x_1, 0, x_3, x_4, x_5, x_6, x_7, 1)$ при всех комбинациях неизвестных переменных и выяснить, может ли какая-то из этих комбинаций дать в результате 0. Фактически задача сводится к проверке разрешимости уравнения $F_1(x_1, 0, x_3, x_4, x_5, x_6, x_7, 1) = 0$ на множестве двоичных значений $(x_1, x_3, x_4, x_5, x_6, x_7)$.

К счастью, в демоварианте авторы немного облегчили жизнь экзаменуемым (и их учителям), выбрав функции-ответы специального вида, которые позволяют решить задачу устно, не выполняя никаких вычислений. Чтобы нам было проще вос-

принимать функции, заменим конъюнкцию знаком “ \wedge ”, а дизъюнкцию — знаком “ \vee ” [4]:

- $x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4 \cdot x_5 \cdot x_6 \cdot \bar{x}_7 \cdot \bar{x}_8$
- $x_1 + x_2 + x_3 + \bar{x}_4 + \bar{x}_5 + \bar{x}_6 + \bar{x}_7 + \bar{x}_8$
- $\bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 \cdot x_5 \cdot \bar{x}_6 \cdot \bar{x}_7 \cdot \bar{x}_8$
- $x_1 + \bar{x}_2 + x_3 + \bar{x}_4 + \bar{x}_5 + \bar{x}_6 + \bar{x}_7 + \bar{x}_8$

В последнем столбце таблицы истинности видим две единицы, откуда сразу следует, что это не может быть цепочка операций “И” (конъюнкций), которая дает только одну единицу; поэтому ответы 1 и 3 заведомо неверные. Следовательно, остаются только варианты 2 и 4, дизъюнкции (логические суммы) переменных и их инверсий.

Анализируем первую строку таблицы истинности. Мы знаем в ней только два значения — $x_2 = 0$ и $x_8 = 1$. Для того чтобы в результате в первой строке получить 0, необходимо, чтобы переменная x_8 входила в логическую сумму с инверсией (тогда из 1 получится 0). Это условие выполняется для обоих оставшихся вариантов, 2 и 4.

Кроме того, переменная x_2 должна входить в логическую сумму без инверсии (иначе соответствующее слагаемое в первой строке равно 1, и это даст в результате 1). Этому условию удовлетворяет только выражение 2. Это и есть правильный ответ.

Рассмотрим теперь второй вариант задачи, когда правильный ответ — конъюнкция переменных или их инверсий.

Задача 2.2. Александра заполняла таблицу истинности для выражения F . Она успела заполнить лишь небольшой фрагмент таблицы:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	F
	0						1	1
1			0					0
			1				1	0

Каким выражением может быть F ?

- $x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge x_6 \wedge \neg x_7 \wedge \neg x_8$
- $x_1 \vee x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee \neg x_6 \vee \neg x_7 \vee \neg x_8$
- $x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge x_4 \wedge x_5 \wedge \neg x_6 \wedge \neg x_7 \wedge x_8$
- $x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee \neg x_6 \vee \neg x_7 \vee \neg x_8$

Решение. Здесь в последнем столбце таблицы видим одну единицу и два нуля, поэтому это не может быть дизъюнкция, которая дает ноль только при одном наборе значений переменных. Таким образом, варианты 2 и 4 заведомо неверные, нужно сделать выбор между ответами 1 и 3.

Рассматриваем “особую” строчку таблицы, в которой функция равна 1. Поскольку мы говорим о конъюнкции, переменная x_2 должна входить в нее с инверсией (это выполняется для обоих оставшихся вариантов), а переменная x_8 — без инверсии. Последнее из условий верно только для варианта 3, это и есть правильный ответ.

Заметим, что функции-ответы не обязательно должны представлять собой дизъюнкцию или

конъюнкцию. В общем случае вычисления будут несколько сложнее.

Задача 2.3. Александра заполняла таблицу истинности для выражения F . Она успела заполнить лишь небольшой фрагмент таблицы:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	F
	0						1	1
1			0					0
			1				1	0

Каким выражением может быть F ?

- $\neg x_1 \wedge x_2 \vee x_2 \wedge \neg x_3 \wedge \neg x_4 \vee x_2 \wedge \neg x_5 \vee x_5 \wedge x_6 \wedge \neg x_7 \wedge \neg x_8$
- $(x_1 \wedge \neg x_2 \vee x_3 \vee x_4) \wedge (x_5 \vee x_6 \vee \neg x_7 \vee x_8)$
- $x_1 \wedge \neg x_8 \vee \neg x_3 \wedge x_4 \wedge x_5 \vee \neg x_6 \wedge \neg x_7 \wedge x_8$
- $x_1 \wedge \neg x_4 \vee x_2 \wedge x_3 \wedge \neg x_4 \vee \neg x_5 \vee \neg x_6 \vee \neg x_7 \vee \neg x_8$

Решение. Среди заданных вариантов ответа нет “чистых” конъюнкций и дизъюнкций, поэтому мы должны проверить возможные значения всех заданных выражений для каждой строки таблицы. Перепишем выражения, используя обозначения логических операций из [4]:

- $\bar{x}_1 \cdot x_2 + x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 + x_2 \cdot \bar{x}_5 + x_5 \cdot x_6 \cdot \bar{x}_7 \cdot \bar{x}_8$
- $(x_1 \cdot \bar{x}_2 + \bar{x}_3 + x_4) \cdot (x_5 + x_6 + \bar{x}_7 + x_8)$
- $x_1 \cdot \bar{x}_8 + \bar{x}_3 \cdot x_4 \cdot x_5 + \bar{x}_6 \cdot \bar{x}_7 \cdot x_8$
- $x_1 \cdot \bar{x}_4 + x_2 \cdot x_3 \cdot \bar{x}_4 + \bar{x}_5 + \bar{x}_6 + \bar{x}_7 + \bar{x}_8$

Подставим известные значения переменных из первой строчки таблицы, $x_2 = 0$ и $x_8 = 1$:

- $\bar{x}_1 \cdot 0 + 0 \cdot x_3 \cdot \bar{x}_4 + 0 \cdot \bar{x}_5 + x_5 \cdot x_6 \cdot \bar{x}_7 \cdot 0 = 0$
- $(x_1 \cdot 1 + \bar{x}_3 + x_4) \cdot (x_5 + x_6 + \bar{x}_7 + 1) = x_1 + \bar{x}_3 + x_4$
- $x_1 \cdot 0 + \bar{x}_3 \cdot x_4 \cdot x_5 + \bar{x}_6 \cdot \bar{x}_7 \cdot 1 = \bar{x}_3 \cdot x_4 \cdot x_5 + \bar{x}_6 \cdot \bar{x}_7$
- $x_1 \cdot \bar{x}_4 + 0 \cdot x_3 \cdot \bar{x}_4 + \bar{x}_5 + \bar{x}_6 + \bar{x}_7 + 0 = x_1 \cdot \bar{x}_4 + \bar{x}_5 + \bar{x}_6 + \bar{x}_7$

Видим, что первое выражение при $x_2 = 0$ и $x_8 = 1$ всегда равно нулю, поэтому вариант 1 не подходит. Остальные выражения вычислимы, то есть могут быть равны как 0, так и 1.

Подставляем в оставшиеся три выражения известные данные из второй строчки таблицы, $x_1 = 1$ и $x_4 = 0$:

- $(1 \cdot \bar{x}_2 + \bar{x}_3 + 0) \cdot (x_5 + x_6 + \bar{x}_7 + x_8) = (\bar{x}_2 + \bar{x}_3) \cdot (x_5 + x_6 + \bar{x}_7 + x_8)$
- $1 \cdot \bar{x}_8 + \bar{x}_3 \cdot 0 \cdot x_5 + \bar{x}_6 \cdot \bar{x}_7 \cdot x_8 = \bar{x}_8 + \bar{x}_6 \cdot \bar{x}_7 \cdot x_8$
- $1 \cdot 1 + x_2 \cdot x_3 \cdot 1 + \bar{x}_5 + \bar{x}_6 + \bar{x}_7 + \bar{x}_8 = 1$

Выражение 4 при этих данных всегда равно 1, поэтому получить $F = 0$, как задано в таблице, невозможно. Этот вариант не подходит.

Остаются выражения 2 и 3. Подставляем в них известные данные из третьей строчки таблицы, $x_4 = 1$ и $x_8 = 1$:

- $(x_1 \cdot \bar{x}_2 + \bar{x}_3 + 1) \cdot (x_5 + x_6 + \bar{x}_7 + 1) = 1$
- $x_1 \cdot 0 + \bar{x}_3 \cdot 1 \cdot x_5 + \bar{x}_6 \cdot \bar{x}_7 \cdot 1 = \bar{x}_3 \cdot x_5 + \bar{x}_6 \cdot \bar{x}_7$

Выражение 2 в этом случае всегда равно 1, поэтому оно не подходит (по таблице истинности при этих данных мы должны получить 0). Выражение 3 вычислимо, это и есть правильный ответ.

Задание 10

“Новое — это хорошо забытое старое”. Разработчики КИМ вернулись к задаче В4 из демоварианта 2012 года [6].

Задача 10. Все четырехбуквенные слова, составленные из букв К, Л, Р, Т, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

- КККК
- КККЛ
- КККР
- КККТ

.....

Запишите слово, которое стоит под номером 67.

Решение. Самый простой вариант решения этой задачи — использование систем счисления. Действительно, здесь расстановка слов в алфавитном порядке равносильна расстановке по возрастанию чисел, записанных в четверичной системе счисления (по количеству используемых букв). Выполним замену $K \rightarrow 0, L \rightarrow 1, R \rightarrow 2, T \rightarrow 3$. Поскольку нумерация слов начинается с единицы, а первое число КККК $\rightarrow 0000$ равно 0, под номером 67 будет стоять закодированное число 66. Его нужно перевести в четверичную систему: $66 = 1002_4$ и выполнить обратную замену (цифр на буквы). В результате получаем слово ЛККР.

Задание 11

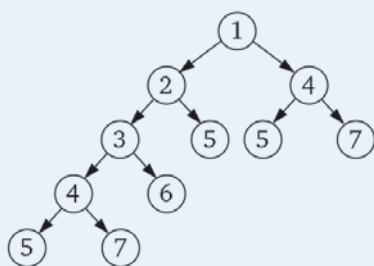
Задание на рекурсию немного изменилось по форме — теперь при рекурсивных вызовах параметр увеличивается, а не уменьшается, и это может сбивать учащихся с толку.

Задача 11. Ниже на языке программирования¹ записан рекурсивный алгоритм F . Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова $F(1)$?

```
procedure F(n: integer);
begin
  writeln(n);
  if n < 5 then begin
    F(n + 1);
    F(n + 3)
  end
end
```

Решение. Поскольку в начале каждого вызова на экран выводится значение единственного параметра функции, достаточно определить порядок рекурсивных вызовов и сложить значения параметров. Поскольку при $n < 5$ выполняется два рекурсивных вызова, решать такую задачу “на бумажке” удобно в виде двоичного дерева (в узлах записаны значения параметров при вызове функции):

¹ Здесь и далее в статье используется язык Паскаль. — Прим. автора.



Складывая все эти числа, получаем ответ 49.

Можно обойтись и без дерева, учитывая, что при каждом вызове с $n < 5$ происходит два рекурсивных вызова. Тогда сумму чисел $S(n)$, полученных при вызове $F(n)$, можно записать как

$$S(n) = \begin{cases} n + S(n+1) + S(n+3), & n < 5 \\ n, & n \geq 5 \end{cases}$$

Выполняем вычисления:

$$\begin{aligned} S(1) &= 1 + S(2) + S(4) \\ S(2) &= 2 + S(3) + S(5) = 7 + S(3) \\ S(3) &= 3 + S(4) + S(6) = 9 + S(4) \\ S(4) &= 4 + S(5) + S(7) = 16 \end{aligned}$$

Теперь остается найти ответ “обратным ходом”:

$$\begin{aligned} S(3) &= 9 + 16 = 25 \\ S(2) &= 7 + 25 = 32 \\ S(1) &= 1 + 32 + 16 = 49 \end{aligned}$$

Ответ: 49.

Задание 14

В некоторых “боевых” вариантах КИМ на прошедшем ЕГЭ-2014 задача на исполнителя Робот, которую многие критиковали, была заменена на более интересную задачу для Чертежника. Эта задача и появилась в демоварианте ЕГЭ-2015. В отличие от задач с Роботом новая формулировка значительно снижает риск случайной ошибки из-за невнимательности.

Задача 14. Исполнитель Чертежник перемещается на координатной плоскости, оставляя след в виде линии. Чертежник может выполнять команду **сместиться на (a, b)**, где a, b — целые числа. Эта команда перемещает Чертежника из точки с координатами (x, y) в точку с координатами (x + a; y + b). Например, если Чертежник находится в точке с координатами (4, 2), то команда **сместиться на (2, -3)** переместит Чертежника в точку (6, -1).

Цикл

ПОВТОРИ число РАЗ
последовательность команд
КОНЕЦ ПОВТОРИ

означает, что последовательность команд будет выполнена указанное число раз (число должно быть натуральным). Чертежнику был дан для исполнения следующий алгоритм (буквами n, a, b обозначены неизвестные числа, при этом $n > 1$):

НАЧАЛО

сместиться на (-3, -3)

ПОВТОРИ n РАЗ

сместиться на (a, b)

сместиться на (27, 12)

КОНЕЦ ПОВТОРИ

сместиться на (-22, -7)

КОНЕЦ

Укажите наименьшее возможное значение числа n, для которого найдутся такие значения чисел a и b, что после выполнения программы Чертежник возвратится в исходную точку.

Решение. Запишем общее изменение координат Чертежника в результате выполнения этого алгоритма:

$$\begin{aligned} \Delta x &= -3 + n(a+27) - 22 = n(a+27) - 25 \\ \Delta y &= -3 + n(b+12) - 7 = n(b+12) - 10 \end{aligned}$$

Поскольку Чертежник должен вернуться в исходную точку, эти величины должны быть равны нулю. Следовательно, нужно найти наименьшее натуральное $n > 1$, при котором система уравнений

$$\begin{cases} n(a+27) - 25 = 0 \\ n(b+12) - 10 = 0 \end{cases}$$

разрешима в целых числах относительно a и b. Несложно заметить, что для этого число n должно быть одновременно делителем чисел 10 и 25. Их наименьший общий делитель равен 5. Таким образом, правильный ответ — 5.

Задание 16

Задачи такого рода уже встречались на пробных компьютерных ЕГЭ, которые проводились в прошлые годы.

Задача 16. Сколько единиц содержится в двоичной записи значения выражения:

$$4^{2014} + 2^{2015} - 8?$$

Решение. Заметим, что все слагаемые этого выражения можно представить как степени числа 2, что существенно облегчает дело:

$$2^{4028} + 2^{2015} - 2^3$$

Нас интересует двоичная запись этого числа. Вспомним, что число 2^N записывается в двоичной системе как единица, за которой следуют N нулей.

Первое слагаемое, 2^{4028} , добавляет одну старшую единицу. Запомним это и разберемся с разностью $2^{2015} - 2^3$. Рассмотрим задачу в общем виде: нужно определить количество единиц в двоичной записи числа $2^N - 2^K$ при $N > K$. Применяя правила вычитания в двоичной системе, находим, что в разности будет $N - K$ единиц, после которых следуют K нулей:



Таким образом, в двоичной записи числа $2^{2015} - 2^3$ содержится $2015 - 3 = 2012$ единиц. Кроме того, как мы помним, еще одну единицу даст слагаемое 2^{4028} , поэтому в исходном числе будет 2013 единиц.

Задание 17

Задача на поисковые запросы в Интернете несколько усложнилась. Если в предыдущие годы большинство задач фактически сводилось к про-

стейшей задаче с двумя областями, теперь придется рисовать три круга Эйлера.

Задача 17. В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

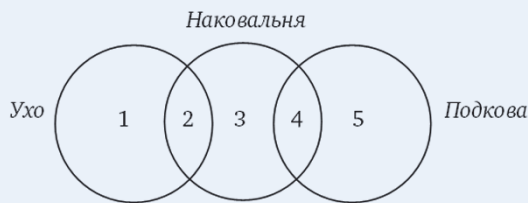
Запрос	Найдено страниц (в сотнях тысяч)
Ухо	35
Подкова	25
Наковальня	40
Ухо Подкова Наковальня	70
Ухо & Наковальня	10
Ухо & Подкова	0

Какое количество страниц (в сотнях тысяч) будет найдено по запросу

Подкова & Наковальня

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Решение. Сразу заметим, что области Ухо и Подкова не пересекаются (см. последнюю строчку таблицы). Поэтому можно нарисовать такую диаграмму:



Здесь пять областей, причем известны следующие данные (N_k обозначает количество найденных страниц, в сотнях тысяч, для области с номером K):

$$\begin{aligned} N_1 + N_2 &= 35 \\ N_4 + N_5 &= 25 \\ N_2 + N_3 + N_4 &= 40 \\ N_1 + N_2 + N_3 + N_4 + N_5 &= 70 \\ N_2 &= 10 \end{aligned}$$

Нас интересует область 4. Находим ответ прямой подстановкой:

$$\begin{aligned} N_1 &= 35 - N_2 = 25 \\ N_5 &= 70 - N_1 - (N_2 + N_3 + N_4) = 70 - 25 - 40 = 5 \\ N_4 &= 25 - N_5 = 20 \end{aligned}$$

Таким образом, ответ — 20.

Возможен и другой вариант решения. Заметим, что в прямую сумму величин областей Ухо, Подкова и Наковальня дважды входят области 2 и 4, поэтому для вычисления N_4 достаточно вычесть из суммы Ухо + Подкова + Наковальня размер их объединения (Ухо | Подкова | Наковальня) и величину области 2 (Ухо & Наковальня). Тогда сразу получаем

$$N_4 = (35 + 25 + 40) - 70 - 10 = 20.$$

Задание 18

Задание 18 — это задание А10 из демоварианта прошлого года. Кроме того, что теперь разработа-

ки не дают вариантов ответа, внешний вид предложенного выражения стал еще более пугающим. Но это как раз пример задания, в котором требуются знания курса информатики, — выражение нужно сначала упростить.

Задача 18. На числовой прямой даны два отрезка: $P = [37; 60]$ и $Q = [40; 77]$. Укажите наименьшую возможную длину такого отрезка A , что формула²

$$(x \in P) \rightarrow ((x \in Q) \wedge \neg(x \in A)) \rightarrow \neg(x \in P)$$

истинна при любом значении переменной x , т.е. принимает значение 1 при любом значении переменной x .

Решение. Следуя [7], упростим запись выражения, перейдя к обозначениям логических операций из [4]:

$$P \rightarrow (Q \cdot \bar{A} \rightarrow \bar{P}),$$

где $P \equiv x \in P$, $Q \equiv x \in Q$ и $A \equiv x \in A$. Раскрываем обе импликации по формуле $A \rightarrow B = \bar{A} + B$:

$$P \rightarrow (Q \cdot \bar{A} + \bar{P}) = \bar{P} + Q \cdot \bar{A} + \bar{P} = Q \cdot \bar{A} + \bar{P}$$

Теперь используем закон де Моргана $\bar{A} \cdot \bar{B} = \overline{A + B}$:

$$\bar{Q} + A + \bar{P}$$

В таком виде выражение уже смотрится совсем не страшно. Сразу видно, что отрезок A должен перекрыть область на числовой оси, которая не входит в область $\bar{Q} + \bar{P}$:



По рисунку видно, что не перекрыт только отрезок $[40; 60]$ (он выделен желтым цветом), его длина — 20, это и есть правильный ответ.

Задание 21

Задание 21 “выросло” из задания В14 прошлого экзамена (анализ программ с циклами и подпрограммами). Как и ранее, при решении нужны некоторые знания математики.

Задача 21.1. Напишите в ответе число различных значений входной переменной k , при которых программа выдает тот же ответ, что и при входном значении $k = 64$. Значение $k = 64$ также включается в подсчет различных значений k .

```
var k, i : longint;
function f(n: longint) : longint;
begin
    f := n * n
end;
begin
    readln(k);
    i := 12;
    while (i > 0) and (f(i) >= k) do
        i := i - 1;
    writeln(i)
end.
```

² На взгляд автора, здесь лучше употребить слово “выражение”, но в оригинале именно так.

Решение. Сначала разберемся, что делает эта программа. В ней используется функция $f(n)$, которая вычисляет квадрат переданного ей числа. В теле основной программы выполняется цикл с условием, который заканчивается, когда значение функции станет меньше k . На каждом шаге цикла уменьшается значение переменной i , начиная с 12. Цикл также заканчивается, когда значение переменной i станет равно 0. После окончания цикла программа выводит значение переменной i .

Итак, функция выводит первое натуральное значение i , квадрат которого меньше, чем введенное с клавиатуры значение переменной k . При $k = 64$ программа выведет значение 7, поскольку это наибольшее натуральное число, квадрат которого меньше, чем 64. Фактически нужно ответить на вопрос: сколько есть таких чисел k , которые меньше или равны $8^2 = 64$ и больше, чем $7^2 = 49$? Действительно, легко проверить, что при $k = 65$ программа выведет значение 8, а при $k = 49$ — значение 6. В диапазоне $[50;64]$ всего $64 - 50 + 1 = 15$ чисел, это и есть правильный ответ.

Задача 21.2. Напишите в ответе число различных значений входной переменной k , при которых программа выдает тот же ответ, что и при входном значении $k = 24$. Значение $k = 24$ также включается в подсчет различных значений k .

```
var i, k: integer;
function f(x:integer):integer;
begin
  if x = 1 then
    f := 1
  else f := x * f(x - 1);
end;
begin
  i := 15;
  readln(k);
  while (i > 0) and (f(i) > k) do
    i := i - 1;
  writeln(i);
end.
```

Решение. В отличие от предыдущей задачи здесь используется рекурсивная функция. Легко понять, что она вычисляет значение факториала переданного ей числа. Программа в цикле уменьшает значение переменной i и выводит первое натуральное значение i , факториал которого меньше или равен введенному числу k . Так для $k = 24$ будет выведено число 4, потому что при $i = 4$ в первый раз нарушается условие $f(i) > k$. То же самое число 4 будет выведено для всех значений k , таких что $4! \leq k < 5! = 120$, то есть для всех k в диапазоне $[24;119]$. Таких чисел $119 - 24 + 1 = 96$.

Заключение

В целом нужно отметить, что принципиальных изменений в проекте КИМ не произошло, процесс развивается эволюционно, как и обещали разработчики. Наблюдается постепенное увеличение роли задач на алгоритмизацию и программирование, снижается количество вопросов на ИКТ-компетентность.

Отметим, что опубликованный проект демо-варианта фактически содержит все новые форму-

лировки задач, которые встретились на реальном экзамене 2014 года. При сравнении его с демо-вариантом КИМ-2014 становится ясно, что при сохранении проверяемого содержания задача из “боевого” варианта по форме может существенно отличаться от задачи из демо-варианта. Это, по замыслу авторов, дает дополнительные преимущества ученикам, которые глубоко знают предмет и способны ориентироваться в изменяющихся условиях.

В КИМ осталась печально известная задача на решение системы логических уравнений (задание 23, аналог “старого” В15). Известно, что ее решаемость бывает ниже, чем решаемость самой сложной задачи на программирование 27 (хотя последняя “стоит” 4 балла, а задача 23 — всего 1 балл). На взгляд автора, эта задача далеко выходит за необходимый уровень подготовки школьников, и в ней сложно найти признаки практической применимости. Поэтому даже сильные преподаватели (и даже в известных физико-математических лицеях!) рекомендуют своим ученикам не решать эту задачу вообще или решать ее в последнюю очередь, когда все остальное решено и проверено и осталось свободное время.

Завис в воздухе вопрос о возможном введении компьютерного ЕГЭ (КЕГЭ), который обсуждается уже несколько лет (во многих регионах даже проводились пробные экзамены). Судя по опубликованному демо-варианту, в 2015 году ждать КЕГЭ точно не нужно.

По-прежнему актуально создание нормального открытого банка заданий для подготовки к ЕГЭ. На существующем сайте [9] практически отсутствует классификация задач, поэтому пользоваться им удается с большим трудом.

Автор благодарит д. ф.-м. н. М.А. Ройтберга за обсуждение этого материала и высказанные замечания.

Литература

1. Демоверсия, спецификация, кодификатор ЕГЭ-2015 по информатике [Электронный ресурс] URL: http://fipi.ru/sites/default/files/document/1409834615/inf11_2015.zip (дата обращения 06.09.2014).
2. Демоверсия, спецификация, кодификатор ЕГЭ-2014 по информатике [Электронный ресурс] URL: http://fipi.ru/sites/default/files/document/1408709790/Inf_ege14.zip (дата обращения 06.09.2014).
3. Подготовка к ЕГЭ по информатике [Электронный ресурс] URL: <http://kpolyakov.spb.ru/school/ege.htm> (дата обращения 06.09.2014).
4. Поляков К.Ю., Еремин Е.А. Информатика. 10-й класс. Углубленный уровень. В двух частях. М.: Бином, 2014.
5. Поляков К.Ю. Язык Python глазами учителя // Информатика, № 9, 2014, с. 4–16.
6. Демоверсия, спецификация, кодификатор ЕГЭ-2012 по информатике [Электронный ресурс] URL: <http://fipi.ru/sites/default/files/document/1408709946/infEGE2012.zip> (дата обращения 06.09.2014).
7. Поляков К.Ю. ЕГЭ-А10: задачи с интервалами // Информатика, № 2, 2013, с. 4–9.
8. Демоверсия, спецификация, кодификатор ЕГЭ-2012 по информатике [Электронный ресурс] URL: <http://fipi.ru/sites/default/files/document/1408709946/infEGE2012.zip> (дата обращения 06.09.2014).
9. Открытый банк заданий ГИА. 11-й класс. Информатика [Электронный ресурс] URL: <http://opengia.ru/subjects/informatics-11/topics/1> (дата обращения 06.09.2014).

журнал

Информатика – Первое сентября

1-е полугодие 2015 года

ПОДПИСКА

на сайте www.1september.ru и в почтовых отделениях РФ



Индекс	Название издания	Периодичность в полугодии	1 месяц		6 месяцев	
			Каталожная цена (руб.)	Подписная цена (руб.)	Каталожная цена (руб.)	Подписная цена (руб.)
Название блока в разделе «Журналы»	ПЕРВОЕ СЕНТЯБРЯ. ЖУРНАЛЫ ИЗДАТЕЛЬСКОГО ДОМА (499)249-31-38					
79066	Информатика – Первое сентября. Бумажная версия С электронными приложениями и презентациями. <i>В июне не выходит.</i> <i>Подписка на июнь не принимается</i> (-) 160 г 64 стр.	5	340.00		1700.00	
12684	Информатика – Первое сентября. Электронная версия на CD (полная копия бумажной версии) <i>В июне не выходит.</i> <i>Подписка на июнь не принимается</i> (-) 75 г	5	118.80		594.00	
сайт 1september.ru	Информатика – Первое сентября. Электронная версия	5	–		–	300.00

Подписку принимают во всех отделениях связи Российской Федерации, включая Крым и Севастополь, а также на сайте www.1september.ru

При оформлении подписки на сайте оплата производится по квитанции в отделении банка или электронными платежами on-line





Урок 1. Робототехника. Вводное занятие

В.В. Тарапата,
преподаватель
информатики
ГБОУ СОШ № 755,
сотрудник
учебно-научного
центра МПГУ,
Москва

► **Образовательная цель:** понимать сущность науки “Робототехника”, знать основные ее задачи. Знать основы алгоритмизации и основные этапы сборки робота. Составлять алгоритмы для исполнителя из набора команд.

Воспитательная цель: воспитание познавательной активности.

Развивающая цель:

1. Развитие речи: обогащение и усложнение словарного запаса.
2. Развитие мышления: умение анализировать, выделять главное, объяснять изученные понятия.

Тип урока: объяснение нового материала.

Трехуровневые задачи урока

1. Понимать сущность науки “Робототехника”.
2. Знать основные правила робототехники, понятие “Робот” (исполнитель), основы алгоритмизации.
3. Уметь формулировать основные задачи робототехники и ее пра-

вила, решать задачи на алгоритмизацию.

Основные понятия

Робототехника, робот, алгоритм, исполнитель, система команд исполнителя.

Структура и ход урока

I этап. Подготовка к восприятию нового материала.

Деятельность учителя

Учитель пишет на доске тему “Робототехника” и рассказывает: “Прежде чем приступить к рассмотрению науки Робототехника, нам необходимо понять, из чего состоит ее название.

Во-первых, слово “робот”. Что же оно означает? Термин “робот”, который придумал в 1920 году писатель, научный фантаст Карел Чапек, происходит от чешского слова “robota”, что означает “тяжелая монотонная работа” или “каторга”.

Первым промышленным роботом стал Unimate, выпущенный в 1961 го-

ду, — это механическая рука, использовавшаяся корпорацией General Motors при производстве автомобилей. Робот выполнял последовательность действий, которая была записана на магнитный барабан.

Итак, *Робот* — это машина, которую можно обучить, т.е. подобно компьютеру запрограммировать (задать ему набор действий, которые он должен выполнять) делать разнообразные виды движений, реагировать на изменения в окружающем мире и выполнять множество видов работ и заданий. Машины, которые выполняют только одну работу и не могут быть переобучены, настоящими роботами не являются, и называют их **автоматами** (примером служат микроволновые печи, кофеварки и т.д.).

Действиями робота всегда управляет микропроцессор, который запрограммирован в соответствии с заданием. Робота всегда можно быстро перепрограммировать на выполнение нового задания.

Он всегда точно следует инструкциям, т.е. выполняет операции по заложенной в него программе.

Запишем: “Робот — это автоматическое устройство для осуществления производственных и других операций по определенной программе (алгоритму)”.

Во-вторых, слово “техника”. Понятие техники охватывает технические изделия, ранее не существовавшие в природе и изготовленные человеком для осуществления какой-либо деятельности, — машины, механизмы, оборудование, аппараты, приспособления, инструменты, приборы и т.д.

Теперь разберемся в самом термине “**Робототехника**”.

Слово “робототехника”, точнее английское “robotics”, было впервые использовано в печати писателем Айзеком Азимовым в научно-фантастическом рассказе “Лжец”, опубликованном в 1941 году. Им же в рассказе “Хоровод” (1942) были удачно сформулированы три закона робототехники — обязательные правила поведения для роботов:

I. Роботу запрещается причинять вред человеку или своим бездействием допустить, чтобы человек был нанесен вред.

II. Робот обязан повиноваться приказам людей, за исключением тех случаев, когда приказы противоречат первому закону робототехники.

III. Робот должен защищать свою жизнь до тех пор, пока такая защита не вступает в противоречие с первым и вторым законами.

Робототехника — это область техники, связанная с разработкой и применением роботов и компьютерных систем управления ими. Существует много типов робототехнических устройств, в том числе роботы-манипуляторы, мобильные роботы, шагающие роботы, средства помощи инвалидам, телеуправляемые и миниатюрные роботы”.

Запишем: “**Робототехника** — это прикладная наука, занимающаяся разработкой автоматизированных систем. Робототехника опирается на такие дисциплины, как механика, физика, электроника, математика и информатика”.

Деятельность учеников

Записывают название темы, слушают, задают вопросы, записывают за учителем.

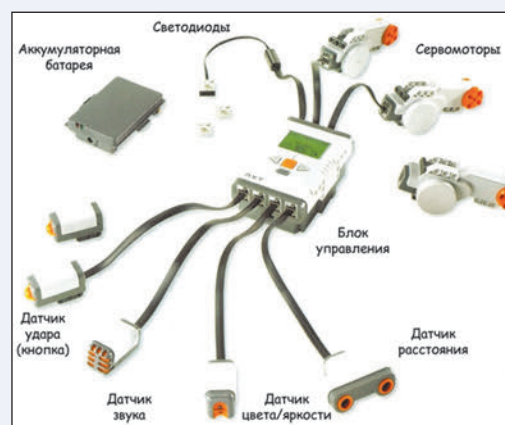
II этап. Изучение нового материала. Основные компоненты набора робототехники. Этапы сборки робота. Решение базовых задач, рассматриваемых в курсе.

1. Набор Lego Mindstorms.

Деятельность учителя

В рамках нашего курса мы будем использовать набор Lego Mindstorms, с помощью которого и будем конструировать роботов. Он состоит из следующих компонентов.

На экран интерактивной доски (экран проектора) выводится следующее изображение.



Это основные компоненты любого робота. Что-то из них может использоваться не в каждом роботе. Это зависит от задачи/задач, решаемых конкретным устройством. Например, если робот не измеряет расстояние до объекта, а реагирует только на свет, то ему точно не понадобится датчик расстояния.

Самым главным элементом на данном изображении является Блок управления. Как мы говорили ранее, все команды робот исполняет с помощью микропроцессора. Иными словами, микропроцессор — это мозг нашего робота. Именно в блоке управления и содержится микропроцессор. Помимо него, блок управления включает в себя память для хранения программ, электроузлы для связи с внешними устройствами (а также Bluetooth для управления с мобильных устройств с помощью специального ПО) и жидкокристаллический дисплей, облегчающий работу с роботом.

Без Блока управления робот работать не будет!

Для работы блока управления требуется аккумуляторная батарея, которую необходимо периодически заряжать от электросети.

Вторым по важности компонентом является сервомотор (в зависимости от набора NXT их количество может варьироваться), подключающийся к блоку управления. Каждый сервомотор имеет вращающийся элемент, который позволяет роботу двигаться. Как и в каких направлениях —

решает проектировщик, добавляя к сервомоторам дополнительные детали из стандартного набора Mindstorms, а также любых наборов Lego. На следующем уроке мы будем собирать нашего первого робота и увидим основные принципы работы сервомоторов.

Далее идут всевозможные датчики.

1. Датчик расстояния. Используется для измерения расстояния до находящегося впереди объекта. Или если реализовано вращение данного датчика, то до объекта, находящегося перед датчиком. В зависимости от того, насколько далеко или близко объект от робота, может выполняться та или иная команда. Самое распространенное использование такого датчика — это объезд препятствий. Как только робот приближается, например, к стенке, датчик подает сигнал об опасном сближении, и робот меняет курс движения. Принцип работы этого датчика основан на посыле ультразвукового сигнала с последующим его отражением от объекта и возврата обратно в датчик. Чем больше время, за которое посланный сигнал вернулся, тем больше расстояние до объекта. И наоборот.

2. Датчик цвета/яркости. В зависимости от используемого набора данный датчик может различать цвета или яркость (оттенки серого). Не стоит расстраиваться, если ваш датчик различает только оттенки серого, так как любой цвет (красный, желтый и т.д.) может быть им также воспринят как один из оттенков серого. Однако это усложняет задачу при программировании робота, так как необходимо заранее знать, какой из оттенков (код оттенка серого) соответствует тому или иному цвету. Это можно узнать вручную с помощью специального ПО, прилагаемого к набору. Обычно такой датчик используется для направления движения робота по цветным полям или в роботе, который собирает кубик Рубика для распознавания цветов.

3. Датчик звука. Реагирует на громкость звука или определенный тембр голоса. Например, можно сделать робота, который будет начинать работу только после распознавания голоса, который необходимо записать на него заранее при программировании. Или же при громком хлопке дверью робот может подавать сигнал “Не хлопайте дверью”. Принцип работы такого датчика ничем не отличается от обычного микрофона.

4. Датчик удара (кнопка). Представляет собой небольшую кнопку и служит для определения столкновения робота с каким-либо препятствием. Например, можно собрать радиоуправляемого робота, который будет обучать аккуратной парковке. Как только будет происходить касание с другим роботом или иным объектом, робот будет подавать сигнал: “Столкновение”. Также данный датчик может служить пусковым механизмом программы. Как, например, лифт не начнет движение, пока мы не нажмем кнопку вызова, так и робот может не начинать выполнение программы, пока мы не нажмем кнопку старта.

И последнее — светодиоды. Используются для подачи светового сигнала, если того требует задача.

Деятельность учеников

Слушают, конспектируют материал, изучают основные компоненты набора, задают вопросы.

2. Этапы сборки робота.

Деятельность учителя

Сборка любого робота состоит из следующих этапов:

1. Постановка задачи — определение среды и области использования робота и действий, которые необходимо выполнить.

2. Проектировка робота — создание ментальной (мысленной) модели робота с подробным описанием действий. Например, если стоит задача объехать препятствия, то необходимо описать габариты робота, расположение необходимых датчиков (в зависимости от того, на каком уровне находятся препятствия и какого они рода), нужную скорость движения, характер движения. При необходимости может быть сделан схематичный чертеж.

3. Создание алгоритма — под конкретную модель находится оптимальный алгоритм, на основе которого будет написана программа для успешного функционирования робота.

4. Сборка робота — первоначальная сборка рабочего механизма, отвечающего требованиям модели и алгоритму.

5. Программирование — написание программы, отладка и загрузка ее в блок управления.

6. Первоначальный тест — запуск программы и проверка функционирования всех систем робота в различных ситуациях в условиях задачи. Выявление неисправностей.

7. Корректировка — исправление технических и/или программных ошибок, влекущих некорректное функционирование робота.

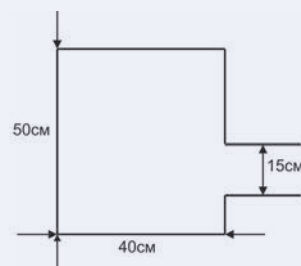
Деятельность учеников

Слушают, записывают этапы сборки робота, задают вопросы.

3. Решение задачи на проектирование с учителем.

Деятельность учителя

На экран интерактивной доски (экран проектора) выводится изображение:



Задача: Робот находится внутри данного помещения. Его начальное местоположение не известно. Требуется создать модель, способную пройти через коридор справа.

Возможное решение: габариты робота не должны превышать 15 сантиметров в ширине (возможность пройти в коридор) и 30 сантиметров в длину (возможность развернуться при необходимости). В головной части робота должен располагаться датчик расстояния для своевременной остановки перед стеной. Также робот должен быть оснащен датчиком удара в хвостовой части для упрощения маневров, связанных с задним ходом. Для большей мобильности подвижная часть оснащается двумя сервомоторами, на которые крепятся большие приводные колеса, располагающиеся в передней части, и одно малое поворотное колесо в задней. Общих осей они имеют не должны.

Деятельность учеников

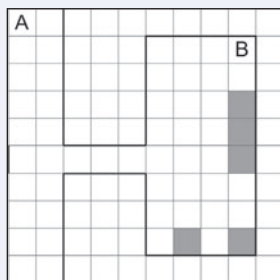
Записывают условие задачи, решают вместе с учителем, предлагают свои варианты решения, анализируют верные и неверные идеи вместе с учителем.

4. Решение задачи на построение алгоритма для исполнителя.

Деятельность учителя

Часто робота называют “исполнителем”, так как он выполняет какие-то заранее определенные программистом команды. Набор таких команд называют “системой команд исполнителя” (СКИ). Решим задачу, в которой заранее определена СКИ.

На экран интерактивной доски (экран проектора) выводится изображение.



Задача:

Исполнитель “Робот” знает следующие команды:

- 1) Идти вверх;
- 2) Идти вниз;
- 3) Идти вправо;
- 4) Идти влево;
- 5) Если сверху свободно/не свободно;
- 6) Если снизу свободно/не свободно;
- 7) Если справа свободно/не свободно;
- 8) Если слева свободно/не свободно;
- 9) Красить.

Необходимо составить алгоритм, в результате исполнения которого Робот переместится из ква-

драта, обозначенного “А”, в квадрат, обозначенный “В”, при этом закрасив клетки, отмеченные серым цветом, и оставшись целым.

“Шаг” робота составляет одну клетку. Жирным шрифтом отмечены стены. Если робот пойдет в стену, он разрушится.

Возможное решение:

1. Если слева свободно, идти вниз.
2. Если снизу свободно, идти вправо.
3. Если снизу не свободно, идти вправо.
4. Если снизу свободно, идти вниз.
5. Идти вправо.
6. Красить.
7. Если справа свободно, идти вправо.
8. Красить.
9. Идти вверх.
10. Идти вверх.
11. Идти вверх.
12. Красить.
13. Идти вверх.
14. Красить.
15. Идти вверх.
16. Красить.
17. Если сверху свободно, идти вверх.

Деятельность учеников

Записывают условие задачи, решают вместе с учителем, предлагают свои варианты решения, анализируют верные и неверные идеи вместе с учителем.

III этап. Закрепление изученного материала.

Деятельность учителя

Учитель еще раз обзорно повторяет все изученное на уроке. Обращает внимание на важные аспекты и основные понятия. Спрашивает о наличии вопросов у учеников. При необходимости отвечает.

Деятельность учеников

Слушают, при необходимости задают вопросы.

IV этап. Домашнее задание.

Деятельность учителя

Повторить изученный на уроке материал, подготовиться к распределительному тесту. По желанию на дополнительную оценку разработать аналогичные задания на проектирование и/или построение алгоритма для исполнителя, предложить решение.

Деятельность учеников

Записывают задание.

Урок 2. Распределительный тест

Цель урока: Выявить у учеников способности к тому или иному виду деятельности для распределения ролей в последующей групповой работе в ходе всего курса.

Структура и ход урока

I этап. Активизация ранее изученного материала.

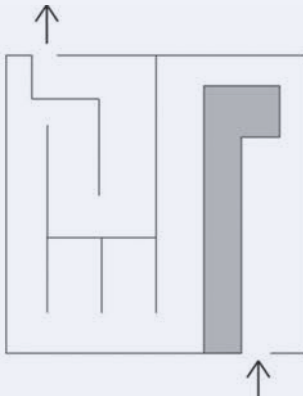
Деятельность учителя	Деятельность учеников
Что такое робот? Что такое робототехника? Какие основные компоненты включает в себя используемый нами набор Lego Mindstorms? Каковы этапы сборки робота?	Вместе с учителем вспоминают основные понятия вводного занятия, отвечают на вопросы. Сдают выполненные работы, которые были даны по желанию.

II этап. Подготовка к выполнению теста.

Деятельность учителя	Деятельность учеников
<p>Тест будет состоять из трех различных заданий, которые помогут выявить те или иные способности (проектирование, построение алгоритмов и программирование, сборка) у каждого из вас.</p> <p>Задачи выполняются на время. Результаты фиксируются учителем в таблице. Торопиться не нужно, работаете в своем обычном темпе.</p> <p>Задание считается выполненным, если решение отвечает всем его требованиям.</p> <p>На первые два задания ограничение по времени составляет 20 минут. На третье — 30 минут. По истечении этого времени все работы сдаются в том виде, в котором они есть. Если вы считаете, что готовы сдать задание досрочно, то его можно сдать, и при возможности учитель проверит правильность его выполнения сразу, время выполнения будет зафиксировано как время сдачи задания. Если задание выполнено неправильно, оно возвращается назад.</p> <p>Каждое задание выполняется индивидуально.</p> <p>По результатам данного теста, которые будут оглашены на следующем занятии, вы будете разбиты на группы по четыре человека:</p> <ol style="list-style-type: none"> 1) Один проектировщик, создающий ментальную модель робота под конкретную задачу (при необходимости для модели может быть составлено письменное описание и чертеж), делающий заказ программисту. 2) Один программист, занимающийся первоначальным построением необходимых алгоритмов для функционирования робота в условиях конкретной задачи, делающий заказ инженерам и программирующий робота на финальной стадии, после его сборки. 3) Два инженера, выполняющие совместный заказ проектировщика и программиста, собирая робота из доступных комплектующих и материалов набора LEGO Mindstorms. 	Слушают, задают вопросы.

III этап. Распределительный тест.

Задание №1. “Проектирование”.

Деятельность учителя	Деятельность учеников
<p>Задание:</p> <p>Разработать собственную модель мобильного робота, проходящего классический лабиринт по правилу левой (правой) стенки, зная, что доступны сенсоры и комплектующие, рассмотренные на вводном занятии.</p> <p>На экран интерактивной доски (экран проектора) выводится изображение.</p> 	Слушают, получают задания, приступают к выполнению.

Деятельность учителя

Указания:

Описать комплектующие, из которых состоит робот; используемые им сенсоры (способы их применения), например: «Сенсор “Расстояние до объекта” устанавливается по направлению движения робота. Измеряет расстояние до впереди идущей стенки. Как только это расстояние становится достаточно мало, робот получает указание выполнить поворот».

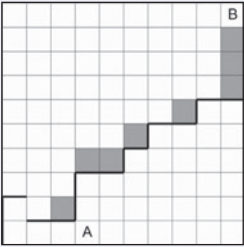
Одна из конструкций закрашена серым цветом, так как она также является стеной, а не пустым пространством.

Каждому ученику выдается аналогичное изображение, напечатанное на бумаге, с заданием и указаниями.

При необходимости на экран интерактивной доски (экран проектора) может быть выведено “Приложение 2”.

Засекается время на выполнение задания, не более 20 минут.

Задание №2. “Алгоритмизация”.

Деятельность учителя	Деятельность учеников
<p>Задание: Исполнитель “Робот” знает следующие команды:</p> <ol style="list-style-type: none"> 1) Идти вверх; 2) Идти вниз; 3) Идти вправо; 4) Идти влево; 5) Если сверху свободно/не свободно; 6) Если снизу свободно/не свободно; 7) Если справа свободно/не свободно; 8) Если слева свободно/не свободно; 9) Красить. <p>Необходимо составить алгоритм, в результате исполнения которого Робот переместится из квадрата, обозначенного “А”, в квадрат, обозначенный “В”, при этом закрасив клетки, отмеченные серым цветом, и оставшись целым.</p> <p>“Шаг” робота составляет одну клетку. Жирным слоем отмечены стены. Если робот пойдет в стену, он разрушится.</p> <p>На экран интерактивной доски (экран проектора) выводится изображение.</p> <div style="text-align: center;">  </div> <p>Каждому ученику выдается аналогичное изображение, напечатанное на бумаге, с заданием и указаниями.</p> <p>Засекается время на выполнение задания, не более 20 минут.</p>	<p>Слушают, получают задания, приступают к выполнению.</p>

Задание №3 “Сборка”.

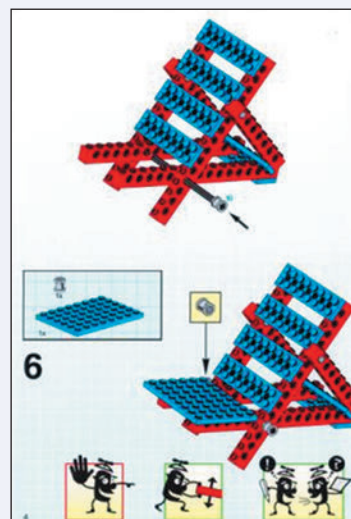
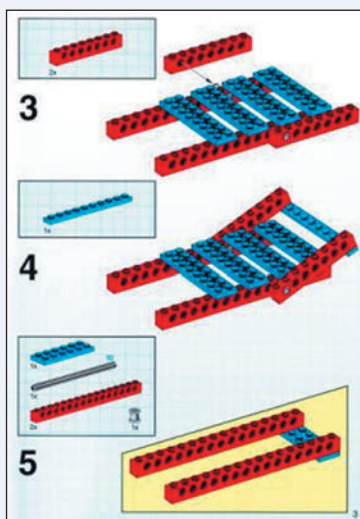
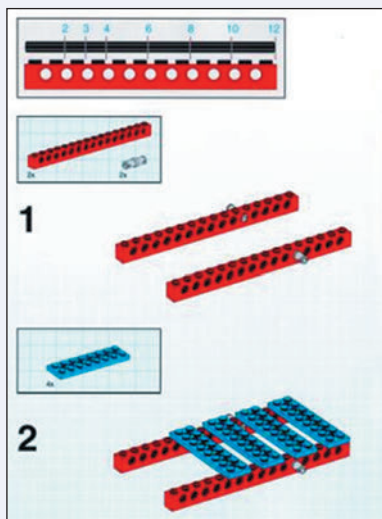
Деятельность учителя	Деятельность учеников
<p>Учащимся выдается дополнительный набор LEGO®, с помощью которого они собирают одну из предложенных учителем моделей (по собственному усмотрению). Для примера выберем дополнительный набор Lego Educational Division.</p> <p>Задание: Собрать модель шезлонга.</p> <div style="text-align: center;">  </div>	<p>Слушают, получают задания, приступают к выполнению.</p>

Деятельность учителя

Ученикам выдается инструкция по сборке. При необходимости она может быть выведена на экран интерактивной доски (экран проектора).



Засекается время на выполнение задания, не более 30 минут.



Этап IV. Окончание теста.

Деятельность учителя	Деятельность учеников
<p>Учитель объявляет о завершении теста. Могут быть оглашены предварительные результаты, если таковые имеются. Можно выделить лидеров по тем или иным заданиям.</p> <p>Отвечает на возникшие вопросы.</p>	<p>Слушают, при необходимости задают вопросы.</p>



Урок 3. Сборка подвижной платформы

Цель урока: Собрать основную подвижную платформу для изучения базовых команд движения робота и последующего рассмотрения работы датчиков (сенсоров).

Структура и ход урока

I этап. Объявление результатов распределительного теста и разбиение на группы.

Деятельность учителя	Деятельность учеников
<p>Объявляет результаты распределительного теста и разбивает учеников на группы из четырех человек (при невозможности такого разбиения учитель действует по ситуации, добавляя или убавляя из групп людей, заранее проанализировав ту или иную перестановку; также учитываются пожелания учащихся). В таком составе учащиеся будут работать на протяжении всего курса.</p> <p>Также можно оценить и рассмотреть выполненные по желанию домашние задания, если таковые имеются.</p>	<p>Слушают, разбиваются на группы, участвуют в обсуждении.</p>

II этап. Подготовка к сборке подвижной платформы. Первая работа с набором Lego Mindstorms.

Деятельность учителя	Деятельность учеников
<p>Выдает наборы Lego Mindstorms на каждую группу.</p> <p>Сегодня мы соберем основную подвижную платформу, на основе которой изучим действие сервомоторов и всех датчиков (сенсоров). Сегодня же напишем первую простейшую программу.</p>	<p>Получают наборы, открывают их, достают инструкции по сборке.</p>

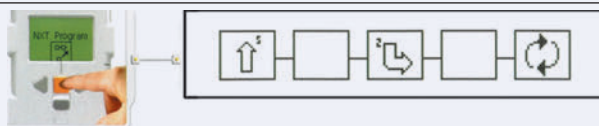
III этап. Сборка подвижной платформы.

Деятельность учителя	Деятельность учеников
<p>Перед тем как ученики приступят к сборке, учитель выводит на экран интерактивной доски “Приложение 4”, в котором показано, как соединять сервомоторы и датчики с блоком управления, и какие гнезда отведены под те или иные элементы. Рассказывает об этом.</p>  <p>Гнезда А, В, С предназначены для подключения сервомоторов и/или светодиодов.</p> <p>Гнезда 1, 2, 3, 4 предназначены для подключения датчиков удара, звука, света/цвета, расстояния соответственно.</p> <p>Затем учитель следит за аккуратностью работы учеников с наборами и ходом сборки в соответствии с инструкцией. При необходимости можно воспользоваться данным руководством, в котором наиболее полно и поэтапно расписаны этапы сборки для конкретного набора Lego Mindstorms второго поколения (с тремя сервомоторами): см. файл Сборка подвижной платформы.docx в электронных приложениях к номеру.</p>	<p>Слушают указания, приступают к сборке платформы, при необходимости задают вопросы.</p>

IV этап. Первая программа.

Деятельность учителя	Деятельность учеников
<p>После того как подвижная платформа собрана, нужно ее протестировать. Заодно посмотрим, как с помощью моноблока можно писать простейшие программы.</p> <p>Для этого необходимо включить моноблок, выбрать пункт “NXT Program”, с помощью кнопок выбора нужно задать программу движения вперед и разворота направо. Программу запустить циклически (см. листинг 1.1, приложение 5).</p> <p>Обратить внимание на то, что остаются пустые квадратики — это места для сенсорных условий, которые подробно рассматриваются при изучении сенсоров (датчиков).</p>	<p>Слушают, при необходимости задают вопросы, следуют инструкциям учителя, запускают программу.</p>

Деятельность учителя



Листинг 1.1

V этап. Закрепление изученного материала.

Деятельность учителя	Деятельность учеников
Учитель предлагает попробовать написать собственные простейшие программы движения платформы и продемонстрировать их работоспособность.	Слушают, пишут собственные простейшие программы с помощью блока управления, задают вопросы.

VI этап. Завершение работы с наборами.

Деятельность учителя	Деятельность учеников
Аккуратно собираем наборы, укладываем все не использованные детали в соответствующие гнезда. Проверяем уровень заряда аккумуляторной батареи, при необходимости ставим на подзарядку. Наборы ставим в шкафы на соответствующее место, сверху наборов ставим собранные платформы.	Собирают наборы, при необходимости подключают блоки управления на подзарядку с помощью учителя, убирают наборы и собранные модели в шкаф.

Урок 4. Изучение работы сенсоров (датчиков)

Цель урока: Изучить принципы работы датчиков звука, расстояния, света/цвета и удара.

Структура и ход урока

I этап. Подготовка к изучению сенсоров.


Деятельность учителя	Деятельность учеников
<p>Достаем наборы и собранные подвижные платформы. Находим инструкцию и необходимые нам датчики. В дальнейшем будем называть их сенсорами.</p> <p>Сегодня мы изучим принципы работы всех доступных нам датчиков.</p> <p>На экран интерактивной доски (экран проектора) выводится следующее изображение.</p> 	Слушают, достают наборы соответственно своей группе, находят сенсоры.
Найдите все датчики и выложите их перед собой.	

II этап. Сенсор звука.

1. Подсоединение сенсора.

Деятельность учителя	Деятельность учеников
<p>В соответствии с инструкцией к вашему набору подсоедините сенсор звука.</p> <p>Учитель следит за аккуратностью работы учеников с наборами и ходом подсоединения в соответствии с инструкцией. При необходимости можно воспользоваться данным руководством, в котором наиболее полно и поэтапно расписаны этапы подсоединения сенсоров для конкретного набора Lego Mindstorms второго поколения (с тремя сервомоторами): Подсоединение сенсоров.</p>	<p>Слушают указания, приступают к подсоединению сенсора звука, при необходимости задают вопросы.</p>

2. Программа-тест сенсора звука.

Деятельность учителя	Деятельность учеников
<p>Включить моноблок, выбрать пункт “NXT Program”, с помощью кнопок выбора нужно задать программу поворота направо, затем в окне сенсорного условия (которое в первой программе оставалось пустым) выбрать иконку микрофона, затем разворот налево и снова иконку микрофона. Программу запустить циклически (см. листинг 1.1, приложение 5).</p> <div style="text-align: center;">  <p>Листинг 1.1</p> </div> <p>В результате выполнения данной программы робот будет поворачивать направо до тех пор, пока не “услышит” громкий звук (например, хлопок ладоней), после этого он приступит к развороту налево до тех пор, пока снова не “услышит” громкий звук. Так как программа запущена циклически, после второго хлопка она начнет исполняться заново, то есть робот снова приступит к повороту направо.</p>	<p>Слушают, при необходимости задают вопросы, следуют инструкциям учителя, запускают программу.</p>

3. Самостоятельное тестирование сенсора звука.

Деятельность учителя	Деятельность учеников
<p>Предлагает ученикам в течение нескольких минут попробовать написать другие простейшие программы, использующие сенсор звука.</p>	<p>Слушают, пишут собственные простейшие программы с помощью блока управления, задают вопросы.</p>

III этап. Сенсор расстояния (ультразвуковой датчик).

1. Подсоединение сенсора.

Деятельность учителя	Деятельность учеников
<p>В соответствии с инструкцией к вашему набору подсоедините сенсор расстояния.</p> <p>Учитель следит за аккуратностью работы учеников с наборами и ходом подсоединения в соответствии с инструкцией. При необходимости можно воспользоваться данным руководством, в котором наиболее полно и поэтапно расписаны этапы подсоединения сенсоров для конкретного набора Lego Mindstorms второго поколения (с тремя сервомоторами): см. файл Сенсоры (датчики).docx в электронных приложениях к номеру.</p>	<p>Слушают указания, приступают к подсоединению сенсора расстояния, при необходимости задают вопросы.</p>

2. Программа-тест сенсора расстояния.

Деятельность учителя	Деятельность учеников
<p>Включить моноблок, выбрать пункт “NXT Program”, с помощью кнопок выбора нужно задать программу движения вперед, затем в окне сенсорного условия выбрать иконку антенны, затем разворот налево и снова иконку антенны. Программу запустить циклически (см. листинг 2.1, приложение 5).</p>	<p>Слушают, при необходимости задают вопросы, следуют инструкциям учителя, запускают программу.</p>

Деятельность учителя



Листинг 2.1

В результате выполнения данной программы робот будет ехать вперед до тех пор, пока не “увидит” впереди себя препятствие (например, стена, рука, книга), после этого он приступит к развороту налево до тех пор, пока снова не “увидит” препятствие. Так как программа запущена циклически, после второго препятствия она начнет исполняться заново, то есть робот снова приступит к движению вперед.

3. Самостоятельное тестирование сенсора расстояния.

Деятельность учителя	Деятельность учеников
Предлагает ученикам в течение нескольких минут попробовать написать другие простейшие программы, использующие сенсор расстояния.	Слушают, пишут собственные простейшие программы с помощью блока управления, задают вопросы.

IV этап. Сенсор цвета/света.

1. Подсоединение сенсора.

Деятельность учителя	Деятельность учеников
В соответствии с инструкцией к вашему набору подсоедините сенсор цвета/света. Учитель следит за аккуратностью работы учеников с наборами и ходом подсоединения в соответствии с инструкцией. При необходимости можно воспользоваться данным руководством, в котором наиболее полно и поэтапно расписаны этапы подсоединения сенсоров для конкретного набора Lego Mindstorms второго поколения (с тремя сервомоторами): см. файл Сенсоры (датчики).docx в электронных приложениях к номеру.	Слушают указания, приступают к подсоединению сенсора цвета/света, при необходимости задают вопросы.

2. Программа-тест сенсора цвета/света.

Деятельность учителя	Деятельность учеников
<p>Включить моноблок, выбрать пункт “NXT Program”, с помощью кнопок выбора нужно задать программу поворота налево, затем в окне сенсорного условия выбрать иконку закрашенного круга, затем поворот направо и иконку пустого круга. Программу запустить циклически (см. листинг 3.1, приложение 5).</p> <p>Листинг 3.1</p> <p>В результате выполнения данной программы робот будет поворачивать налево до тех пор, пока не “увидит” что-то темное (так как выбран закрашенный круг; например, темную/черную поверхность), после этого он приступит к повороту направо до тех пор, пока не “увидит” что-то светлое (так как выбран пустой круг; например, светлую/белую поверхность). Так как программа запущена циклически, после светлой поверхности она начнет исполняться заново, то есть робот снова приступит к повороту налево.</p> <p>Для корректного функционирования данной программы лучше всего заранее подготовить контрастную тестовую поверхность, одна часть которой должна быть темной (черной), а вторая — светлой (белой). При правильной настройке и достаточном цветовом контрасте поверхности робот будет ехать ровно по “стыку” темной и светлой поверхностей.</p> <p>Если расположить сенсор иначе (например, так, чтобы он “смотрел” в направлении движения робота), то можно аналогичным образом протестировать реакцию на степень освещения. Например, запускать робота из освещенного пространства в темное и обратно, а также задавать начало работы по световому сигналу (например, по яркому свету фонарика, направленному на сенсор).</p>	Слушают, при необходимости задают вопросы, следуют инструкциям учителя, запускают программу.

3. Самостоятельное тестирование сенсора цвета/света.


Деятельность учителя	Деятельность учеников
Предлагает ученикам в течение нескольких минут попробовать написать другие простейшие программы, использующие сенсор цвета/света.	Слушают, пишут собственные простейшие программы с помощью блока управления, задают вопросы.

V этап. Сенсор удара (кнопка).

1. Подсоединение сенсора.

Деятельность учителя	Деятельность учеников
В соответствии с инструкцией к вашему набору подсоедините сенсор удара. Учитель следит за аккуратностью работы учеников с наборами и ходом подсоединения в соответствии с инструкцией. При необходимости можно воспользоваться данным руководством, в котором наиболее полно и поэтапно расписаны этапы подсоединения сенсоров для конкретного набора Lego Mindstorms второго поколения (с тремя сервомоторами): см. файл Сенсоры (датчики).docx в электронных приложениях к номеру.	Слушают указания, приступают к подсоединению сенсора удара, при необходимости задают вопросы.

2. Программа-тест сенсора цвета/света.

Деятельность учителя	Деятельность учеников
Включить моноблок, выбрать пункт “NXT Program”, с помощью кнопок выбора нужно задать программу движения назад, затем в окне сенсорного условия выбрать иконку пальца, нажимающего кнопку, затем движение вперед в течение пяти секунд, а второе сенсорное условие оставить пустым. Программу запустить циклически (см. листинг 4.1, приложение 5).	Слушают, при необходимости задают вопросы, следуют инструкциям учителя, запускают программу.
 <p>Листинг 4.1</p>	
В результате выполнения данной программы робот будет двигаться назад до тех пор, пока не столкнется с каким-либо препятствием, после этого он приступит к движению вперед в течение пяти секунд. Так как программа запущена циклически, после завершения пяти секунд робот снова начнет двигаться назад.	

3. Самостоятельное тестирование сенсора удара.

Деятельность учителя	Деятельность учеников
Предлагает ученикам в течение нескольких минут попробовать написать другие простейшие программы, использующие сенсор удара.	Слушают, пишут собственные простейшие программы с помощью блока управления, задают вопросы.

VI этап. Окончание работы с наборами. Закрепление изученного материала.

Деятельность учителя	Деятельность учеников
Собираем наборы, ставим в шкаф, собранные модели ставим сверху ваших наборов. Мы познакомились с принципами работы всех доступных сенсоров. На следующем занятии я расскажу вам о соревнованиях робомашин и творческих роботов (например, робот, взбирающийся по канату или собирающий кубик Рубика), а также об основных правилах сборки роботов. Как вы видите, вашу фантазию могут ограничить только технические возможности роботов. По какому пути идти дальше, мы решим с вами вместе на следующем занятии.	Собирают наборы, убирают в шкафы вместе с моделями, слушают, задают вопросы.

Урок 5. Правила робототехники.

Выбор курса

Образовательная цель: понимать правила робототехники. Уметь следовать им.

Воспитательная цель: воспитание познавательной активности, ответственности перед коллективом за результаты труда.

Развивающая цель:

1. Развитие речи: обогащение и усложнение словарного запаса.
2. Развитие мышления: умение анализировать, выделять главное, объяснять изученные понятия.

Тип урока: объяснение нового материала.

Трехуровневые задачи урока	Основные понятия
1. Понимать сущность правил робототехники.	Правила робототехники.
2. Уметь применять правила на всех этапах сборки робота.	
3. Знать, как и где правилами можно пренебречь.	

Структура и ход урока

I этап. Активизация ранее изученного материала.

Деятельность учителя	Деятельность учеников
Какие правила робототехники сформулировал Айзек Азимов? Возможно ли применение данных правил в рамках набора Lego Mindstorms?	Вместе с учителем вспоминают основные правила робототехники, анализируют, отвечают и задают вопросы.

II этап. Подготовка к восприятию нового материала.

Деятельность учителя	Деятельность учеников
Три правила, сформулированные Айзеком Азимовым, — это базовые правила робототехники. К сожалению, использование только этих трех правил не позволяет эффективно заниматься робототехникой. Поэтому важно узнать еще несколько полезных правил и советов для оптимизации своего труда.	Слушают, задают вопросы.

III этап. Изучение нового материала. Правила робототехники.

Деятельность учителя	Деятельность учеников
<ol style="list-style-type: none"> 1. Больше деталей, следовательно, больше вес робота, а следовательно, меньше скорость. Данным правилом можно пренебречь в том случае, если стоит задача — построение “робота-тяжеловеса” для сдвига объектов (грузы, другие роботы и т.д.). 2. Перегруз сервомоторов может привести к неисправностям. Не нужно гнаться за скоростью в ущерб аппаратной составляющей. Ведь лучше медленный, но рабочий робот, чем быстрый и неисправный. 3. Робота необходимо содержать в чистоте. Грязь, вода, пыль губительны для него. 4. Перед началом и по окончании работы следует всегда проверять уровень заряда батареи. При необходимости нужно ставить робота на подзарядку. 5. В целях экономии заряда батареи следует своевременно выключать робота, если он не производит никаких действий. 6. Подключать и отключать сенсоры, сервомоторы, светодиоды, а также модифицировать робота с помощью дополнительных деталей следует только в выключенном состоянии. В противном случае это может повлечь техническую или программную ошибку, а затем неисправность. 7. На этапе первичного тестирования следует выявить все ошибки в работе робота. В противном случае, например на соревнованиях, это сыграет с вами злую шутку. 8. Больше площадь соприкосновения с поверхностью — больше сцепление с ней, а следовательно, больше устойчивость и тяговые характеристики робота. 9. Увеличить скорость вращения колес робота можно с помощью шестеренок. Большая шестеренка, подсоединенная к сервомотору, вращающая малую, подсоединенную к колесу, дает большую скорость. Пока большая шестеренка делает один оборот, малая делает несколько. Так же можно сделать и наоборот, вращение от малой шестеренки передается на большую, это увеличивает проходимость робота. 	Слушают, записывают правила, задают вопросы.

IV этап. Выбор курса.

Деятельность учителя	Деятельность учеников
<p>Цель наших занятий — выйти на соревнования роботов. Существует несколько этапов: от окружного до международного.</p> <p>Основные соревнования проходят в секторе робомашин, где от каждой команды требуются машины трех разных типов: для участия в гонке; для участия в перетягивании каната; для сортировки шариков по цветам.</p> <p>Также существуют соревнования в творческих замыслах, но они требуют высокой оригинальности конструкций и идей, а значит, в разы сложнее.</p> <p>Предлагаю вам выбрать курс, по которому мы пойдем.</p> <p>Если это соревнования робомашин, то каждая из наших команд собирает свою машину для каждого этапа соревнований, и мы устраиваем свои собственные соревнования. Какая из машин побеждает в данной категории, та и остается. Остальные разбираются, и начинается сборка машины для следующей категории. Затем мы подаем заявку на участие в соревнованиях и участвуем в них.</p> <p>Либо творческий путь, который, собственно говоря, названием все о себе говорит.</p>	<p>Слушают, задают вопросы, голосуют.</p>

V этап. Окончание работы.

Деятельность учителя	Деятельность учеников
<p>Собираем наборы, ставим в шкаф, собранные модели ставим сверху ваших наборов.</p>	<p>Собирают наборы, убирают в шкафы вместе с моделями.</p>

VI этап. Домашнее задание.

Деятельность учителя	Деятельность учеников
<p>В зависимости от выбранного пути дается задание начать проектировку того или иного робота.</p>	<p>Получают карточки и записывают задание.</p>

Далее работа данного курса сводится к творческой и самостоятельной деятельности учащихся. Учитель выступает в роли помощника или компаньона в преодолении трудностей. При необходимости изучается стандартное ПО написания программ для роботов. Также может быть изучен один из более функциональных, но оттого и сложных языков программирования, используемых Lego Mindstorms.

В ходе творческой работы часто возникают ситуации, связанные со знаниями информатики и физики, что позволяет данному курсу мощно реализовывать механизм рефлексии и активизации познавательного интереса к данным областям науки.



МДМ Банк

Верное решение!

Кредит наличными для учителей*

- от **15%** годовых в рублях
- до **2 500 000** рублей

* Предложение распространяется на сотрудников государственных образовательных учреждений. Ставка от 15% до 25% годовых (устанавливается банком индивидуально в зависимости от платежеспособности, категории клиента, срока и суммы кредита). Сумма кредита — от 60 тыс. до 2,5 млн руб. Срок — от 1 до 5 лет. Без комиссий за предоставление, обслуживание и досрочное погашение кредита, получение наличных денежных средств через кассу или банкомат банка. Комиссия при проведении операции через кассу или банкомат стороннего банка (без учета комиссий стороннего банка) — 1% от суммы операции, мин. 100 руб. Предусмотрены штрафы: при допущении просроченной задолженности - неустойка в размере 0,05% от суммы просроченной задолженности за каждый день просрочки; при несоблюдении условий договора залога — в размере 5000 рублей. Прочие условия предоставления кредита — по тел. 8-800-2003-700 или в офисах банка. Предложение действительно на 09.09.2014 г. ОАО «МДМ Банк». Ген. лицензия ЦБ РФ № 323 от 05.12.2012 г.



Резинава

8 800 2003 700 www.mdm.ru



Фокусы с “информатическим” секретом

Д.М. Златопольский,
Москва

► В статье описаны 10 фокусов, связанных с десятичными системами счисления, а также программы и листы электронных таблиц, с помощью которых можно демонстрировать фокусы на компьютере. Ряд фокусов описывается впервые.

1. Отгадывание числа

Имеется достаточно известная игра на отгадывание задуманного числа, в которой один из участников игры загадывает целое число, например, меньшее 100. Второй участник должен отгадать это число, делая несколько попыток. В случае несовпадения названного и задуманного чисел первый участник сообщает, какое из них больше, после чего делается следующая попытка — и т.д. до отгадывания. Вы, конечно, уже догадались, как должен действовать второй участник — в первой попытке он должен назвать число 50, потом,

в зависимости от ответа, — 25 или 75 и т.п. Как известно, такой принцип, позволяющий отгадать число за минимально возможное количество попыток, называется “методом бинарного поиска” (“деления интервала поиска пополам”).

А если на вопросы можно отвечать только “да” или “нет”, а вопросы типа “Задуманное число больше ...?”, в которых фигурируют слова “больше”, “больше либо равно”, “меньше”, “не меньше”, “превышает” и т.п., а также вопросы типа “Задуманное число находится в интервале от ... до ... ?” задавать нельзя? Как отгадать задуманное число в таком случае, например, целое число из интервала от 600 до 1000 включительно, задав не более 10 вопросов?

Одна из возможных серий вопросов, заведомо приводящая к успеху, такова.

1-й вопрос: “Разделите задуманное число на 2. Является ли единица

остатком при таком делении?”. Если ответ *да*, то запишем 1, если *нет* — 0 (иначе говоря, мы запишем остаток от деления задуманного числа на 2).

2-й вопрос: “Разделите на 2 то частное, которое получилось при первом делении. Является ли единица остатком при таком делении?”. Снова при ответе *да* запишем нуль, а при ответе *нет* — единицу.

Каждый следующий вопрос будем составлять по тому же самому образцу, то есть так: “Разделите на 2 то частное, которое получилось при предыдущем делении. Является ли единица остатком при таком делении?” Всякий раз мы пишем ноль при положительном ответе и единицу при отрицательном.

Повторив эту процедуру 10 раз, мы получим 10 цифр, каждая из которых есть ноль или единица.

После ответов на все вопросы записанные цифры надо расположить в обратном порядке — получится двоичное число, соответствующее задуманному десятичному (оно будет 10-значным). Действительно, система наших вопросов воспроизводит ту самую процедуру, с помощью которой делается перевод некоторого числа в двоичную систему. При этом десяти вопросов достаточно, потому что каждое число от 500 до 1000 записывается в двоичной системе с помощью не более чем десяти знаков.

Переведя полученное число в десятичную систему счисления, получим задуманное число.

Если считать, что задуманное число уже заранее переведено в двоичную систему, то система вопросов, с помощью которой его можно узнать: нужно о каждой его цифре спросить, равна она нулю или нет. Для этого можно задать такие вопросы:

1. “Является ли единица крайней справа цифрой двоичного числа, соответствующего задуманному десятичному числу?”

2. “Является ли единица второй справа цифрой двоичного числа, соответствующего задуманному десятичному числу?”

...

10. “Является ли единица десятой справа цифрой двоичного числа, соответствующего задуманному десятичному числу?”

2. Волшебная таблица

Посмотрите на таблицу, показанную на рис. 1.

С помощью этой таблицы можно отгадать любое задуманное положительное число, не большее 31, если задумавший число скажет, в каких столбцах оно встречается (номера столбцов приведены в первой строке таблицы и выделены полужирным начертанием). Например, если задумано число 22, то по названным номерам столбцов (5, 3 и 2) его можно вычислить следующим образом:

$$2^{5-1} + 2^{3-1} + 2^{2-1} = 16 + 4 + 2.$$

Секрет этой “волшебной” таблицы в том, что в таблице в первом столбце справа выписаны те десятичные числа, которым в двоичной системе

счисления соответствуют числа, оканчивающиеся на 1, во втором — числа, у которых в двоичном представлении вторая от конца цифра равна 1, в третьем — десятичные числа, которым в двоичной системе соответствуют числа, имеющие 1 в третьем справа разряде, и т.д. Так, число 22 в двоичной системе счисления имеет вид 10110, и записано это число в пятом, третьем и втором столбцах. Если вспомнить правило перевода чисел из двоичной системы в десятичную, то по записи 10110 соответствующее десятичное число можно получить так: $2^4 + 2^2 + 2^1$ (это так называемая “развернутая запись числа”), или, “привязываясь” к нашей нумерации столбцов таблицы, — $2^{5-1} + 2^{3-1} + 2^{2-1}$.

5	4	3	2	1
16	8	4	2	1
17	9	5	3	3
18	10	6	6	5
19	11	7	7	7
20	12	12	10	9
21	13	13	11	11
22	14	14	14	13
23	15	15	15	15
24	24	20	18	17
25	25	21	19	19
26	26	22	22	21
27	27	23	23	23
28	28	28	26	25
29	29	29	27	27
30	30	30	30	29
31	31	31	31	31

Рис. 1

Можно также демонстрировать аналогичный фокус, отгадывая возраст учащихся, их день и месяц рождения и т.п.

3. Семь табличек

Подготовьте семь табличек (см. рис. 2–8 на с. 28).

С их помощью можно отгадать любое число от 1 до 127, если задумавший его скажет, в каких таблицах (они имеют номера от 1 до 7¹) это число встречается².

Запишем каждое из чисел от 1 до 127 в двоичной системе. Каждое из этих чисел содержит в двоичной записи не более семи цифр (в частности, $127_{10} = 1111111_2$). Впишем данное число A в таблицу с номером k ($k = 1, 2, \dots, 7$), если в его двоичной записи на k -м, считая справа налево, месте стоит единица, и не будем его туда вписывать, если в его двоичной записи на k -м месте

¹ Номер таблицы на рис. 2–8 на с. 28 приведен в ее нижней части.

² Когда вы будете демонстрировать кому-то этот фокус, можете также предложить передать вам те карточки, на которых имеется задуманное число.

1	3	5	7	9	11	13	15
17	19	21	23	25	27	29	31
33	35	37	39	41	43	45	47
49	51	53	55	57	59	61	63
65	67	69	71	73	75	77	79
81	83	85	87	89	91	93	95
97	99	101	103	105	107	109	111
113	115	117	119	121	123	125	127

1
Рис. 2

2	3	6	7	10	11	14	15
18	19	22	23	26	27	30	31
34	35	38	39	42	43	46	47
50	51	54	55	58	59	62	63
66	67	70	71	74	75	78	79
82	83	86	87	90	91	94	95
98	99	102	103	106	107	110	111
114	115	118	119	122	123	126	127

2
Рис. 3

4	5	6	7	12	13	14	15
20	21	22	23	28	29	30	31
36	37	38	39	44	45	46	47
52	53	54	55	60	61	62	63
68	69	70	71	76	77	78	79
84	85	86	87	92	93	94	95
100	101	102	103	108	109	110	111
116	117	118	119	124	125	126	127

3
Рис. 4

стоит ноль. Например, число 57, которое в двоичной системе записывается как 0111001, должно быть занесено в первую, четвертую, пятую и шестую таблички, число 1 — только в первую, число 127 — во все семь табличек и т.д. Таким образом, говоря, в каких табличках содержится данное число, задумавший его тем самым сообщает вам двоичную запись этого числа. Остается лишь перевести ее в десятичную.

Можно поставить вопрос наоборот: если кто-то укажет произвольное число от 1 до 127, то можно сказать, в каких табличках оно есть, а в каких его нет. Для этого достаточно перевести названное число в двоичную систему и назвать номера тех разрядов, в которых получится единица.

Примечание. В каждой из приведенных на рис. 2–8 табличек числа выписаны в порядке возрастания, поэтому структура этих табличек довольно легко обнаруживается. Однако внутри каждой из них числа можно было бы переставить совершенно произвольно, замаскировав, таким образом, способ построения табличек. Можно также не нумеровать карточки, а выкрасить их в разные цвета.

Задание для самостоятельной работы учащихся

Самостоятельно подготовьте таблицы, аналогичные описанным, для отгадывания чисел от 1 до 31 включительно. Сколько таблиц понадобится?

8	9	10	11	12	13	14	15
24	25	26	27	28	29	30	31
40	41	42	43	44	45	46	47
56	57	58	59	60	61	62	63
72	73	74	75	76	77	78	79
88	89	90	91	92	93	94	95
104	105	106	107	108	109	110	111
120	121	122	123	124	125	126	127

4
Рис. 5

16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63
80	81	82	83	84	85	86	87
88	89	90	91	92	93	94	95
112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127

5
Рис. 6

32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63
96	97	98	99	100	101	102	103
104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127

6
Рис. 7

64	65	66	67	68	69	70	71
72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87
88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103
104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127

7
Рис. 8

4. Волшебная карточка

Этот фокус — аналогичный предыдущим, но более эффектный. Для его демонстрации необходимо подготовить шесть³ карточек с числами (см. нижние карточки на рис. 9) и одну “волшебную” карточку (почему “волшебную” — скоро поймете). В принципе карточки с числами не отличаются от карточек, использовавшихся в предыдущем фокусе, однако числа на них не расположены в порядке возрастания, так что ключевые числа (т.е. числа, каждое из которых меньше всех остальных на данной карточке) занимают различные положения среди остальных чисел. Кроме того, некоторые числа на каждой карточке повторяются. На всех карточках имеются отверстия (на рис. 9

³ Для случая, когда отгадываемое число находится в диапазоне от 1 до 64.

они показаны в виде закрашенных кружков). Отверстия в “волшебной” карточке соответствуют местам, где на шести карточках расположены ключевые числа, а на каждой из шести карточек отверстия сделаны в тех же местах, что и на “волшебной” карточке, за исключением одного, где проставлено ключевое число данной карточки.

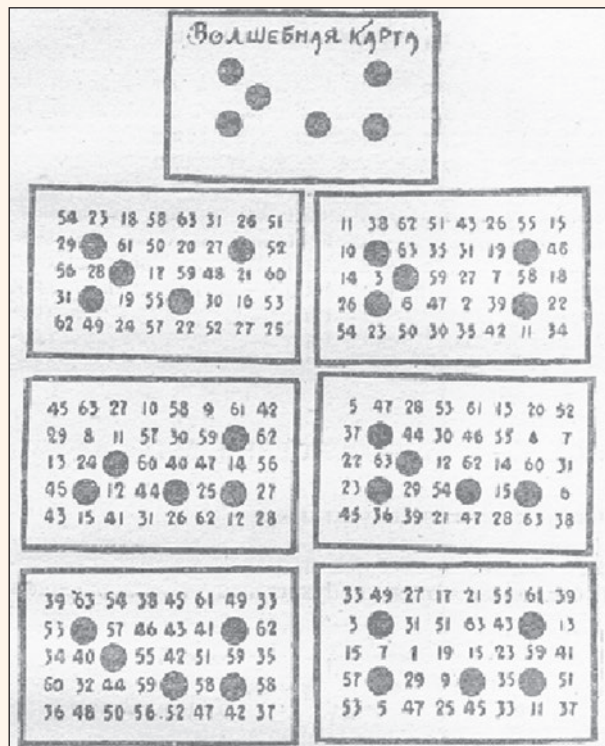


Рис. 9

После того как зритель отберет все карточки, на которых имеется задуманное число, вы кладете их друг на друга и накрываете “волшебной” карточкой. Теперь, чтобы получить задуманное число, нужно сложить числа, видимые сквозь отверстия.

5. Еще фокусы с карточками

Еще ряд эффектных фокусов можно продемонстрировать с использованием карточек, представленных на рис. 10.

Их можно изготовить из картона. Отверстия должны быть чуть больше диаметра карандашей, спиц и т.п. приспособлений, которые вы будете использовать при демонстрации фокусов. Отверстия можно сделать с помощью дырокола. Удобно сначала сделать пять отверстий в одной карточке, а затем использовать ее как шаблон для того, чтобы наметить отверстия на других карточках. Проревав в каждой карточке по пять отверстий, прорежьте промежуток, отделяющий некоторые отверстия от края, так, как показано на рисунке. Отверстия, доходящие до края карт, соответствуют цифре 0, остальные отверстия соответствуют цифре 1⁴. Та-

⁴ Или, наоборот, что не меняет существа вопроса (именно так описан фокус в книге [1]).

ким образом, каждой карте можно сопоставить некоторое двоичное число от 0 до 31 (на рис. 10 карточки нарисованы в случайном порядке). Срезанный правый верхний угол (см. рис. 10) позволяет следить за тем, чтобы карты были ориентированы одинаково (не переворачивались).

Хотя изготовить карты довольно хлопотно, ваш труд будет вознагражден восхищением зрителей, которым вы будете демонстрировать фокусы.

Прежде чем описывать фокусы, заметим следующее. Если собрать все карточки в колоду и, держа ее одной рукой, поставить на стол вертикально, после чего вставить, например, карандаш в одно из отверстий, а потом немного приподнять карандаш, то часть карт (их будет 16) окажется надетой на него. Если затем встряхнуть карандаш, то остальные карты (тоже 16 штук) упадут на стол. Какие именно? Ответ получить несложно — на столе окажутся карты, на которых для соответствующего отверстия прорезан промежуток до края, а на карандаше — карты, у которых такой прорези нет. Можно также сказать, что это будут две группы карт: у которых в выбранном отверстии “выполнен” 0 и 1 соответственно. Для краткости так и будем называть эти группы карт: “группа 0” и “группа 1”.

Первый фокус — отгадывание задуманного зрителем числа из интервала 1–31. Как и ранее, задумавший число должен назвать вам столбцы таблицы на рис. 1⁵, в которых оно встречается. Вы

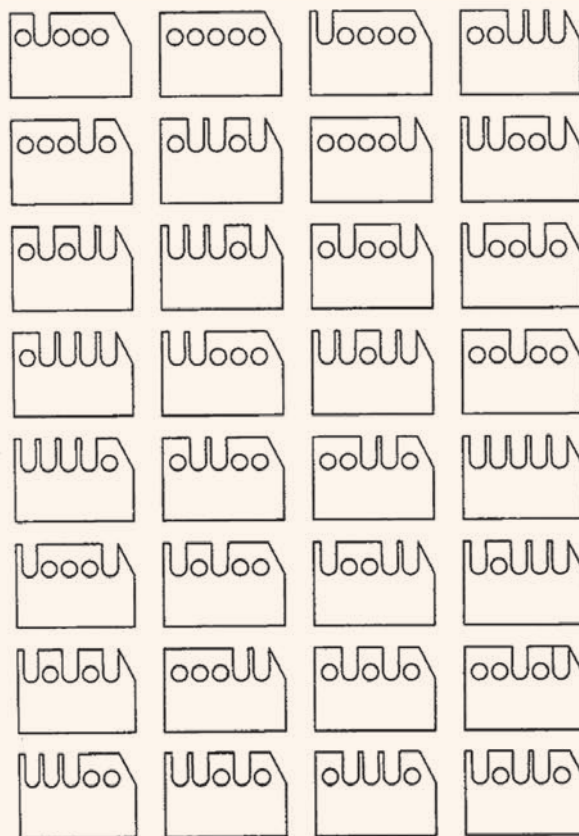


Рис. 10

⁵ Такая таблица должна быть также подготовлена для демонстрации фокуса.

последовательно продеваете карандаш в каждое из отверстий и пять раз проводите “манипуляции” по разделению карточек на две указанные чуть выше группы. Для отверстий, соответствующих указанному зрителем столбцам (считая отверстия, как и столбцы, справа налево), вы отбираете карты группы 1, для остальных — карты группы 0. После обработки каждого “разряда” количество отобранных карт уменьшится вдвое, и последняя отобранная карта будет соответствовать задуманному числу. Выбирать отверстия лучше так, чтобы последние “манипуляции” проводились с отверстием, соответствующим одному из столбцов, названных зрителем, — в этом случае задуманная им карта окажется на карандаше.

Например, задумавший число назвал вам 2-й, 3-й и 4-й столбцы. Для первого справа отверстия вы отбираете 16 карт, оказавшихся на столе (группа 0), для 5-го — также карточки этой группы (их будет 8), для остальных — карточки, оставшиеся на карандаше. Последней оставшейся карточкой будет такая:

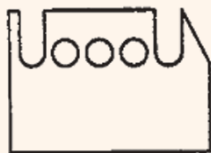


Рис. 11

Она соответствует десятичному числу, двоичное представление которого 01110, то есть 14. Именно это число и задумал зритель.

Если хотите, на каждой карточке можно заранее написать соответствующее десятичное число. Тогда вам не надо будет каждый раз переводить числа из двоичной системы в десятичную. Числа, соответствующие карточкам, изображенным на рис. 10, приведены в таблице:

23	31	15	24
29	18	30	6
20	2	22	13
16	7	4	27
1	19	25	0
14	11	12	8
10	28	21	26
3	5	17	9

Еще два фокуса заключаются в быстрой сортировке карт: нужно расположить их так, чтобы соответствующие карточкам числа последовательно возрастали от 0 до 31.

Перетасуйте карты, как игральные, и сложите их колодой. Проденьте карандаш в крайнее правое отверстие и разделите карточки на две группы. Снимите с карандаша надетые на него карты и положите их в низ остальной колоды. Повторите аналогичные действия для 2-го, 3-го (считая справа налево) и других отверстий. В результате все карты будут расположены по порядку от 0 до 31.

Если же написать на каждой из карт текст из 32 символов (включая “пробелы”), то можно, например, получить новогоднее или другое поздравление. Примеры соответствия чисел (карточек) и символов приведены на рис. 12–13.

П	О	З	Д	Р	А	В	Л	Я	Ю		С		Н	О	В	Ы
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
М		2	0	1	5	-	м		Г	О	Д	О	М	!		
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		

Рис. 12

П	О	З	Д	Р	А	В	Л	Я	Е	М		О	Л	Ю		С
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	Д	Н	Е	М		Р	О	Ж	Д	Е	Н	И	Я	!		
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		

Рис. 13

Задания для самостоятельной работы учащихся

Подумайте над тем, как демонстрировать описанные фокусы, если отверстиям, доходящим до края карт, сопоставить цифру 1, а остальным отверстиям — цифру 0.

6. Фокус с цветными цифрами⁶

С помощью трех таблиц с цветными цифрами (рис. 14–16) можно отгадать задуманное зрителем число, большее 0 и меньше 64, всего за три вопроса.

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
1								

Рис. 14

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
2								

Рис. 15

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
3								

Рис. 16

Первый вопрос: “Какой цвет имеет ваше число в таблице 1?” Второй вопрос: “Какой цвет имеет ваше число в таблице 2?” Разумеется, третий вопрос — о его цвете в табл. 3. Получив ответы, можно, заглянув в такую “шпаргалку” (или выучив ее⁷):

Таблица	Синий	Красный	Черный	Зеленый
1	0	1	2	3
2	0	4	8	12
3	0	16	32	48

определить задуманное число.

Если задумавший число ответил на вопросы, скажем, так: “Синий, зеленый, красный”, — то вы складываете числа 0, 12 и 16, стоящие на пересечении строк и столбцов, соответствующих названным цветам в соответствующих таблицах. Проверьте, что число 28 действительно имеет названные цвета в таблицах (и только оно!).

⁶ Фокус придумал Анатолий Павлович Савин (1932–1998), замечательный человек и популяризатор математики, бесценный член редколлегии физико-математического журнала “Квант”, с которым автор имел честь быть знакомым и сотрудничать.

⁷ О том, как это сделать, — см. далее.

Секрет фокуса в том, что использована четверичная система счисления (с цифрами 0, 1, 2 и 3). Хотя, конечно, это условность. Можно использовать другие символы, например, букву С для нуля, букву К для 1, букву Ч для 2 и букву З для 3.

Число 28 запишется в этой системе как $1 \times 4^2 + 3 \times 4 + 0 \times 1 = 130_4$, или в нашей буквенной записи: КЗС. Если прочесть эту запись справа налево, то получится: “синий зеленый красный”. Это сочетание цветов и назовет задумавший число 28 в качестве цветов в таблицах 1, 2 и 3!

Выпишем цифры десятичных чисел 1–63 при записи в четверичной системе:

Десятичное число	Цифры числа в четверичной системе		
	Последняя	Средняя	Первая
1	1	0	0
2	2	0	0
3	3	0	0
4	0	1	0
5	1	1	0
6	2	1	0
7	3	1	0
...			
11	3	2	0
12	0	3	0
...			
15	3	3	0
16	0	0	1
17	1	0	1
...			
61	1	3	3
62	2	3	3
63	3	3	3

Видно, что последние цифры периодически чередуются через четыре, средние цифры сгруппированы по 4, первая — по 16. Но именно так расположены цветные числа в таблицах 14–16.

Таким образом, у синих чисел в четверичной записи стоит в конце цифра 0, у красных — 1, у черных — 2, у синих — 3. Цвета чисел в табл. 2 точно таким способом определяют вторую цифру в четверичной записи числа, а в табл. 3 — третью. То же самое говорит и “шпаргалка”, с помощью которой можно определить число по его цветам в таблицах. Обратим внимание на то, что числа помещены в таблицах 7×9 для того, чтобы немного запутать зрителей, — если использовать таблицы размером 8×8 , то систему можно обнаружить гораздо быстрее:

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63
1							

Рис. 17

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Рис. 18

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Рис. 19

Задание для самостоятельной работы учащихся

Самостоятельно подготовьте таблицы, аналогичные описанным, для отгадывания чисел от 1 до 26, использующие троичную систему счисления. Сколько таблиц понадобится? Как должна выглядеть “шпаргалка” для расчетов, аналогичная приведенной в описании фокуса?

7. Фокус “15 названий”

Возьмите какие-либо 15 названий — пусть, к примеру, это будут термины, связанные с информатикой и компьютерами (см. табличку на рис. 20).

1. Бит
2. Системный блок
3. Ноутбук
4. Монитор
5. Принтер
6. Джойстик
7. Сканер
8. Байт
9. Логика
10. Система счисления
11. “Мышка”
12. Цифра
13. Число
14. Разряд
15. Программа

Рис. 20

Вы, показывая зрителям эту таблицу, просите одного из них задумать какой-нибудь термин. Потом, стоя спиной к таблицам, показанным на рис. 21–24, вы просите сказать, в каких из этих таблиц присутствует задуманное слово или словосочетание (номер таблицы указан в правом верхнем углу). Полу-

чив ответ, вы сразу называете его. Затем вы обращаетесь к другому зрителю и опять безошибочно называете задуманное слово.

1
Сканер
Число
Логика
Бит
Принтер
Программа
Ноутбук
“Мышка”

Рис. 21

2
Сканер
Система счисления
Программа
“Мышка”
Системный блок
Ноутбук
Разряд
Джойстик

Рис. 22

3
Цифра
Число
Монитор
Принтер
Джойстик
Сканер
Программа
Разряд

Рис. 23

4
Цифра
Байт
Разряд
Логика
Система счисления
Программа
Число
“Мышка”

Рис. 24

Конечно, после разъяснений к нескольким последним фокусам вы уже поняли, в чем секрет данного фокуса и как его демонстрировать. Единственная трудность — запомнить номера 15 слов (словосочетаний). При оформлении таблички на рис. 20 использованы некоторые приемы, облегчающие запоминание. Например, слова *бит* и *байт* даны соответственно под номерами 1 и 8 (1 байт = 8 бит), номер 13 (“несчастливое” число) связано со словом *число*, а перед ним, под номером 12, приведено слово *цифра*. В жизни (не считая уроков информатики) мы пользуемся десятичной системой счисления, поэтому

словосочетание *система счисления* дано под номером... — правильно — 10!

И еще. Чтобы больше “засекретить” фокус, можно в табличке на *рис. 20* номера терминов не приводить (но запомнить их вам, естественно, придется).

8. Фокусы с октаэдром

Посмотрите, пожалуйста, на рисунок:

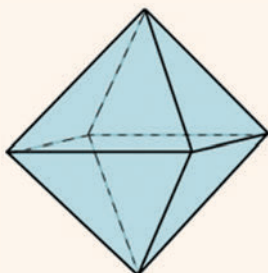


Рис. 25

На нем изображен так называемый “октаэдр”, один из пяти правильных многогранников — тело, имеющее 8^8 граней в виде равностороннего треугольника, 12 ребер, 6 вершин (в каждой вершине сходятся 4 ребра).

Октаэдр можно изготовить из бумаги, используя следующую развертку (см. *рис. 26*):

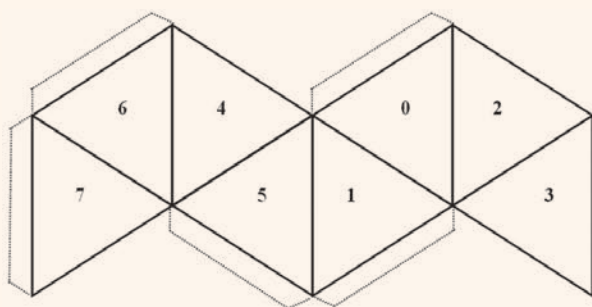


Рис. 26

Удобно сначала получить один равносторонний треугольник (один из возможных вариантов его построения описан в приложении 3). Вырезав получившийся треугольник, можно затем, используя его как шаблон, нарисовать на плотной бумаге шесть треугольников, как изображено выше на развертке. Участки развертки, ограниченные пунктирными линиями (они понадобятся при склейке октаэдра), можно потом пририсовать к треугольникам.

Возникает вопрос — а при чем здесь фокусы? Дело в том, что если на гранях тетраэдра написать числа, указанные на развертке (см. *рис. 27*), то его можно использовать для демонстрации следующего фокуса.

Попросите кого-нибудь задумать любое число от 0 до 7. Положите октаэдр на стол или, держа его в руках, покажите так, чтобы загадавший мог видеть только грани с цифрами 1, 3, 5 и 7, и спросите, ви-

дит ли он задуманное им число. Если он ответит положительно, то запомните про себя число 1.



Рис. 27

Затем вы переворачиваете октаэдр так, чтобы загадавшему были видны только грани с цифрами 2, 3, 6 и 7, и снова задайте тот же вопрос. Если он ответит положительно, то запомните про себя число 2.

В третий (и последний) раз вы повторяете свой вопрос, повернув октаэдр так, чтобы загадавший мог видеть только грани с цифрами 4, 5, 6 и 7. Утвердительный ответ в этом случае оценивается числом 4.

Сложив в уме оценки всех трех ответов, вы получите задуманное число (или 0, если человек, загадавший число, трижды ответит “нет”). Например, если ответы были: “нет”, “да” и “да”, то задуманное число — $6(2 + 4)$, а если “да”, “да” и “нет”, то $3(1 + 2)$.

Секрет фокуса заключается в том, что любое число от 1 до 7 можно представить как сумму чисел 1, 2 и 4:

- $1 = 1$
- $2 = 2$
- $3 = 2 + 1$
- $4 = 4$
- $5 = 4 + 1$
- $6 = 4 + 2$
- $7 = 4 + 2 + 1$

Так вот, четыре числа, которые вы показываете в первый раз (1, 3, 5 и 7), “содержат в себе” единицу, во второй раз (2, 3, 6 и 7) — число 2, в третий раз (4, 5, 6 и 7) — число 4. Отвечая на вопросы, человек, загадавший число, невольно подсказывает вам, нужно ли учитывать 1, 2 или 4 в искомой сумме.

Чтобы было легче отыскать нужные при демонстрации фокуса положения октаэдра, можно как-нибудь пометить три вершины, которые должны быть обращены к вам, когда вы показываете октаэдр зрителям. Например, можно в качестве меток использовать одну, две и три точки:

⁸ По-гречески *восемь* – *октó* (*окто*), откуда и название многогранника.



Рис. 28

Одну точку следует поставить у вершины, которая принадлежит граням с числами 0, 2, 4 и 8, две точки — с числами 0, 1, 4 и 5, три — с числами 0, 1, 2 и 3. При первом показе к вам должна быть обращена вершина с одной точкой, при втором — с двумя, при третьем — с тремя.

Можно также не писать числа на гранях, а окрасить грани октаэдра в разные цвета и отгадывать задуманный цвет. Для этого вы должны каждому из чисел от 0 до 7 сопоставить один из восьми цветов, использованные цвета показать зрителям и предложить кому-то из них выбрать и запомнить любой цвет. Вы потом этот цвет назовете.

Удобно грань с номером 0 сделать белой, а для остальных использовать цвета радуги (кроме фиолетового) и черный:

Число на грани	Цвет
1	Красный
2	Оранжевый
3	Желтый
4	Зеленый
5	Голубой
6	Синий
7	Черный

Конечно, при этом вам придется запомнить эти цвета и числа.



Рис. 29

При демонстрации фокуса теперь нужно трижды показывать зрителю октаэдр и спрашивать, видит ли он “свой” цвет. В первый раз вы показываете ему красную, желтую, голубую и черную грани, во второй — оранжевую, желтую, синюю и черную, в третий — зеленую, голубую, синюю и черную.

По полученной после ответов сумме (как и ранее) вы сможете назвать запомненный цвет. Например, если ответы были: “да”, “нет” и “да”, то это голубой (номер этого цвета равен 5), а если “нет”, “да” и “нет” — то оранжевый (2).

В таком варианте фокуса для отыскания нужного положения октаэдра также можно пометить три соответствующие вершины:



Рис. 30

9. Карточный фокус “Последняя карта”

Зрителю, задумавшему некоторую карту и объявившему ее, вручают колоду карт, в которой карты расположены картинками вниз, и просят, сняв верхнюю карту, переложить ее в низ колоды. Затем снять следующую карту и выложить ее на стол. Очередную верхнюю карту снова положить в низ колоды, следующую за ней — снова выложить на стол, и т.д. до тех пор, пока в руках у зрителя не окажется одна-единственная карта. Она-то и окажется той, которую задумал зритель перед демонстрацией фокуса (естественно, что фокусник подготовил колоду соответствующим образом).

Возникает вопрос — в каком месте исходной колоды должна находиться задуманная карта, чтобы в конце концов она оказалась последней, оставшейся в руках зрителя?

Ясно, что положение задуманной карты в колоде зависит от числа карт в колоде. Но где именно?

Обозначим общее число карт в колоде — n , искомым номер карты — k и проанализируем несколько начальных значений n .

Когда $n = 2$, задуманная карта должна быть первой (сначала она уйдет в низ “колоды”, потом верхняя карта будет убрана на стол и останется только задуманная карта).

Пусть $n = 3$. Так как мы знаем “правильное” положение задуманной карты при $n = 2$, то нам достаточно подумать над тем, где она должна находиться при $n = 3$, чтобы после двух “манипуляций”, в результате которых n станет равно 2, эта карта оказалась первой. Из рис. 31 следует, что ее номер должен быть равен трем.



Рис. 31

Рассуждая аналогично при $n = 4$, получаем $k = 1$ (после двух перемещений получим $n = 3$, а задуманная карта будет третьей — см. рис. 32–33).

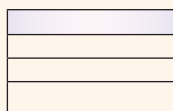


Рис. 32

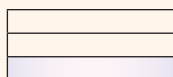


Рис. 33

Далее можно получить, что:

- при $n = 5$ — $k = 3$;
- при $n = 6$ — $k = 5$;
- при $n = 7$ — $k = 7$;
- при $n = 8$ — $k = 1$;
- при $n = 9$ — $k = 3$.

Составим таблицу:

Число карт в колоде, n	Двоичное представление	Номер места, k	Двоичное представление
2	10	1	1
3	11	3	11
4	100	1	1
5	101	3	11
6	110	5	101
7	111	7	111
8	1000	1	1
9	1001	3	11

Анализ показывает, что для определения номера места k следует записать число карт в колоде в двоичной системе счисления и перенести единицу старшего разряда в конец числа. Число, которое получится в результате такой операции, совпадает с номером того места, на котором должна находиться задуманная карта, если считать по порядку, начиная с самой верхней карты в колоде. Например, используется колода из 52 карт. В двоичной системе число 52 запишется как 110100. Перенесем единицу старшего разряда в конец: 101001. В десятичной системе это число равно 41, значит, задуманная карта должна быть 41-й, считая сверху. При демонстрации искомым номер целесообразно определять так:

1) из общего числа карт n вычесть максимальную степень двойки, не превышающую n ;

2) полученную разность удвоить;

3) к результату прибавить 1.

Примеры:

1) при $n = 52$:

$$- 52 - 32 = 20;$$

$$- 20 \times 2 = 40;$$

$$- 40 + 1 = 41;$$

2) при $n = 36$:

$$- 36 - 32 = 4;$$

$$- 4 \times 2 = 8;$$

$$- 8 + 1 = 9.$$

Вариант фокуса — подготовив колоду соответствующим образом, заранее объявить, какая последняя карта останется в руках у зрителя.

Чтобы еще более “удивить” зрителей, можно при каждой демонстрации фокуса менять число карт в колоде.

Задания для самостоятельной работы учащихся

1. Определите, сколько карт должно быть в колоде, если нужно, чтобы последней осталась:
 - самая верхняя карта;
 - самая нижняя карта.
2. Подумайте, может ли последней остаться карта, которая в исходной колоде была второй сверху.

10. Карточный фокус “Ваша карта”

В заключение — один очень эффектный карточный фокус. Он известен под названием задачи Жергонна о трех стопках карт (в честь французского математика Жозефа Жергонна, который первым занялся анализом этой задачи еще в начале XIX века). В нем используется колода из 27 карт. Зритель запоминает одну из них и, держа колоду открытыми картами вниз, вынимает из нее по одной карте и раскладывает их слева направо в три стопки картинками вверх (первая карта идет в первую стопку, вторая — во вторую, третья — в третью, четвертая — в первую и т.д.). Каждая стопка будет состоять из девяти карт. Указав фокуснику стопку, в которой лежит запомненная карта, зритель складывает стопки в любом порядке (возможны шесть вариантов размещения стопок), не перетасовывая карты внутри стопок. Затем он опять переворачивает колоду картинками вниз и начинает еще раз раскладывать карты в три стопки картинками вверх. Показав, где теперь лежит задуманная карта, зритель повторяет ту же самую процедуру в третий раз, после чего колода, составленная из трех стопок, кладется на стол.

Все это время фокусник ни разу не прикасается к картам, но тем не менее он мгновенно говорит, в каком месте лежит задуманная карта.

Секрет этого фокуса, связанного с троичной системой счисления, заключается в том, чтобы заметить, куда каждый раз зритель клал стопку

с задуманной картой — под колоду, в середину ее или наверх. Обозначим эти три положения цифрами 0 (когда стопка с запомненной картой находится в верхней части колоды), 1 (когда стопка лежит в середине) и 2 (когда стопка положена в самый низ). Запомнив три положения нужной стопки, фокусник составляет из этих цифр трехзначное число, имея в виду, что оно представлено в троичной системе счисления. Затем он “читает” это число справа налево (можно также сразу первую из замеченных цифр ставить в младший, крайний справа разряд, следующую — во второй и последнюю — в старший, первый, разряд). Если перевести полученное трехзначное число, рассматривая его как троичное, в десятичную систему и прибавить единицу, то результат укажет номер искомой карты — ее можно найти, начав счет с верхней карты колоды.

Например, если первый раз стопка с запомненной картой была положена вверх колоды (этому положению соответствует цифра 0), во второй раз — в ее середину (1) и, наконец, последний раз — в самый низ (2). В результате получается троичное число 210, равное десятичному числу 21. Значит, нужная карта — 22-я сверху.

Чтобы перевести троичное число в десятичную систему “в уме”, следует запомнить “вклад” каждой троичной цифры в результат:

Номер цифры	1			2			3		
Цифра	0	1	2	0	1	2	0	1	2
“Вклад”	0	9	18	0	3	6	0	1	2

что сделать несложно.

Конечно, вы можете все манипуляции проводить сами (а зритель только трижды называет стопку со “своей” картой). В этом случае можно, с целью облегчения расчетов, всегда класть стопку с запомненной зрителем картой в середину. При этом эффектным является нахождение карты в колоде руками, находящимися за спиной.

Задания для самостоятельной работы учащихся

1. Объясните, почему указанная фокусником карта будет именно той, которую запомнил зритель.

2. Подумайте, как нужно самому демонстрировать фокус, чтобы некоторая запомненная зрителем карта оказалась в конце концов в месте, которое заранее назовет фокусник или зритель.

В заключение заметим следующее. К сожалению, в книгах [1–2]⁹, в которых описан данный фокус, имеются неточности, из-за которых продемонстрировать фокус не удастся. Так в [1] ошибочно указано, что после третьего сбора трех стопок в одну ее нужно также перевернуть картинками на картах вниз. Кроме того, в книге в приведенном примере неправильно определен номер искомой карты. Эти ошибки сделаны при подготовке русского издания книги. В [2] вообще не говорится о том, что собранные стопки перед раскладыванием нужно перевернуть.

Только в книге [3], выпущенной ограниченным тиражом и являющейся перепечаткой книг того же автора, изданных в 1908–1911 гг., этот фокус описан правильно, хотя о двукратном переворачивании карт тоже прямо не говорится (в начале описания фокуса указывается: “карты в руках раскладывающего повернуты крапом вверх, и раскладывающий поворачивает их лицом вверх при распределении на три кучки”, потом говорится, что второй раз они раскладываются “на три кучки, как указано выше”; о переворачивании всей колоды после третьего сбора карт ничего не говорится, что правильно).

Литература

1. Гарднер М. Математические досуги. М.: Мир, 1972.
2. Гашков С.Б. Занимательная компьютерная арифметика. Математика и искусство счета на компьютерах и без них. М.: URSS, 2012.
3. Игнатьев Е.И. В царстве смекалки. Ростов-на-Дону: Ростовское книжное изд-во, 1995.
4. Савин А. Угадай число. / Квант, 1987, № 9.

Ответы на задания для самостоятельной работы учащихся

3. Семь табличек

Понадобятся пять таблиц:

1	3	5	7
9	11	13	15
17	19	21	23
25	27	29	31

2	3	6	7
10	11	14	15
18	19	22	23
26	27	30	31

16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31

4	5	6	7
12	13	14	15
20	21	22	23
28	29	30	31

8	9	10	11
12	13	14	15
24	25	26	27
28	29	30	31

9. Карточный фокус “Последняя карта”

1. Поскольку двоичная запись самой верхней карты имеет вид 1, то число карт в колоде может выражаться двоичными числами 10, 100, 1000, 10 000, ... (или, в десятичной системе, 2, 4, 8, 16, ...).

⁹ А также в ряде источников в Интернете.

Если требуется, чтобы в руках у зрителя осталась самая нижняя карта, то в колоде должно быть 11, 111, 1111, 11111, ... карт (число карт записано в двоичной системе), или, в десятичной системе — 3, 7, 15, 31, ... карт.

2. Нет. Более того, последняя карта никогда не может находиться в колоде на четном месте (считая сверху), поскольку порядковый номер такой карты, выраженный в двоичной системе, всегда заканчивается единицей (так как старшая цифра числа, равная 1, перемещается в конец числа), а двоичные числа, оканчивающиеся цифрой 1, нечетные.

10. Карточный фокус “Ваша карта”

1. Обсудим схему перемещения карт при демонстрации фокуса.

Пусть a, b, c означают цифры, соответствующие положениям стопки с запомненной картой в колоде в первый, второй и третий раз: 0 — верхняя часть колоды, 1 — в середине и 2 — стопка положена в самый низ.

Итак, после первого расклада и сбора трех стопок в общую получена колода (на рис. 34 показан вариант при $a = 1$, стопка с запомненной картой оттенена):

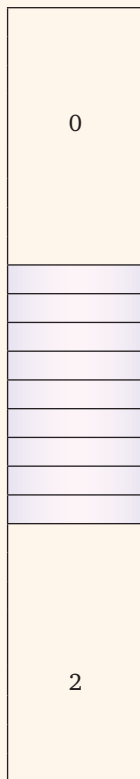


Рис. 34

Далее формируется второй расклад:

1) вся колода переворачивается;

2) сначала распределяются по трем группам $(2 - a)$ стопок¹⁰ из 9 карт, которые в сложной колоде находились под стопкой с запомненной картой, что даст во втором раскладе $9 \times (2 - a)/3 = 6 - 3a$ карт в каждой стопке;

¹⁰ В рассматриваемом случае — одна стопка, обозначенная на рис. 34 цифрой 2. При $a = 2$ таких стопок не будет.

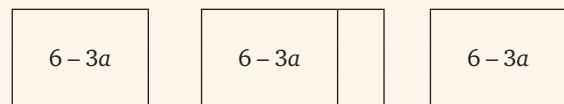


Рис. 35

2) затем раскладывается та стопка, в которой была запомненная карта, что добавляет еще по три карты в каждую группу второго расклада:

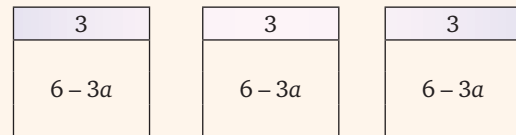


Рис. 36

3) раскладываются a стопок:

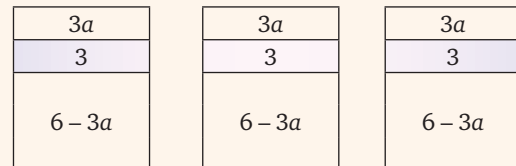


Рис. 37

Так что если теперь указать стопку, в которой находится запомненная карта, то она будет там в числе последних (верхних) $3a + 3$ карт.

Собираем стопки:

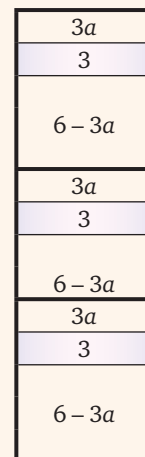


Рис. 38

После этого при третьем раскладе после переворачивания всей колоды на три группы распределяются:

1) $(2 - b)$ стопок;

2) из стопки с “нужной” картой (см. рис. 37):

— $(6 - 3a)$ карт;

— 3 карты из тройки с запомненной картой;

— $3a$ карты;

3) b стопок.

В результате каждая стопка будет иметь вид:

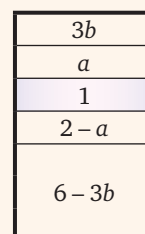


Рис. 39

Собираем стопки последний раз:

3b
a
1
2 - a
6 - 3b
3b
a
1
2 - a
6 - 3b
3b
a
1
2 - a
6 - 3b

Рис. 40

Перед стопкой с запомненной картой (она — одна из затененных на рис. 34) будет размещено c стопок. Значит, эта карта будет при счете сверху вниз иметь номер:

$$9c + 3b + a + 1.$$

Если вспомнить понятие “развернутая форма записи числа”, то можно понять, что выражение $9c + 3b + a = c3^2 + 3b + a$ есть десятичное число, соответствующее троичному числу вида cba , а именно такое число и получается при демонстрации фокуса.

2. Нужно определить места размещения стопки с запомненной картой при известном номере N карты в последней собранной колоде. Это можно сделать следующим образом:

- 1) вычтешь из N единицу;
- 2) полученное число представить в виде суммы максимально возможных чисел из последней строки табл. 5;

3) по полученным значениям определить три цифры троичного числа, соответствующие вкладу каждого из слагаемых (также используя информацию из табл. 5); при необходимости указывается также цифра 0, в том числе начальная;

4) цифры этого числа, прочитанные справа налево, указывают места размещения стопок: 0 — стопка с запомненной картой должна быть в верхней части колоды, 1 — в середине и 2 — внизу.

Например, если нужно, чтобы карта зрителя оказалась 12-й:

1) $12 - 1 = 11$;

2) $11 = 9 + 2$;

3) соответствующее число — 102;

4) число-результат — 201 (первый раз — вниз колоды, второй раз — наверх, третий раз — в ее середину).

При $N = 19$:

1) $19 - 1 = 18$;

2) $18 = 18 + 0 + 0$;

3) соответствующее число — 200;

4) число-результат — 002 (первые два раза стопка с запомненной картой должна быть положена в верх колоды, в третий раз — в ее низ).

При $N = 6$:

1) $6 - 1 = 5$;

2) $5 = 3 + 2$;

3) соответствующее число — 012;

4) число-результат — 210.

Программы, моделирующие некоторые из описанных фокусов

П1.1. Фокус “Волшебная таблица”

В программе, моделирующей этот фокус, таблица с числами, которая названа “волшебной” (см. рис. 1), представим в виде двумерного массива из 16 строк и пяти столбцов с именем *табл*.

Заполнение этого массива в программе на школьном алгоритмическом языке проведем следующим образом:

1) 1-й столбец слева:

```
нц для i от 1 до 16
    табл[i, 1] := 15 + i
```

кц

2) 2-й столбец:

```
нц для i от 1 до 8
    табл[i, 2] := 7 + i
```

кц

нц для i от 9 до 16

```
    табл[i, 2] := 15 + i
```

кц

3) 3-й столбец:

```
нц для i от 1 до 4
    табл[i, 3] := i + 3
```

кц

нц для i от 5 до 8

```
    табл[i, 3] := i + 7
```

кц

нц для i от 9 до 12

```
    табл[i, 3] := i + 11
```

кц

нц для i от 13 до 16

```
    табл[i, 3] := i + 15
```

кц

4) 4-й столбец:

```
нц для i от 1 до 16
```

```
    если mod(i, 2) = 1
```

то

```
        табл[i, 4] := i * 2
```

иначе

```

табл[i, 4] := i * 2 - 1
все
кц
5) 5-й столбец:
нц для i от 1 до 16
табл[i, 5] := i * 2 - 1
кц

```

Убедитесь в правильности использованных зависимостей между номерами строк и значениями элементов в столбцах.

При выводе таблицы нумерацию столбцов проведем слева направо, что отличается от нумерации, приведенной на рис. 1.

Теперь о том, как определить задуманное число. Можно поступить так:

1) узнать количество столбцов таблицы, в которых имеется задуманное число. Пусть имя этой величины — *сколько*;

2) сколько раз © узнать номер столбца и для каждого номера (*ном*) учесть его “вклад” в общее искоемое значение следующим образом:

```

выбор
при ном = 1: ответ := ответ + 16
при ном = 2: ответ := ответ + 8
при ном = 3: ответ := ответ + 4
при ном = 4: ответ := ответ + 2
при ном = 5: ответ := ответ + 1
все

```

— где *ответ* — искоемое число.

Такая методика позволяет вводить номера столбцов в любом порядке.

Вся программа имеет вид:

```

алг Фокус_Волшебная_таблица
нач цел таб табл[1:16, 1:5],
цел сколько, номер, i, j, ответ
вывод нс, "Задумайте целое число
от 1 до 31"
|Приостановка программы до нажатия
любой клавиши
вывод нс, "Для продолжения -
нажмите любую клавишу"
нц
кц_при клав <> 0
|Заполнение таблицы
|1-й столбец
... (см. выше)
|Выводим таблицу
нц для i от 1 до 16
вывод нс
нц для j от 1 до 5
если табл[i, j] > 9
то
вывод табл[i, j], " "
иначе
вывод " ", табл[i, j], " "
все
кц
кц
|и номера столбцов

```

```

вывод нс, "-----"
вывод нс, "1 2 3 4 5"
|Определение задуманного числа
ответ := 0
вывод нс, "В каком количестве столбцов
таблицы имеется Ваше число? "
ввод сколько
нц для i от 1 до сколько
вывод нс, "Введите номер ", i, "-го
столбца "
ввод номер
выбор
при номер = 1: ответ := ответ + 16
... (см. выше)
все
кц
|Приостановка программы до нажатия
любой клавиши
вывод нс, "Для продолжения - нажмите
любую клавишу"
нц
кц_при клав <> 0
|Вывод ответа
вывод нс, "Вы задумали число:", ответ
кон

```

Язык Паскаль

```

Uses CRT;
Var табл: array [1..16, 1..5] of byte;
skolko, nom, i, j, otvet: byte;
BEGIN
clrscr;
writeln('Задумайте целое число
от 1 до 31');
{Приостановка программы до нажатия
любой клавиши}
writeln('Для продолжения - нажмите
любую клавишу');
repeat
until keypressed;
{Заполнение таблицы
1-й столбец}
for i := 1 to 16 do
табл[i, 1] := 15 + i;
{2-й столбец}
for i := 1 to 8 do табл[i, 2] := 7 + i;
for i := 9 to 16 do табл[i, 2] := 15 + i;
{3-й столбец}
for i := 1 to 4 do табл[i, 3] := i + 3;
for i := 5 to 8 do табл[i, 3] := i + 7;
for i := 9 to 12 do табл[i, 3] := i + 11;
for i := 13 to 16 do табл[i, 3] := i + 15;
{4-й столбец}
for i := 1 to 16 do
if i mod 2 = 1 then табл[i, 4] := i * 2
else табл[i, 4] := i * 2 - 1;
{5-й столбец}
for i := 1 to 16 do табл[i, 5] := i * 2 - 1;
{Выводим таблицу}

```

```

for i := 1 to 16 do
  begin
    writeln;
    for j := 1 to 5 do write(tabl[i, j]:4)
    end;
    {и номера столбцов}
    writeln;
    writeln('-----');
    writeln('  1  2  3  4  5');
    otvet := 0;
    write('В каком количестве столбцов
          таблицы имеется Ваше число? ');
    readln(skolko);
    for i := 1 to skolko do
      begin
        write('Введите номер ",
              i, "-го столбца ');
        readln(nom);
        case nom of
          1: otvet := otvet + 16;
          2: otvet := otvet + 8;
          3: otvet := otvet + 4;
          4: otvet := otvet + 2;
          5: otvet := otvet + 1
        end
      end;
    end;
    write('Вы задумали число: ', otvet)
  END.

```

Задание для самостоятельной работы учащихся

Разработайте программу, моделирующую аналогичный фокус с отгадыванием задуманного числа из интервала 1–63.

П1.2. Фокус “Семь табличек”

Отличие программы для демонстрации данного фокуса от предыдущей состоит в том, что нужно выводить семь табличек с числами.

Числа каждой из этих табличек можно записывать в один массив из восьми строк и восьми столбцов с именем *табл* и после заполнения данными из таблички сразу же выводить его на экран (семь раз).

Заполнение массива *табл* проводится следующим образом:

1) 1-й таблицы (см. рис. 2):

```

|1-я строка
нц для j от 1 до 8
табл[1, j] := 2 * j - 1
кц

```

```

|Остальные строки
нц для i от 2 до 8
  нц для j от 1 до 8
    табл[i, j] := табл[i - 1, j] + 16
  кц
кц

```

2) 2-й таблицы (см. рис. 3):

```

|1-я строка
нц для j от 1 до 8

```

```

если mod(j, 2) = 1
  то
    табл[1, j] := j * 2
  иначе
    табл[1, j] := j * 2 - 1
все
кц
|Остальные строки
нц для i от 2 до 8
  нц для j от 1 до 8
    табл[i, j] := табл[i - 1, j] + 16
  кц
кц

```

3) 3-й таблицы (см. рис. 4):

```

|1-я строка
нц для j от 1 до 4
табл[1, j] := j + 3
кц
нц для j от 5 до 8
табл[1, j] := табл[1, j - 4] + 8
кц

```

```

|Остальные строки
нц для i от 2 до 8
  нц для j от 1 до 8
    табл[i, j] := табл[i - 1, j] + 16
  кц
кц

```

4) 4-й таблицы (см. рис. 5):

```

|1-я строка
нц для j от 1 до 8
табл[1, j] := j + 7
кц
|Остальные строки
нц для i от 2 до 8
  нц для j от 1 до 8
    табл[i, j] := табл[i - 1, j] + 16
  кц
кц

```

5) 5-й таблицы (см. рис. 6):

```

|1-я строка
нц для j от 1 до 8
табл[1, j] := j + 15
кц

```

```

|2-я строка
нц для j от 1 до 8
табл[2, j] := табл[1, j] + 8
кц

```

```

|3-я строка
нц для j от 1 до 8
табл[3, j] := табл[2, j] + 24
кц

```

```

|4-я строка
нц для j от 1 до 8
табл[4, j] := табл[3, j] + 8
кц

```

```

|Остальные строки
нц для i от 5 до 8
  нц для j от 1 до 8
    табл[i, j] := табл[i - 4, j] + 64
  кц
кц

```


6) 6-й таблицы (см. рис. 7):

```
|1-я строка
нц для j от 1 до 8
  табл[1, j] := j + 31
кц
|2-я, 3-я и 4-я строки
нц для i от 2 до 4
  нц для j от 1 до 8
    табл[i, j] := табл[i - 1, j] + 8
  кц
кц
|Остальные строки
нц для i от 5 до 8
  нц для j от 1 до 8
    табл[i, j] := табл[i - 4, j] + 64
  кц
кц
```

7) 7-й таблицы (см. рис. 8):

```
|1-я строка
нц для j от 1 до 8
  табл[1, j] := j + 63
кц
|Остальные строки
нц для i от 2 до 8
  нц для j от 1 до 8
    табл[i, j] := табл[i - 1, j] + 8
  кц
кц
```

Изменится также “вклад” каждой таблицы в общее искомое значение:

```
выбор
  при ном = 1: ответ := ответ + 1
  при ном = 2: ответ := ответ + 2
  при ном = 3: ответ := ответ + 4
  при ном = 4: ответ := ответ + 8
  при ном = 5: ответ := ответ + 16
  при ном = 6: ответ := ответ + 32
  при ном = 7: ответ := ответ + 64
```

все

— где *ном* — номер таблицы, в которой имелось задуманное число, *ответ* — искомое число.

Всю программу предлагаем читателям собрать самостоятельно. При этом, так как вывод заполненных таблиц проводится многократно, с целью уменьшения размера программы целесообразно использовать вспомогательную процедуру вывода на экран массива *табл*.

Примечание. В программе должно быть сообщение о том, что следует запомнить номера таблиц, в которых имеется запомненное число.

Задание для самостоятельной работы учащихся

Разработайте программу, моделирующую аналогичный фокус с отгадыванием задуманного числа из интервала 1–63.

П1.3. Фокус “15 названий”

Программа для демонстрации данного фокуса также во многом аналогична рассмотренным. Отличия следующие.

1) понадобится массив с перечнем исходных терминов (см. рис. 20). Имя этого массива — *список_слов*, размер — 15 элементов. Вспомним “секрет” фокуса и заметим, что порядковые номера терминов определяют содержание других демонстрируемых табличек;

2) информация, выводимая в виде таблиц на рис. 21–24, в программе хранится в одномерном массиве из восьми элементов с данными строкового типа. Пусть имя этого массива — *табл*. Для табл. 1 (рис. 21) в массиве *табл* должны быть представлены термины, номера которых в двоичной записи имеют 1 в последнем разряде, для табл. 2 (рис. 22) — 1 во втором справа разряде и т.д.;

3) для получения ответа следует, используя, как и ранее, “вклад” каждой из таблиц, подсчитать номер искомого термина и по нему определить сам термин, используя массив *список_слов*.

Вся программа имеет вид:

```
лит табл табл [1:8]
цел i
алг Фокус_15_названий_
нач лит таб список_слов[1:15],
  цел сколько, номер_таблицы,
  номер_слова,
  лит ответ
|Заполняем массив со всеми названиями
список_слов[1] := "Бит"
список_слов[2] := "Системный блок"
список_слов[3] := "Ноутбук"
список_слов[4] := "Монитор"
список_слов[5] := "Принтер"
список_слов[6] := "Джойстик"
список_слов[7] := "Сканер"
список_слов[8] := "Байт"
список_слов[9] := "Логика"
список_слов[10] := "Система счисления"
список_слов[11] := "'Мышка'"
список_слов[12] := "Цифра"
список_слов[13] := "Число"
список_слов[14] := "Разряд"
список_слов[15] := "Программа"
|и выводим его
вывод нс, "Запомните какой-либо термин:"
нц для i от 1 до 15
  вывод нс, список_слов[i]
кц
вывод нс, "Если в таблицах, которые
будут вам предложены, имеется этот
термин, то запомните номер
соответствующей таблицы"
|Приостановка программы до нажатия
любой клавиши
вывод нс, "Для продолжения
нажмите любую клавишу"
нц
кц_при клав <> 0
|Заполнение 1-й таблицы
```

```

табл[1] := "Бит"
табл[2] := "Ноутбук"
табл[3] := "Принтер"
табл[4] := "Сканер"
табл[5] := "Логика"
табл[6] := "'Мышка'"
табл[7] := "Число"
табл[8] := "Программа"
вывод нс, "Таблица № 1"
Вывод_таблицы
|2-я таблица
табл[1] := "Сканер"
табл[2] := "Система счисления"
табл[3] := "Программа"
табл[4] := "'Мышка'"
табл[5] := "Системный блок"
табл[6] := "Ноутбук"
табл[7] := "Джойстик"
табл[8] := "Разряд"
вывод нс, "Таблица № 2"
Вывод_таблицы
|3-я таблица
табл[1] := "Цифра"
табл[2] := "Число"
табл[3] := "Монитор"
табл[4] := "Принтер"
табл[5] := "Джойстик"
табл[6] := "Сканер"
табл[7] := "Программа"
табл[8] := "Разряд"
вывод нс, "Таблица № 3"
Вывод_таблицы
|4-я таблица
табл[1] := "Байт"
табл[2] := "Логика"
табл[3] := "Система счисления"
табл[4] := "'Мышка'"
табл[5] := "Цифра"
табл[6] := "Число"
табл[7] := "Разряд"
табл[8] := "Программа"
вывод нс, "Таблица № 4"
Вывод_таблицы
|Определение номера задуманного слова
вывод нс, "В каком количестве таблиц
имеется Ваше слово? "
ввод сколько
номер_слова := 0
нц для i от 1 до сколько
  вывод нс, "Введите номер ", i, "-й
таблицы "
  ввод номер_таблицы
  |Учитываем "вклад" данной таблицы
  выбор
    при номер_таблицы = 1:
      номер_слова := номер_слова + 1
    при номер_таблицы = 2:
      номер_слова := номер_слова + 2
    при номер_таблицы = 3:

```

```

      номер_слова := номер_слова + 4
    при номер_таблицы = 4:
      номер_слова := номер_слова + 8
  все
кц
  |Вывод ответа
вывод нс, "Вы выбрали слово: ",
      список_слов[номер_слова]
кон

```

— где *Вывод_таблицы* — вспомогательная процедура вывода на экран массива *табл*.

Язык Паскаль

```

{Фокус "15 названий"}
Uses CRT;
Var spisok_slov: array [1..15]
  of string[30];
  tabl: array [1..8] of string[30];
  skolko, nomer_tablitsi, nomer_slova,
  i : byte;
Procedure Vivod_tablitsi;
begin
  for i := 1 to 8 do writeln(tabl[i])
end;
BEGIN
clrscr;
{Заполняем массив со всеми названиями}
spisok_slov[1] := 'Бит';
spisok_slov[2] := 'Системный блок';
spisok_slov[3] := 'Ноутбук';
spisok_slov[4] := 'Монитор';
spisok_slov[5] := 'Принтер';
spisok_slov[6] := 'Джойстик';
spisok_slov[7] := 'Сканер';
spisok_slov[8] := 'Байт';
spisok_slov[9] := 'Логика';
spisok_slov[10] := 'Система счисления';
spisok_slov[11] := "'Мышка'";
spisok_slov[12] := 'Цифра';
spisok_slov[13] := 'Число';
spisok_slov[14] := 'Разряд';
spisok_slov[15] := 'Программа';
{и выводим его}
writeln('Запомните какой-либо термин:');
for i := 1 to 15 do
  writeln(spisok_slov[i]);
write ('Если в таблицах, которые
будут вам предложены, ' );
writeln('имеется этот термин,
то запомните номер
соответствующей таблицы');
{Приостановка программы до нажатия
любой клавиши}
writeln('Для продолжения - нажмите
любую клавишу');
repeat
until keypressed;
clrscr;
{Заполнение 1-й таблицы}

```

```

tabl[1] := 'Вит';
tabl[2] := 'Ноутбук';
tabl[3] := 'Принтер';
tabl[4] := 'Сканер';
tabl[5] := 'Логика';
tabl[6] := '"Мышка"';
tabl[7] := 'Число';
tabl[8] := 'Программа';
writeln('Таблица № 1');
Vivod_tablitsi;
{Приостановка программы до нажатия
любой клавиши}
writeln('Для продолжения нажмите
любую клавишу');
repeat
until keypressed;
{2-я таблица}
tabl[1] := 'Сканер';
tabl[2] := 'Система счисления';
tabl[3] := 'Программа';
tabl[4] := '"Мышка"';
tabl[5] := 'Системный блок';
tabl[6] := 'Ноутбук';
tabl[7] := 'Джойстик';
tabl[8] := 'Разряд';
clrscr;
writeln('Таблица № 2');
Vivod_tablitsi;
{Приостановка программы до нажатия
любой клавиши}
writeln('Для продолжения - нажмите
любую клавишу');
repeat
until keypressed;
{3-я таблица}
tabl[1] := 'Цифра';
tabl[2] := 'Число';
tabl[3] := 'Монитор';
tabl[4] := 'Принтер';
tabl[5] := 'Джойстик';
tabl[6] := 'Сканер';
tabl[7] := 'Программа';
tabl[8] := 'Разряд';
clrscr;
writeln('Таблица № 3');
Vivod_tablitsi;
{Приостановка программы до нажатия
любой клавиши}
writeln('Для продолжения нажмите
любую клавишу');
repeat
until keypressed;
{4-я таблица}
tabl[1] := 'Байт';
tabl[2] := 'Логика';
tabl[3] := 'Система счисления';
tabl[4] := '"Мышка"';
tabl[5] := 'Цифра';
tabl[6] := 'Число';

```

```

tabl[7] := 'Разряд';
tabl[8] := 'Программа';
clrscr;
writeln('Таблица № 4');
Vivod_tablitsi;
{Определение номера задуманного слова}
writeln('В каком количестве таблиц
имеется Ваше слово? ');
readln(skolko);
nomer_slova := 0;
for i := 1 to skolko do
begin
writeln('Введите номер ', i, '-й
таблицы ');
readln(nomer_tablitsi);
{Учитываем "вклад" данной таблицы}
case nomer_tablitsi of
1: nomer_slova := nomer_slova + 1;
2: nomer_slova := nomer_slova + 2;
3: nomer_slova := nomer_slova + 4;
4: nomer_slova := nomer_slova + 8
end
end;
{Вывод ответа}
writeln('Вы выбрали слово: ',
spisok_slov[nomer_slova])
END.

```

Задание для самостоятельной работы учащихся

Разработайте программу, моделирующую аналогичный фокус с использованием перечня из 31 термина.

П1.4. Карточный фокус "Ваша карта"

Разработаем программу, моделирующую такой вариант фокуса. Его демонстрирует фокусник, сидящий у компьютера, а зрители запоминают карту и трижды указывают номера стопок, в которых находится эта карта при раскладывании карт.

Основные этапы демонстрации фокуса:

1. Формирование исходного набора карт.
2. Раскладывание карт на три стопки.
3. Запрос к зрителю о стопке с запомненной картой.
4. Сбор трех стопок в общую колоду.
5. Переворачивание колоды (карты — картинками вниз).
6. Раскладывание карт на три стопки.
7. Запрос к зрителю о стопке с запомненной картой.
8. Сбор трех стопок в общую колоду.
9. Переворачивание колоды (карты — картинками вниз).
10. Раскладывание карт на три стопки.
11. Запрос к зрителю о стопке с запомненной картой.
12. "Отгадывание" запомненной карты.

Весь набор карт будем моделировать в виде массива из 27 элементов с именем *колода*. Про-

ще всего его заполнить конкретными неповторяющимися двухсимвольными значениями, например:

```
колода[1] := "ДП"
колода[2] := "ТП"
...
колода[8] := "ВВ"
колода[9] := "6Б"
...
колода[20] := "КТ"
колода[21] := "8Т"
...
колода[26] := "КЧ"
колода[27] := "7Ч"
```

— в которых первый символ соответствует достоинству карты (“шестерка” — 6, “валет” — “В” и т.д.), второй — масти (пики — “П” и т.д.). При этом при раскладывании карт будем считать, что карта колода[1] находится наверху колоды, а колода[27] — внизу (все карты расположены картинкой вниз).

Стопки смоделируем в виде трех массивов *стопка1*, *стопка2* и *стопка3* из 9 элементов каждый. Ясно, что тип элементов будет таким же, как и в массиве *колода*. Но здесь условимся, что карты *стопка[1]*, *стопка[2]* и *стопка[3]* находятся внизу соответствующей стопки. Учитывая это, обсудим, как из общей колоды (массива *колода*) получить три стопки (три массива).

Если проанализировать места размещения карт в первой стопке:

```
стопка1[1] = колода[1]
стопка1[2] = колода[4]
...
стопка1[9] = колода[25]
```

то можно первую стопку сформировать так:

```
нц для i от 1 до 9
  стопка1[i] := колода[i * 3 - 2]
кц
```

Аналогично для второй и третьей стопок

```
нц для i от 1 до 9
  стопка2[i] := колода[i * 3 - 1]
кц
```

и

```
нц для i от 1 до 9
  стопка3[i] := колода[i * 3]
кц
```

Можно три оператора цикла объединить в один:

```
нц для i от 1 до 9
  стопка1[i] := колода[i * 3 - 2]
  стопка2[i] := колода[i * 3 - 1]
  стопка3[i] := колода[i * 3]
кц
```

После формирования стопок (заполнения массивов) их нужно вывести на экран:

```
нц для i от 9 до 1 шаг -1
  вывод нс, стопка1[i], " ", стопка2[i],
    " ", стопка3[i]
кц
```

Обратим внимание на то, что элементы массивов выводятся, начиная с последних значений.

Под стопками укажем также их номера:

```
вывод нс, "1 2 3"
```

После ввода номера стопки с запомненной зрителем карты три стопки собираются в общую колоду. С целью упрощения программы примем, что первая стопка карт всегда будет первой сверху в общем наборе, вторая — второй, третья — третьей (это не повлияет на эффективность фокуса). При таком допущении сбор карт трех стопок в общую может быть проведен так:

```
|Первая стопка в общей колоде
нц для i от 1 до 9
  колода[i] := стопка1[10 - i]
кц
|Вторая стопка
нц для i от 10 до 18
  колода[i] := стопка2[19 - i]
кц
|Третья стопка
нц для i от 19 до 27
  колода[i] := стопка3[28 - i]
кц
```

Предлагаем читателям самостоятельно убедиться в правильности приведенных зависимостей между значениями элементов массивов.

Итак, после первого раскладывания карт они собраны в общей стопке. Для нахождения запомненной карты нам надо узнать положение стопки с этой картой в общей стопке. Это же придется сделать еще дважды. Эти три значения (количество стопок, размещенных в общей колоде выше стопки с запомненной картой) удобно хранить в массиве, который назовем *положение*.

Если номер стопки, который указал зритель, обозначить *номер_стопки*, то с учетом допущений о размещении отдельных стопок каждый элемент массива *положение* определяется как *номер_стопки - 1*.

Далее перед вторым раскладыванием собранные карты надо перевернуть. Мы пришли к задаче перестановки всех элементов массива *колода* в обратном порядке. Эта задача решается следующим образом:

```
|Первую половину массива колода
|обмениваем местами со второй половиной
нц для i от 1 до 13
  всп := колода[i]
  колода[i] := колода[27 - i + 1]
  колода[27 - i + 1] := всп
кц
```

— где *всп* — вспомогательная величина литерного типа.

Осталось обсудить методику нахождения запомненной зрителем карты.

Номер элемента массива “колода” с запомненной картой так и назовем — *номер_запомненной_карты*. С использованием массива *положение* искомое может быть рассчитано следующим образом:

```

номер_запомненной_карты :=
положение[3] * 9 +
положение[2] * 3 +
положение[1] + 1

```

С учетом того, что этапы 2–5 и 6–9 (см. выше) идентичны, всю программу можем оформить с использованием оператора цикла, параметром которого является номер расклада (имя величины — *расклад*):

```

алг Фокус
нач лит таб колода [1:27],
лит таб стопка1[1:9],
        стопка2[1:9], стопка3[1:9]
цел номер_запомненной_карты,
        номер_стопки, расклад, i,
цел таб положение[1:3], лит всп
|Исходная колода
колода[1] := "ДП"
колода[2] := "ТП"
... (см. выше)
|Первые два расклада
нц для расклад от 1 до 2
вывод нс, расклад, "-й расклад"
|Раскладываем карты на 3 стопки
|Формируем стопки
нц для i от 1 до 9
    стопка1[i] := колода[i * 3 - 2]
    стопка2[i] := колода[i * 3 - 1]
    стопка3[i] := колода[i * 3]
кц
|Выводим их
нц для i от 9 до 1 шаг -1
    вывод нс, стопка1[i], " ",
        стопка2[i], " ",
        стопка3[i]

кц
|Выведем также номера стопок
вывод нс, "1 2 3"
|Ввод номера стопки
вывод нс, "Укажите номер стопки
        с запомненной картой "
ввод номер_стопки
|Собираем стопки
нц для i от 1 до 9
    колода[i] := стопка1[10 - i]
кц
нц для i от 10 до 18
    колода[i] := стопка2[19 - i]
кц
нц для i от 19 до 27
    колода[i] := стопка3[28 - i]
кц
|Определяем, сколько стопок
|расположено над стопкой с запомненной
|картой
положение[расклад] := номер_стопки - 1
|Переворачиваем колоду
нц для i от 1 до 13
    всп := колода[i]

```

```

колода[i] := колода[27 - i + 1]
колода[27 - i + 1] := всп

```

кц

кц

```

|Третий расклад
вывод нс, "3-й расклад"
|Формируем стопки
... (см. выше)
|и выводим их
... (см. выше)
|Ввод номера стопки
... (см. выше)
|Собираем стопки
... (см. выше)
|Определяем, сколько стопок расположено
|над стопкой с запомненной картой
положение[3] := номер_стопки - 1
|Определяем и выводим искомую карту
номер_запомненной_карты :=
положение[3] * 9 + положение[2] * 3 +
положение[1] + 1
вывод нс, "Вы запомнили карту: ",
        колода[номер_запомненной_карты]
кон

```

Язык Паскаль

Uses CRT;

```

Var koloda: array [1..27] of string[2];
    stopka1, stopka2, stopka3:
array [1..9] of string[2];
    nomer_zapomnennoi_karti,
    nomer_stopki, rasklad, i: byte;
    pologenie: array [1..3] of byte;
    vsp: string[2];

```

BEGIN

```

clrscr;
{Исходная колода}
koloda[1] := 'ДП';
koloda[2] := 'ТП';
...
{Первые два расклада}
for rasklad := 1 to 2 do
    begin
        writeln(rasklad, '-й расклад');
        {Раскладываем карты на 3 стопки.
        Формируем стопки}
        for i := 1 to 9 do
            begin
                stopka1[i] := koloda[i * 3 - 2];
                stopka2[i] := koloda[i * 3 - 1];
                stopka3[i] := koloda[i * 3]
            end;
        {Выводим их}
        for i := 9 downto 1 do
            writeln(stopka1[i], ' ', stopka2[i],
                ' ', stopka3[i]);
        {Выведем также номера стопок}
        writeln ('1 2 3');
        {Ввод номера стопки}
    end

```

```

write('Укажите номер стопки
      с запомненной картой ');
readln(nomer_stopki);
{Собираем стопки}
for i := 1 to 9 do
  koloda[i] := stopka1[10 - i];
for i := 10 to 18 do
  koloda[i] := stopka2[19 - i];
for i := 19 to 27 do
  koloda[i] := stopka3[28 - i];
{Определяем, сколько стопок
  расположено над стопкой
  с запомненной картой}
pologenie[rasklad] := nomer_stopki - 1;
{Переворачиваем колоду}
for i := 1 to 13 do
  begin
    vsp := koloda[i];
    koloda[i] := koloda[27 - i + 1];
    koloda[27 - i + 1] := vsp;
  end;
clrscr
end;
{Третий расклад}
writeln('3-й расклад');
{Формируем стопки}

```

```

... (см. выше)
{и выводим их}
... (см. выше)
{Ввод номера стопки}
... (см. выше)
{Собираем стопки}
... (см. выше)
{Определяем, сколько стопок расположено
  над стопкой с запомненной картой}
pologenie[3] := nomer_stopki - 1;
clrscr;
{Определяем и выводим искомую карту}
nomer_zapomnennoi_karti :=
  pologenie[3] * 9 + pologenie[2] * 3 +
  pologenie[1] + 1;
writeln ('Вы запомнили карту: ',
        koloda[nomer_zapomnennoi_karti])
END.

```

Задания для самостоятельной работы учащихся

Разработайте варианты программы, в которых:

- 1) действия по формированию третьего разряда выполняются не отдельно, а в операторе цикла с параметром *расклад* (что сократит размер программы);
- 2) исходный набор карт формируется случайным образом.

Приложение 2

Оформление листов электронных таблиц для демонстрации фокусов

Фокусы “Волшебная таблица”, “Семь табличек” и “15 названий” можно демонстрировать, используя электронную таблицу Microsoft Excel или подобную программу. Так как оформление листа для этих фокусов аналогичное, то опишем его особенности на примере фокуса “Волшебная таблица”.

Оформим верхнюю часть листа так, как показано на рис. П2.1 (см. также рис. 1).

	A	B	C	D	E
1	18	10	6	6	5
2	19	11	7	7	7
3	20	12	12	10	9
...					
16	30	30	30	30	29
17	31	31	31	31	31
18	В каких столбцах имеется задуманное число:				
19					
20					
21	Вы задумали число:				

Рис. П2.1

После того как задумавший число сообщает, в каких столбцах таблицы оно имеется, следует в соответствующие ячейки диапазона A19:E19 внести некоторые отметки (это может сделать и сам задумавший число):

19	+		+		+
20					
21	Вы задумали число:			21	

В ячейке D21 выводится задуманное число.

Чтобы демонстрировать фокус, следует вне зоны видимости листа, например в строке 40, получить весомости каждого из пяти двоичных разрядов, причем делать это только в случае, когда в соответствующем разряде стоит отметка. Соответствующие формулы:

- в ячейке A40: =ЕСЛИ(НЕ(ЕПУСТО(A19));16);
- в ячейке B40: =ЕСЛИ(НЕ(ЕПУСТО(B19));8);
- в ячейке C40: =ЕСЛИ(НЕ(ЕПУСТО(C19));4);
- в ячейке D40: =ЕСЛИ(НЕ(ЕПУСТО(D19));2);
- в ячейке E40: =ЕСЛИ(НЕ(ЕПУСТО(E19));1).

Тогда задуманное число в ячейке D21 может быть найдено по формуле: =СУММ(A40:E40).

Однако при таком оформлении листа в ячейке D21 будет выводиться не только искомое значение, но и (по мере ввода отметок в ячейки A19:E19) все промежуточные результаты. Кроме того, после удаления всех отметок в ячейке D21 будет выводиться нулевое значение. Как можно устранить этот недостаток?

Можно сделать так, что значение в ячейке D21 будет выводиться только после заполнения ячейки D20:

18	В каких столбцах имеется задуманное число:				
19					
20	Отгадать Ваше число?			Да	
21	Вы задумали число:				

Соответствующая формула в ячейке D21:
 $\text{=ЕСЛИ(ЕПУСТО(D20);"";СУММ(A40:E40))}$.

Эффективным является вариант демонстрации рассматриваемого фокуса с использованием элементов управления — так называемых “переключателей”:

	A	B	C	D	E
1	18	10	6	6	5
2	19	11	7	7	7
3	20	12	12	10	9
...					
16	30	30	30	30	29
17	31	31	31	31	31
18	В каких столбцах имеется задуманное число:				
19	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
20	Отгадать Ваше число?			Да	
21	Вы задумали число:			21	

Рис. П2.2

Методика размещения их на листе зависит от версии программы. В любом случае после размещения следует:

- удалить надпись рядом с переключателем;
- установить связь каждого переключателя с той или иной ячейкой. Имеется в виду, что при выборе (включении) переключателя в связанной с ним ячейке будет представлено значение ИСТИНА (когда переключатель не выбран — ЛОЖЬ). Методика установления этой связи также зависит от версии программы (используется пункт **Формат элемента управления** или **Формат объекта**).

Если переключатели в ячейках A19:E19 связать соответственно с ячейками A39:E39, то вероятности каждого из пяти двоичных разрядов в ячейках A40:E40 вычисляются по формулам (соответственно):

=ЕСЛИ(A39;16) ;
 =ЕСЛИ(B39;8) ;
 =ЕСЛИ(C39;4) ;
 =ЕСЛИ(D39;2) ;
 =ЕСЛИ(E39;1) .

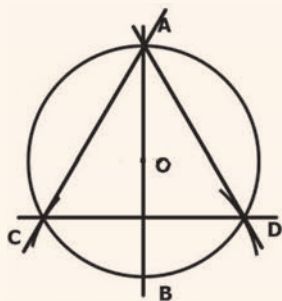
Задуманное число в ячейке D21 по-прежнему рассчитывается по формуле: =СУММ(A40:E40) .

Примечание. Для демонстрации фокуса “Семь табличек” можно использовать семь листов электронной таблицы.

Приложение 3

Алгоритм построения равностороннего треугольника

1. Нарисуйте окружность радиуса 2,5–3 см с центром в точке **O**:



2. Начертите отрезок прямой, проходящей через центр окружности и пересекающей ее в двух точках; пусть точки пересечения отрезка с окружностью — **A** и **B**.

3. Установите ножку циркуля с иглой в точку **B** и раствором циркуля, равным радиусу, сделайте засечки в точках **C** и **D**.

4. При помощи линейки попарно соедините точки **A**, **C** и **D** — отрезки **AC**, **AD** и **CD** будут являться сторонами равностороннего треугольника.



МИР ИНТЕРНЕТА

Как работают веб-серверы?

В предыдущем выпуске “В мир информатики” была опубликована статья “Как устроен и работает Интернет?”, в которой рассматривались вопросы происхождения Интернета и базовые технологии сети. Предлагаем новый материал, связанный с работой “скрытых” тружеников Интернета — веб-серверов.

В.А. Юнев,

эксперт по стратегическим технологиям,
Microsoft, Москва

► Важнейшими изобретениями, ставшими основой современной глобальной сети, являются технологии гипертекста и его передачи, а также интернет-браузеры — программы для просмотра веб-страниц.

Огромный вклад в изобретение и создание этих технологий внес Тим Бернерс-Ли — британский ученый, который работал над концепцией технологий Интернета в конце 80-х годов в Европейской организации по ядерным исследованиям.

Бернерс-Ли предложил глобальный гипертекстовый проект, в котором содержимое (контент) страниц, состоящее из информации разного типа — текста, изображений, других медиа-форматов, — соединялось бы друг с другом и само с собой посредством гиперссылок. Сегодня, когда мы щелкаем в веб-браузере на ссылку, мы не задумываемся над тем, как такая простая концепция появилась в реальной жизни. Кажется, что идея гиперссылок лежит на поверхности, однако, как и все подобные идеи, она потребовала времени на изобретение и воплощение в том практическом виде, в котором мы сейчас ее ис-



Тим Бернерс-Ли

пользуем. Проект объединения документов в сети с помощью гипертекста был назван Бернерсом-Ли “Всемирной паутиной” (World Wide Web).

Контент с гиперссылками потребовал новый формат представления информации, который был разработан в виде спецификации гипертекстового языка разметки, или HTML (*Hyper Text Markup Language*). Потребовалось изобрести и правила (или протоколы) обмена гипертекстовыми данными между компьютерами в сети. Таким протоколом стал стандарт протокола передачи гипертекста, или http (*Hyper Text Transfer Protocol*).

Бернерс-Ли и сегодня продолжает свою работу над Всемирной паутиной в качестве главы специальной независимой организации W3C (Консорциум Всемирной Паутины), которая разрабатывает и утверждает спецификации и стандарты технологий, используемых в глобальной сети.

Сервер — это любой компьютер, на котором запущено программное обеспечение, отвечающее на запросы на получение документов и других данных. Программы, которые запрашивают и отображают документы (такие, как браузер), называются клиентами. Термины “серверный” и “клиентский” применительно к функциям обозначают, на какой машине производится обработка. Клиентские функции выполняются на машине пользователя, серверные функции — на удаленной машине.

Д.Роббинс,
автор книги “Web-дизайн”

Протокол HTTP и запросы

Важнейшей частью современного Интернета является протокол обмена гипертекстовыми данными HTTP (*Hyper Text Transfer Protocol*). Этот протокол обладает массой положительных качеств, во главе которых стоит простота.

Вообще протокол — это правила обмена данными между двумя разными точками. Такими точками в Интернете являются ваш компьютер и компьютер в сети, с которого вы хотите получить информацию, например, веб-сайт *microsoft.com*, который содержит новости, ссылки на программы и другую информацию, представленную в гипертекстовом виде.

Протокол HTTP построен по принципу запрос – ответ. Клиентский компьютер посылает запрос в определенном формате (как правило, это специально оформленная строка текста), на что сервер или отвечающий компьютер в сети должен вернуть определенный ответ, понятный клиенту.

Каждый такой запрос клиента на сервер (и ответ сервера) состоит из трех частей:

- 1) стартовая строка;
- 2) заголовки;
- 3) тело сообщения.

Рассмотрим конкретный пример запроса. В ответ на ваше желание посетить сайт *microsoft.com* ваш браузер отправит запрос на сервер (компьютер в сети) примерно следующего содержания:

```
GET /en/us/default.aspx HTTP/1.1
HOST www.microsoft.com
Accept text/html,application/xhtml+xml,*/*
Accept-Language ru-RU
User-Agent Mozilla/5.0 (compatible; MSIE 9.0;
Windows NT 6.1; WOW64; Trident/5.0)
Accept-Encoding gzip, deflate
```

Здесь клиент (браузер, которым вы пользуетесь) сформировал шесть строк: первая стартовая строка и пять строк заголовков. Стартовая строка очень важна при обмене данными, она указывает на тип запроса, адрес необходимого содержимого и протокол, по которому клиент собирается принять данные.

Стартовая строка запроса

В приведенном выше примере клиент отправляет на сервер запрос типа GET с просьбой вернуть содержимое файла по адресу */en/us/default.aspx*, используя протокол передачи данных HTTP версии 1.1.

Что такое “запрос типа GET”? Запросы на сервер могут быть нескольких типов, тип определяется в стартовой строке. Перечислим самые значимые типы запросов: GET, POST, HEAD, PUT, DELETE. Назначение запроса сервер понимает из его типа. Например, запрос типа GET (получить) означает, что клиент просит передать ему определенное содержимое на сервере.

Другие запросы означают для сервера следующие действия:

— POST (отправить) — клиент отправляет с запросом некие данные (возможно, заполненную форму), которые необходимо обработать;

— HEAD (заголовок) — клиент просит вернуть ему одни только заголовки ответа, но не содержимое ресурса. Этот тип запроса используется для получения метаданных (сопутствующих данных) ресурса, которые могут использоваться вашим браузером для дальнейшей работы;

— PUT (поместить) — этот тип запроса похож на POST, но используется с целью добавить данные на удаленный ресурс;

— DELETE (удалить) — запрос такого типа предполагает удаление некоего содержимого на сервере.

Существует еще ряд типов запросов, но их использование не распространено.

Кроме типа запроса и адреса содержимого, которое мы запрашиваем с сервера, стартовая строка содержит еще один параметр — протокол передачи данных и его версию. В нашем случае этот параметр держит значение HTTP/1.1.

Протокол HTTP за свою долгую историю изменялся крайне редко и, это может показаться удивительным, имеет всего три версии: 0.9, 1.0 и 1.1. Объяснить такую стабильность протокола можно его простотой, заложенной изначально. Кроме того, такая фундаментальная вещь, как правила обмена информацией в глобальной сети, не должна меняться часто. И все же существуют разные версии протоколов HTTP, которые содержат минимальные различия между собой. Именно поэтому клиент и сервер обмениваются данными о версиях протоколов, которые поддерживают. Это позволяет найти клиентскому компьютеру и серверу в сети “общий язык”, который понятен обоим и на котором они могут разговаривать.

Заголовки запроса

Последующие данные запроса — это заголовки, которые также очень важны при обмене данными. Заголовки, в отличие от стартовой строки, могут иметь произвольные данные формата “ключ–значение”. В нашем случае Host, Accept, Accept-Language, User-Agent и Accept-Encoding — это ключи, а данные напротив — это значения ключей.

Для чего нужны заголовки? Заголовки — это возможность передать на сервер дополнительную информацию, которая корректирует запрос или добавляет в него условия. Заголовки могут рассказывать о клиенте серверу, чтобы тот предоставлял пользователю более точные данные.

Например, заголовок **Accept-Language**, который автоматически формирует браузер, рассказывает серверу о том, какой предпочтительный язык используется на компьютере пользователя (в нашем случае ru-RU — это русский). Имея такие данные от клиента, сервер может автоматически вернуть вместо англоязычной страницы русскоязычную страницу. А это, согласитесь, уже большая помощь пользователю.

Другие заголовки не менее важны и полезны:

— Host — определяет, относительно какого имени компьютера необходимо вернуть содержимое, в нашем случае мы запрашиваем содержимое */en/us/default.aspx* с сервера *www.microsoft.com*;

— Accept — определяет формат содержимого (контента), в котором мы ожидаем увидеть запрошенные данные. В нашем случае мы просим вернуть данные в формате *text/html, application/xhtml+xml, */**. Эти форматы данных понятны браузеру. Получив данные в таких форматах, браузер сможет отобразить их в понятном пользователю виде;

— User-Agent — содержит строку, в которой передается информация о вашем браузере. Эта информация важна серверу, поскольку из-за большого разнообразия браузеров иной раз контент, возвращаемый пользователю, требуется видоизменить. Особенно полезно бывает изменить контент, когда пользователь заходит на сайт с мобильного устройства (смартфона);

— Ассерт-Encoding — строка, которая говорит серверу о том, что клиент поддерживает возможность сжатия данных при передаче по определенным алгоритмам (gzip, deflate). Сжатие данных при передаче позволяет быстрее передать данные, и вы быстрее получите ответ (веб-страницу) от сервера. В нашем случае браузер сообщает серверу о том, что он готов принять данные в сжатом виде, а сервер, если он обладает функцией сжатия, теперь может вернуть сжатые данные.

Существует огромная масса других заголовков, которые могут быть переданы от клиента на сервер. Вы можете определять собственные заголовки, если уверены, что сервер сможет распознать их.

Ответ сервера и содержимое запроса (тело сообщения)

Последней частью сообщений, передаваемых между клиентом и сервером, является тело сообщения, или его содержимое. В случае GET-запроса тело сообщения может быть пустым, так как мы производим запрос и не передаем данные. Когда клиент формирует запросы другого типа, например POST или PUT, то тело сообщения содержит данные, которые клиент хочет отправить на сервер.

В ответ на наш GET-запрос сервер формирует данные, например, генерирует веб-страницу и возвращает ее нам вместе с сообщением, которое имеет тот же формат, что и запрос:

```
HTTP/1.1 200 OK
Cache-Control public
Content-Type text/html; charset=utf-8
Expires Mon, 14 Mar 2014 10:41:48 GMT
Last-Modified Mon, 14 Mar 2014 07:01:03 GMT
ETag 634356576630000000
Server Microsoft IIS/7.5
X-AspNet-Version 4.0.30319
```

Наверняка вы уже распознали структуру этих данных. Первой строкой идет стартовая строка, а затем заголовки сообщения. Вслед за этими данными, как правило, следует содержимое, которое мы запрашивали. В нашем случае это веб-страница /en/us/default.aspx. На рис. 1 представлено это содержимое, отображенное с помощью инструментов разработчика в браузере Internet Explorer 11.

Получив от сервера содержимое, браузер отображает его в понятном пользователю виде, например, в виде веб-страницы.

Вернемся к ответу сервера. Обратите внимание на стартовую строку: HTTP/1.1 200 OK. Вы можете определить, что первым параметром стартовой строки в ответе сервера идет протокол и его номер. Таким образом сервер в ответ на запрос сообщает клиенту, что он готов “разговаривать” на языке протокола HTTP версии 1.1.

Следующим за протоколом параметром является код состояния (результата запроса). Это очень важный параметр и свойство протокола HTTP. В ответ на запросы клиента сервер должен уметь сообщать о своем состоянии, о возможных ошибках при запросе, о статусе выполняемой операции. Таким сообщением для клиента являются коды состояния. В нашем случае сервер возвращает код 200. Это означает, что запрос завершился без ошибок. Однако существует большое количество других кодов статуса:

— 301 — код сообщает клиенту, что ресурс, который он запросил, был перемещен на новое место. Обычно в ответ на это клиент снова запрашивает содержимое, но уже с нового местоположения. Такой процесс называется “редиректом” (перенаправлением);

— 304 — код сообщает клиенту, что содержимое, которое он запросил, не изменилось со времени последнего запроса. В связи с этим клиент может использовать данные, которые он уже загрузил для повторного отображения, это увеличивает скорость отображения страниц и исключает ненужные запросы на сервер;

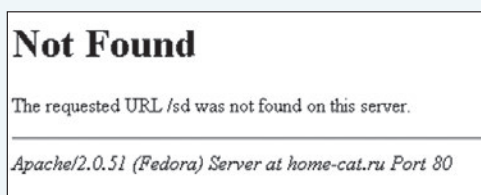
— 401 — код сообщает клиенту, что запрос требует идентификации, то есть передачи логина и пароля;

```
<!DOCTYPE html>
<html class="ui-mobile js flexbox canvas no-touch backgrounds cssanimations csstransforms csstransforms3d csstransitions fontface video audio svg inlinesvg" xmlns="http://www.w3.org/1999/xhtml">
  <head>...</head>
  <body class="mscom-hp-theme-layout" style="visibility: visible;">
    <div class="row-fluid" data-cols="1" data-view1="1" data-view2="1" data-view3="1" data-view4="1">...</div>
    <div class="row-fluid pagebody" data-cols="1" data-view1="1" data-view2="1" data-view3="1" data-view4="1">...</div>
    <iframe name="_bkframe" title="bk" id="__bkframe" src="javascript:void(0)" style="border: 0px currentColor; border-image: none; width: 0px; height: 0px;">...</iframe>
    <div class="row-fluid" data-cols="1" data-view1="1" data-view2="1" data-view3="1" data-view4="1">
      <div class="span bp0-col-1-1 bp1-col-1-1 bp2-col-1-1 bp3-col-1-1 mscom-remove-margin-left">
        <section class="row-grey">
          <div id="96ff19a3-2403-0d94-fcc9-eebb42d21e29">
            <div class="CSPvNext CMSvNextComp mscom-footer">...</div>
          </div>
        </section>
      </div>
    </div>
```

Рис. 1. Содержимое тела сообщения, полученное в ответ на запрос

— 403 — код сообщает клиенту, что такой запрос запрещен в связи с ограничениями прав пользователя;

— 404 — наверное, наиболее известный в Интернете код, который говорит клиенту о том, что ресурс, который он пытается запросить (например, веб-страница), на сервере отсутствует (страницу могли удалить или пользователь ошибся при вводе адреса);



— 500 — код сообщает клиенту, что при обработке запроса на сервере произошла ошибка и данные не могут быть возвращены. Такого рода коды показывают, что на сервере содержатся какие-то ошибки или на нем ведутся административные работы.

Существует огромное множество других кодов, которые используются реже. Полный перечень можно посмотреть, например, в Википедии.

Следующий за кодом статуса текст является сопроводительным к статусу. В нашем случае для кода 200 он имеет значение ОК, то есть “все хорошо”. Для кода с номером 304 такой текст будет иметь значение “Not modified”, то есть “не было изменено”.

Обратите внимание на заголовки сообщения, которые нам вернул сервер. Естественно, что они отличаются от тех заголовков, которые передавал

на сервер наш браузер. Давайте рассмотрим их подробнее.

Заголовки Cache-Control, Content-Type, Expires, Last-Modified и ETag отвечают за состояние контента, который был возвращен сервером. Заголовок Last-Modified содержит дату последнего изменения контента на сервере. Expires сообщает клиенту, когда контент будет считаться устаревшим. Все эти данные предназначены для повышения эффективности обмена данными через механизм кэширования. Кэширование — это процесс сохранения некоторого контента для повторного использования без загрузки с сервера. Если некоторое содержимое находится в кэше вашего браузера, то браузер будет стараться использовать его тогда, когда это возможно. В определении возможности использования контента из кэша браузеру помогают заголовки сервера, с которыми контент был получен. Например, основываясь на Expires, браузер может знать, когда следует запросить контент повторно.

Заголовок Server со значением Microsoft-IIS/7.5 сообщает клиенту о том, какой веб-сервер используется на сервере. В нашем случае таким веб-сервером является сервер IIS версии 7.5.

Последний из рассматриваемых заголовков, X-AspNet-Version, является нестандартным, то есть введенным компанией Microsoft самостоятельно для того, чтобы информировать клиента о том, какая версия платформы веб-разработки ASP.NET установлена на сервере. Эти сведения могут быть полезны при разработке разного программного обеспечения.

ЗАДАЧНИК

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

Задание “Три вопроса по музыке” (апрельский выпуск “В мир информатики”, рубрика “Поиск информации”)

Ответы

1. Самой подходящей музыкой для индийского фильма “Кинотанцор” является диско.

2. Из-за сакуры под снегом в Японию пригласили певца Сальваторе Адамо (в его репертуаре была песня “Падает снег”, ставшая очень популярной в Японии под названием “Сакура под снегом”).

3. В фильме “Ширли-Мырли” была использована музыка Вагнера, Моцарта, П.И. Чайковского, Мендельсона, Россини и других композиторов.

Ответы прислали:

— Барановская Татьяна и Жукова Ирина, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Бульбова Лидия, Волошин Марк, Гашимов Геннадий, Дощик Константин, Пешехонов Илья, Карякин Александр, Кобоскин Никита, Лазуренко Глеб, Лебедева Екатерина, Максимов Тимофей, Назаркина Татьяна, Остапчук Валентина, Петрова Алена, Сейфуллаев Руслан и Чунин Павел, г. Пионерский Калининградской обл., школа № 2, учитель **Багрова О.А.**;

— Волковец Владимир, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Галяпин Глеб, Калинина Ирина и Ларионова Юлия, г. Воронеж, лицей № 2, учитель **Комбаров С.И.**;

— Дибров Сергей, средняя школа поселка Осинковка, Алтайский край, учитель **Евдокимова А.И.**;

— Игнатьева Екатерина и Мезина Ольга, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Кириллова Л.Н.**;

— Казанец Елена, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Кошкарлова Анастасия и Милованова Дарья, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Лежнева Александра, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Лоскутова Светлана и Хомутова Евгения, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Новиков Сергей, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**

Кроссворд (апрельский выпуск “В мир информатики”)

Ответы

По горизонтали: 1. Скаляр. 3. Горнер. 7. Дискета. 8. Мега. 9. Формат. 10. Кобол. 15. Тувинка. 16. Начало. 18. Цифра. 20. Три. 21. Квант. 22. Диаграмма. 25. Один. 26. Байт. 27. Бумага. 28. Тренога. 29. Сигнал.

По вертикали: 1. Сегмент. 2. Лада. 3. Граф. 4. Норма. 5. Растр. 6. Скобка. 10. Конец. 11. Лента. 12. Евклид. 13. Ячейка. 14. Колонтитул. 17. Атрибут. 19. Формула. 23. Истина. 24. Магнит.

Ответы прислали:

— Акулич Станислав, средняя школа поселка Осинька, Алтайский край, учитель **Евдокимова А.И.**;

— Бульбова Лидия, Кратик Александр и Назаркина Татьяна, г. Пионерский Калининградской обл., школа № 2, учитель **Багрова О.А.**;

— Герасимов Кирилл, Гумбатов Туркан, Денисов Илья, Корвель Анастасия, Куга Яна, Луист Татьяна, Тикка Виктория, Трубина Алина, Фомина Анастасия, Худенко Игорь и Шумская Софья, Республика Карелия, г. Сегежа, школа № 5, учитель **Меньшиков В.В.**;

— Землянская Надежда, Челябинская обл., г. Златоуст, школа № 9, учитель **Мусатова И.Б.**;

— Зубов Владислав, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Иванов Николай, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Игнатьева Екатерина и Мезина Ольга, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Кириллова Л.Н.**;

— Мазанова Екатерина, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Мищенко Маргарита, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнеева М.В.**;

— Машонина Ирина, г. Фрязино Московской обл., школа № 4, учитель **Сенюта Е.И.**;

— Кошкарлова Анастасия и Милованова Дарья, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

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

Судоку (апрельский выпуск “В мир информатики”)

Ответы представили:

— Айзетулова Ландыш, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Кириллова Л.Н.**;

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

— Бульбова Лидия, Кобоскин Никита и Польский Марк, г. Пионерский Калининградской обл., школа № 2, учитель **Багрова О.А.**;

— Зубов Владислав, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Иванов Николай, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Кошкарлова Анастасия, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Краснова Диана, Свердловская обл., г. Ревда, школа № 10, учитель **Игошева А.А.**;

— Машонина Ирина, г. Фрязино Московской обл., школа № 4, учитель **Сенюта Е.И.**;

— Шадрин Алина и Шигаев Никита, Челябинская обл., г. Златоуст, школа № 9, учитель **Мусатова И.Б.**

Ответы на вопросы 3-й части викторины «Да будет “свет”!» прислали:

— Барановская Татьяна и Жукова Ирина, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Галяпин Глеб, Калинина Ирина и Ларионова Юлия, г. Воронеж, лицей № 2, учитель **Комбаров С.И.**;

— Дибров Сергей, средняя школа поселка Осинька, Алтайский край, учитель **Евдокимова А.И.**;

— Жилкина Владислава, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Игнатьева Екатерина и Мезина Ольга, Чувашская Республика, г. Канаш, Канашский педагогический колледж, преподаватель **Кириллова Л.Н.**;

— Кошкарлова Анастасия и Милованова Дарья, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Лежнева Александра, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Лоскутова Светлана и Хомутова Евгения, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**

Решения головоломок “судоку”, опубликованных в майском выпуске “В мир информатики”, прислали:

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

— Бульбова Лидия и Сушко Ирина, г. Пионерский Калининградской обл., школа № 2, учитель **Багрова О.А.**;

— Димакова Арина и Одинцова Екатерина, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Иванов Николай, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Козин Василий, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Малыгин Илья и Муравьев Иван, г. Рязань, школа № 44, учитель **Марцинкевич Е.Е.**;

— Стороженко Степан, г. Пенза, школа № 512, учитель **Гаврилова М.И.**

Задание “Три вопроса” (майский выпуск “В мир информатики”, рубрика “Поиск информации”)

Ответы

1. Герой фильма “Берегись автомобиля”, которого играл Олег Николаевич Ефремов, исполнял роль Максима Подберезовикова.

2. Предмет одежды, который чаще всего можно встретить на портретах Анны Андреевны Ахматовой, — шаль.

3. На выставке 1895 года в Мюнхене была впервые показана собака породы боксер (немецкий боксер).

Ответы представили:

— Бородюк Анна и Василенко Татьяна, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Бульбова Лидия, Дощик Константин, Лазуренко Глеб, Максимов Тимофей, Назаркина Татьяна, Сейфуллаев Руслан, Сушко Ирина и Чунин Павел, г. Пионерский Калининградской обл., школа № 2, учитель **Багрова О.А.**;

— Дибров Сергей, средняя школа поселка Осинька, Алтайский край, учитель **Евдокимова А.И.**;

— Димакова Арина и Одинцова Екатерина, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**;

— Жукова Ирина, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Леженников Тарас, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнеева М.В.**;

— Лоскутова Светлана и Хомутова Евгения, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Новиков Сергей, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**

Решение головоломки “Как получить верное равенство?” прислала также Мищенко Маргарита, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнеева М.В.** Маргарита представила ответы к кроссворду из мартовского выпуска и правильное решение головоломки “Все цифры в квадратах”. Как и ряд других читателей, принявших активное участие в решении кроссвордов, она будет награждена дипломом. Поздравляем!

Тарас Леженников из той же школы прислал правильные решения задач “Очередь в музей” и “Разлив бензина”.

Разбор задачи “Дома в Зеленом городе” и числового ребуса со словом-числом **ЭКРАН** будет проведен в следующем выпуске.

Яблоки

После того как каждое из 52 яблок разрезали пополам, стало 134 половинок. В какой системе счисления велся счет при этом?

Кто сказал правду?

Билет на проезд в общественном транспорте считается счастливым, если в его шестизначном номере сумма первых трёх цифр равна сумме последних трёх цифр.

Как-то между тремя приятелями состоялся такой разговор:

— Однажды мне попался счастливый билет, у которого каждая цифра, начиная со второй, была либо вдвое больше, либо вдвое меньше предыдущей, — заявил Петя.

— А мне, помню, достался счастливый билет, у которого каждая цифра, начиная со второй, была либо вдвое больше, либо втрое меньше предыдущей, — сообщил Коля.

— А у моего счастливого билета каждая цифра, начиная со второй, была либо вдвое больше, либо вчетверо меньше предыдущей, — сказал Вася.

Чьи слова могли быть правдой?

Кодирование чисел

Для кодирования натуральных чисел с помощью буквенных последовательностей был предложен следующий принцип, основанный на использовании латинских букв: А, В, С и D.

Числам 1, 2, 3 и 4 ставятся в соответствие указанные четыре буквы.

Последующим 16 числам ставятся в соответствие двухбуквенные коды в следующем порядке: 5=AA, 6=AB, 7=AC, 8=AD, 9=BA, 10=BB, ..., 18=DB, 19=DC, 20=DD. Аналогично для последующих чисел используются трехбуквенные коды (от 21=AAA до 84=DDD), четырехбуквенные и т.д. Укажите буквенный код числа 295.

В ответе нужно записать последовательность из латинских букв. Решение получить, не выписывая все 295 чисел. Вспомогательные расчеты желательно провести с помощью электронной таблицы Microsoft Excel или др.

Автор задачи — Сергей Котляров

Обсуждение ответа

Четверо ребят обсуждали ответ к задаче. Коля сказал: “Это число 9”. Роман: “Это простое число”. Катя: “Это четное число”. А Наташа сказала, что это число — 15.

Назовите это число, если и девочки, и мальчики ошиблись ровно по одному разу.

Банкир и конверты

Некому банкиру нужно было встретиться с важным клиентом, которому он должен выдать наличными заранее неизвестную сумму от 1 до 1 000 000 у.е. Чтобы не тратить время на отсчитывание денег, банкир дал указание своим кассирам заготовить некото-

рое количество конвертов с деньгами, на которых написаны содержащиеся в них суммы, чтобы потом просто отдать клиенту один или несколько конвертов, в которых и будет содержаться требуемая им сумма.

MICROSOFT EXCEL УГЛУБЛЕННО

Работа с датами в электронных таблицах

В электронной таблице Microsoft Excel и других подобных программах можно работать не только с числами и текстами, но и с датами. Даты можно сравнивать между собой, складывать и вычитать, а также использовать в других вычислениях. Например, можно вычислить число дней между двумя датами, определить, какой день недели приходился на ту или иную дату, и т.п. В данном разделе будут описаны соответствующие возможности электронных таблиц.

Ввод значений даты

Для того чтобы ввести в ячейку дату, следует указать номер дня, номер месяца и две последние цифры года через точку (12.12.87), дефис (12-12-87) или символ “/” (12/12/87). Можно вводить также первые три буквы названия месяца (12-дек-87 и т.п.; для даты в мае месяце необходимо написать слово *май*). Текущий год можно не указывать — он будет добавлен к введенной дате автоматически. При вводе значений даты происходит их автоматическое распознавание, и общий формат ячейки¹ заменяется на встроенный формат даты. Так, если ввести, например, значение 12-12-87 или 12 дек 87, то в ячейке отобразится 12.12.87², а в строке формул для данной ячейки будет выведено: 12.12.1987. Но если в ячейке указать 22.10.28, то в строке формул вместо ожидаемой даты 22.10.1928 вы увидите другую — 22.10.2028. Дело в том, что, если при вводе даты указаны только две последние цифры года, Microsoft Excel добавит первые две цифры по следующим правилам:

— если введенное число лежит в интервале от 00 до 29, то оно интерпретируется как год с 2000-го по 2029-й;

— если введенное число лежит в интервале от 30 до 99, то оно интерпретируется как год с 1930-го по 1999-й.

¹ Если в ячейке не установлен какой-либо специальный формат (числовой, процентный, финансовый, формат дат и т.д.), то данные в ней выводятся в так называемом “общем” формате, используемом для отображения как текстовых, так и числовых значений различного типа.

² Если такой формат установлен в операционной системе Windows. Изменение установок осуществляется с помощью Панели управления (кнопка Пуск, пункт Настройка), пиктограмма Язык и стандарты, вкладка Дата, поле Краткий формат даты.

1. Какое наименьшее количество конвертов необходимо иметь?
2. Какова будет в этом случае полная сумма во всех конвертах?
3. Как, используя подготовленные конверты, набрать требуемую сумму?

Таким образом фирма Microsoft в свое время позаботилась о переходе в третье тысячелетие. Поэтому года с 1900-го по 1929-й следует указывать полностью.

По умолчанию значения даты выравниваются в ячейке по правому краю. Если не происходит автоматического распознавания формата даты, то введенные значения интерпретируются как текст, который выравнивается в ячейке по левому краю.

Представление дат в ячейках

Формат представления даты в ячейке, отображаемый после ввода значения, может быть изменен следующим образом:

- 1) на вкладке Главная в группе Ячейки (см. рис. 1) щелкнуть на кнопке Формат и выбрать пункт Формат ячеек;
- 2) в появившемся окне Формат ячеек (см. рис. 2) на вкладке Число в разделе Числовые форматы выбрать пункт Дата;
- 3) после выбора нужного типа щелкнуть на кнопке ОК.

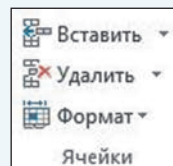


Рис. 1

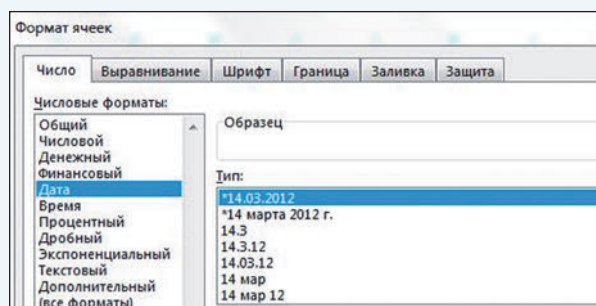


Рис. 2

Так, вместо значения 05.12.87 можно получить 5.12.87; 5 дек 87; 5 Декабрь, 1987 и другие представления одной и той же даты. При этом значение даты, отображаемое в строке формул, не меняет-

ся (оно не зависит от формата ее представления в ячейке).

Действия с датами

Как уже отмечалось, даты можно складывать и вычитать, сравнивать между собой. Можно также умножать и делить их на числа! Для того чтобы понять, как это реализуется, необходимо разобраться, как хранятся даты в компьютере. Введем в ячейку A1 дату 1 января 1900 года (напомним, что для этого следует ввести 1-1-1900, а не 1-1-00). С помощью маркера заполнения распространим (скопируем) введенное значение на ячейки A2: A10 (в них появятся даты, в которых будут значения, соответствующие 2, 3, ..., 10 января 1900 года). Скопируем диапазон ячеек A1:A10 в B1:B10. Изменим формат представления данных в блоке B1:B10 на *Общий* (см. рис. 2). Мы увидим, что в этом диапазоне появятся значения 1, 2, ..., 10. Из этого можно сделать важный вывод: дата в электронных таблицах — это количество дней, прошедших от 1 января 1900 года. Такая форма внутреннего представления дат и позволяет выполнять над ними различные арифметические операции и операции сравнения.

Основные функции для работы с датами³

В программе Excel имеется ряд функций для работы с датами. Рассмотрим некоторые из них.

1. Функции ДЕНЬ, МЕСЯЦ и ГОД

Эти функции возвращают, соответственно, номер дня в месяце, номер месяца в году и год для некоторой даты.

Их синтаксис: ДЕНЬ(дата), МЕСЯЦ(дата) и ГОД(дата),

— где аргумент дата — адрес ячейки, содержащей дату, либо дата, заданная в общем или числовом формате (12345), либо как текст (например, "15-4-93" или "15-Апр-1993").

День возвращается как целое число в диапазоне от 1 до 31. Месяц определяется как целое в интервале от 1 (Январь) до 12 (Декабрь). Значение года возвращается как целое число в интервале 1900–9999.

Примеры

1. Если в ячейке A2 указана дата 26.10.49, то ДЕНЬ(A2) равняется 26, МЕСЯЦ(A2) равняется 10, ГОД(A2) равняется 1949.

2. ДЕНЬ("4-Янв") равняется 4, МЕСЯЦ("4-Янв") равняется 1.

3. ДЕНЬ("15-Апр-1993") равняется 15, МЕСЯЦ("15-Апр-1993") равняется 4, ГОД("15-Апр-1993") равняется 1993.

4. ДЕНЬ("11.8.93") равняется 11, МЕСЯЦ("11.8.93") равняется 8, ГОД("11.8.93") равняется 1993.

³ Таблица соответствия имен функций в электронных таблицах Microsoft Excel и Oracle OpenOffice.org Calc приведена в конце статьи.

2. Функция ДЕНЬНЕД

Функция возвращает номер дня недели, соответствующий некоторой дате. Ее синтаксис: ДЕНЬНЕД(дата; тип),

— где дата — аргумент, аналогичный используемому в описанных выше функциях;

тип — число, которое определяет вариант возвращаемых значений:

Тип	Возвращаемые значения
1 или опущен	от 1 (Воскресенье) до 7 (Суббота)
2	от 1 (Понедельник) до 7 (Воскресенье)
3	от 0 (Понедельник) до 6 (Воскресенье)

Примеры

1) Если в ячейке A2 указана дата 26.10.49, то ДЕНЬНЕД(A2) равняется 3 (Среда).

2) ДЕНЬНЕД("15.2.90") равняется 5 (Четверг).

3) ДЕНЬНЕД("15.2.90"; 2) равняется 4 (Четверг).

3. Функция СЕГОДНЯ

Функция возвращает дату текущего дня, отслеживаемую компьютером. Ее синтаксис: СЕГОДНЯ () — без аргументов, но с обязательными скобками.

4. Функция ДАТА

Функция позволяет "собрать" дату из значений года, номера месяца и номера дня. Ее синтаксис: ДАТА(год; месяц; день), где:

— год — это число от 1900 до 2078;

— месяц — это число, представляющее номер месяца в году;

— день — это число, представляющее номер дня в месяце.

Например, ДАТА(45; 5; 9) есть 9 мая 1945 года.

Задачи для самостоятельной работы

1. В ячейках B2 и B3 получить число 37 135 (указанное число ни в одну из ячеек не вводить):

	A	B	C
1			
2		01.сен.01	
3		01.09.01	
4			

2. В ячейках A2 и A3 получить число 18 197 (указанное число ни в одну из ячеек не вводить):

	A	B	C
1			
2	26 Октябрь, 1949		
3	26.10.49		
4			

3. В ячейке B2 будет записана некоторая дата. Получить в ячейках B3:B5, соответственно, номер дня в месяце, номер месяца и год этой даты:

	A	B	C
1			
2	Введите дату →		
3	Число в этой дате:		
4	Месяц в этой дате:		
5	Год в этой дате:		
6			

4. Оформите лист для определения года, номера месяца и номера дня рождения:

	A	B	C	D
1				
2	Введите дату вашего рождения →			
3	Вы родились в		году	
4	Номер месяца вашего дня рождения:			
5	Вы родились		числа	
6				

Искомые значения должны быть получены в ячейках В3:В5.

5. Сотрудники отдела кадров обычно подсчитывают стаж работы на предприятии следующим образом. Выписывается текущая дата в виде 2007, 27 сентября, а под ней — дата начала работы работника на этом предприятии в аналогичном виде. Затем попарно вычитаются значения года, номера месяцев и номера дней в месяце. Например, если работник начал работать на предприятии 19 мая 2003 года, то его стаж работы составляет 4 года 8 месяцев и 6 дней. Оформите лист для расчета стажа работы по описанной методике с использованием данных типа Дата. Принять, что номер месяца и номер текущего дня больше соответствующих значений момента поступления на работу.

6. По дате, указанной в ячейке, определить номер дня недели, на который приходилась эта дата (понедельник — 1, вторник — 2, ..., воскресенье — 7).

7. Научный сотрудник забыл точную дату конференции, на которой ему необходимо присутствовать, но помнит, что она должна начаться в четверг в период с 1 по 8 октября 2007 года. Помогите ему определить точную дату начала конференции.

8. В ячейке В5 будет записана некоторая дата. В ячейке В3 получить дату дня, который будет через 100 дней после указанной даты:

	A	B	C
1			
2	Введите дату →		
3	Дата через 100 дней после указанной:		
4			

9. В ячейке В2 будет записана некоторая дата. В ячейке В3 получить дату дня, который был за 200 дней до указанной даты.

	A	B	C
1			
2	Введите дату →		
3	Дата за 200 дней до указанной:		
4			

10. В ячейке В2 получить дату текущего дня, в ячейке В4 — номер дня недели (понедельник — 1,

вторник — 2, ..., воскресенье — 7), который будет через некоторое число дней после текущего дня (это число будет указано в ячейке В3):

	A	B	C
1			
2	Дата сегодняшнего дня:		
3	Количество дней после сегодняшнего:		
4	Искомый номер дня:		
5			

11. В ячейке В2 получить дату текущего дня, в ячейке В4 — номер дня недели (понедельник — 1, вторник — 2, ..., воскресенье — 7), который был за некоторое число дней до текущего дня (это число будет указано в ячейке В3):

	A	B	C
1			
2	Дата сегодняшнего дня:		
3	Количество дней до сегодняшнего:		
4	Искомый номер дня:		
5			

12. В ячейке В2 и В3 будут указаны даты двух событий. Определить, сколько дней прошло между этими событиями.

13. Определите свой возраст в днях и неделях.

14. Для текущей даты вычислить:

- а) порядковый номер дня с начала года;
- б) сколько дней осталось до конца года.

15. Определить, сколько дней длится первое полугодие года и сколько — второе.

16. В ячейке В2 указана дата некоторого события, произошедшего в первой половине XX века. Необходимо в ячейке В3 получить дату дня, до которого от 1 января 1900 года прошло в два раза больше дней, чем от 1 января 1900 года до дня данного события.

17. В ячейке В2 запишите дату вашего рождения, а в ячейке В3 получите дату текущего дня. Определить дату того дня, когда число дней вашей жизни станет в два раза больше, чем число прожитых дней до текущего дня. Дату получить в формате вида “12 Апрель, 2017”.

18. Известна дата рождения Пети. Определить дату рождения Коли, если известно, что число дней, прожитых им до текущего дня, в два раза меньше, чем число дней, прожитых Петей.

19. В ячейке В2 запишите дату вашего рождения, а в ячейке В3 получите дату текущего дня. Определить номера дней недели (понедельник — 1, вторник — 2, ..., воскресенье — 7), которые будут, когда число дней вашей жизни станет в 2, 3, 4 и 5 раз больше, чем число прожитых дней до текущего дня.

20. После того как в ячейки В2 и В3 будут введены даты двух событий, определить, какое событие произошло раньше.

21. Производственное совещание проходит по вторникам и пятницам. Составьте их расписание на февраль 2015 года в виде:

	A	B	C	D	...
1	Дата	1 фев	2 фев	3 фев	
2	Совещание				
3					

— где в строке 2 в ячейках, соответствующих вторникам и пятницам, должен быть указан какой-нибудь символ (“С”, “+” или т.п.).

22. На листе представлены сведения о дате рождения учеников класса:

	A	B	C	D
1				
2	№	Фамилия, имя	Дата рождения	
3	1	Азаров Павел		
4	2	Бородянская Анна		
5	3	Войлуков Кирилл		
...				
27	25	Ющенко Мария		
28				

В диапазоне ячеек D3: D27 поставить знак “+” для тех учеников, дата рождения которых:

- приходится на среду;
- приходится на 10-е число месяца;
- приходится на август.

23. В ячейке B2 указана дата некоторого события. В ячейке B3 получить дату дня, который будет через три года после этого события.

24. В ячейке B2 указана дата некоторого события. В ячейке B3 получить дату дня, который был за пять месяцев до этого события.

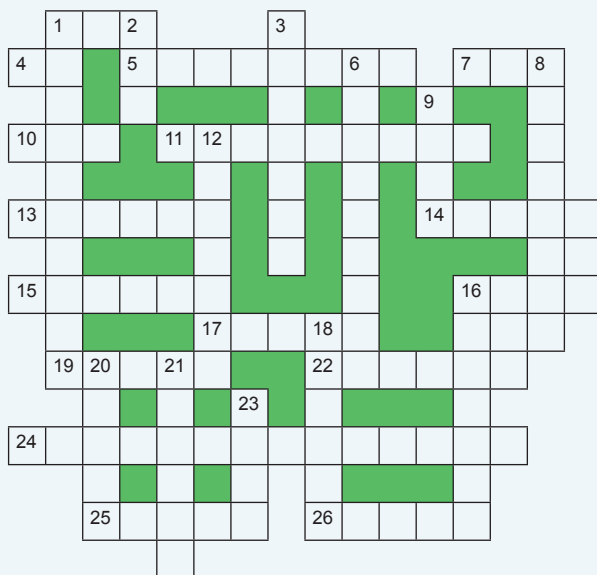
25. В ячейке B2 указана дата некоторого события. В ячейке B3 получить дату дня, который будет через n лет, m месяцев и k дней после этого события. Значения n , m и k вводятся в отдельные ячейки.

Таблица соответствия имен функций в электронных таблицах Microsoft Excel и Oracle OpenOffice.org Calc

Microsoft Excel	Oracle OpenOffice.org Calc
ДЕНЬ	DAY
МЕСЯЦ	MONTH
ГОД	YEAR
ДЕНЬНЕД	WEEKDAY
ДАТА	DATE

“ЛОМАЕМ” ГОЛОВУ

Решите, пожалуйста, кроссворд.



По горизонтали

- Цифра восьмеричной системы счисления.
- Крик, громкий, с криком, разговор.
- Составная часть управления объектами и процессами, заключающаяся в наблюдении за объектом с целью проверки соответствия наблюдаемого состояния объекта желаемому и необходимому состоянию.
- В программировании — характеристика величины, определяющая множество ее допустимых значений и применимых к ней операций.

- Нижняя граница стека.
- “Мозг” компьютера.
- Форма взаимодействия пользователя с компьютером.
- Просмотр регистров состояний группы устройств, а также выявление точки зрения группы людей.
- Положение, состояние.
- Семейство совместимых друг с другом компьютеров.
- Так называют часть оператора цикла — операторы, повторяемые при его выполнении.
- Совокупность точек графического изображения.
- Структура, содержание.
- Процесс разметки диска на участки, подготовка его к записи информации.
- Совокупность характеристик символа или абзаца.

26. Цирковая...

По вертикали

- Электронный прибор на полупроводниковом кристалле — элемент микросхемы.
- Буква греческого алфавита, которой, как правило, обозначают неизвестную величину.
- $\frac{1}{100}$.
- Распечатка текста программы.
- Совокупность правил, регламентирующих формат и процедуры обмена информацией между двумя или несколькими устройствами или процессами.

9. Французский инженер, изобретатель системы кодировки символов для телетайпов, в честь которого названа единица скорости передачи информации.

12. Место хранения информации в процессоре.

16. Элемент стандартного устройства для ввода информации в компьютер.

18. Главная часть какого-либо предмета, служащая его опорой, основанием.

20. Координаты ячейки в электронной таблице.

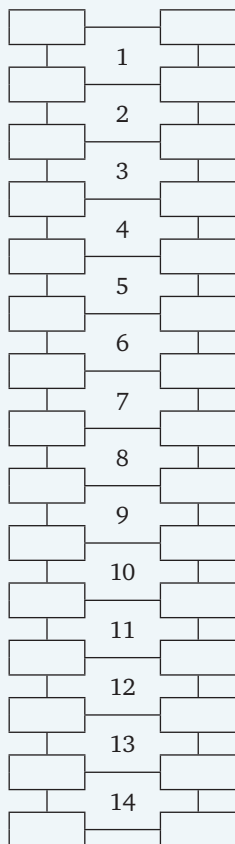
21. Так называют информацию, которая передается по Интернету.

23. ... да гладь.

Слова в квадратах

Впишите в квадратики буквы приведенных ниже терминов вокруг номеров так, чтобы их можно было прочесть (читать термины можно как по часовой стрелке, так и против).

- | | |
|----------|-------------|
| 1. Ввод. | 8. Клон. |
| 2. Анод. | 9. Тело. |
| 3. Шина. | 10. Стекло. |
| 4. Шифр. | 11. Диск. |
| 5. Фрак. | 12. Дизель. |
| 6. Абак. | 13. Зеро. |
| 7. Банк. | 14. Порт. |



Ответы (можно не все) присылайте в редакцию.

Числовой ребус

Решите, пожалуйста, числовой ребус:

$$\text{КАПЛЯ} + \text{КАПЛЯ} + \text{КАПЛЯ} = \text{ОЗЕРКО}$$

Как принято в таких головоломках, в ней одинаковыми буквами зашифрованы одинаковые цифры, разными буквами — разные цифры. Еще раз обратим внимание на то, что редакция имеет в виду решение числовых ребусов методом рассуждений.

Неисправный калькулятор

На калькуляторе цифры изображаются с помощью так называемых “7-сегментных” индикаторов:



Представьте себе калькулятор, на котором отображаются оба числа, над которыми проводится действие, а также результат:



Но этот калькулятор сломался — у него горят не все сегменты. Сможете ли вы определить, с какими числами выполнялось умножение? Не горят (а должны) 25 сегментов.

Сколько раз встречается слово?

Сколько раз встречается некоторое слово-существительное в таблице с буквами? Слова можно читать в любом направлении (включая обратные и диагональные), но обязательно по прямой и без разрывов.

А	А	А	А	А	А
Ш	Ш	Ш	Ш	Ш	К
А	А	А	А	К	К
К	К	К	А	Ш	А
К	А	Ш	А	К	Ш
А	А	К	А	Ш	А

Числовые ребусы в троичной системе

В приведенных ниже ребусах зашифрованы числа, записанные в троичной системе счисления. Одинаковым буквам соответствуют одинаковые цифры. Звездочкой (“*”) может быть любая цифра.

1.	2.	3.
$\begin{array}{r} \text{M} \\ + \text{N N} \\ \hline * * \end{array}$	$\begin{array}{r} * * \\ + * * \\ \hline * 0 \end{array}$	$\begin{array}{r} \text{B} \\ + \text{A *} \\ \hline \text{A 1} \end{array}$

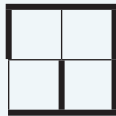
Ответы присылайте в редакцию (можно решать не все ребусы).

Робот в лабиринте

Вы, конечно, знаете об исполнителе Робот. Как известно, он перемещается по клетчатому полю,

на котором могут быть стены. Он понимает четыре команды: вверх, вправо, вниз и влево — и, исполняя их, смещается на одну клетку в соответствующем направлении, если ему ничего не мешает, и ничего не делает — если мешает стена⁴.

Представьте себе, что Робот находится в лабиринте с единственным выходом:

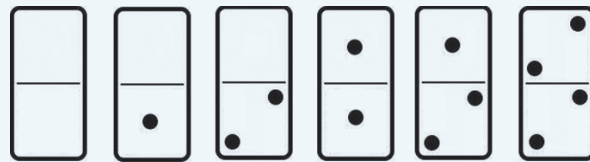


Но сам лабиринт вы не видите.

Ваша задача — дать Роботу такую последовательность команд, по которой он вышел из лабиринта, где бы ни находился. Сможете? (Положение Робота в лабиринте неизвестно.)

Шесть костяшек домино

Можно ли из шести костяшек домино (см. рисунок) сложить прямоугольник 3×4 так, чтобы во всех трех строчках точек на костяшках было поровну и во всех четырех столбцах точек тоже было поровну?



Задание предназначено для учеников 1–7-х классов.

Автор задачи — Наталья Стрелкова

КРЕПКИЙ ОРЕШЕК

В этой рубрике, как обычно, мы рассматриваем задачи, решение которых вызвало трудности.



значений цифр восьми букв-цифр требует рассмотрения 81 миллиона вариантов!).

Кроме Тараса, ответы представили:
— Журавлева Анастасия, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;

— Одинцова Екатерина, г. Челябинск, школа № 124, учитель **Юртаева Г.Ю.**

Редакция благодарит всех перечисленных читателей. При этом еще раз обратим внимание на то, что, публикуя числовые ребусы, мы имеем в виду их решение путем рассуждений.

Приведем начало анализа указанного ребуса.

Прежде всего ясно, что $C \geq 3$.

Учитывая это, исследуем разряды десятков и сотен и найдем сочетания значений цифр **Т** и **С**, дающих в результате одно и то же значение **М** (см. табл. внизу).

Сравнив возможные значения **М** в столбцах 7–13 и в столбцах 3–5, можно установить, что они могут быть равны при следующих сочетаниях цифр **Т** и **С**

Числовой ребус “Трижды ШЕСТЬ”

Напомним, что требовалось решить числовой ребус:

$$\begin{array}{r} \text{Ш Е С Т Ь} \\ + \text{Ш Е С Т Ь} \\ \text{Ш Е С Т Ь} \\ \hline \text{С У М М А} \end{array}$$

Тарас Леженников, Краснодарский край, г. Приморско-Ахтарск, школа № 22 (учитель **Корнеева М.В.**), решил ребус, разработав компьютерную программу. Как он сам пишет, программа имеет недостаток — выполняется долго в связи с большим числом комбинаций (полный перебор возможных

Т	ЗТ	Цифра М			“В уме” из разряда десятков перейдет:	Цифра М						
		при переносе “в уме” из разряда единиц 0, 1 или 2				При значениях С, равных:						
		0	1	2		3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	0	1	2	0	9	2	5	8	1	4	7
1	3	3	4	5	0	9	2	5	8	1	4	7
2	6	6	7	8	0	9	2	5	8	1	4	7
3	9	9	0	1	0 или 1	—	2/3	5/6	8/9	1/2	4/5	7/8
4	12	2	3	4	1	0	—	6	9	2	5	8
5	15	5	6	7	1	0	3	—	9	2	5	8
6	18	8	9	0	1 или 2	0/1	3/4	6/7	—	2/3	5/6	8/9
7	21	1	2	3	2	1	4	7	0	—	6	9
8	24	4	5	6	2	1	4	7	0	3	—	9
9	27	7	8	9	2	1	4	7	0	3	6	—

⁴ У “настоящего” Робота из учебников и учебных компьютерных программ команда Роботу в направлении, закрытом стеной, считается ошибкой.

(невозможные значения **М** в столбцах 7–13 зачеркнуты):

- 1) T = 0, C = 4; 8) T = 3, C = 7;
- 2) T = 0, C = 7; 9) T = 4, C = 7;
- 3) T = 1, C = 5; 10) T = 6, C = 3;
- 4) T = 1, C = 8; 11) T = 6, C = 9;
- 5) T = 2, C = 6; 12) T = 7, C = 3;
- 6) T = 2, C = 9; 13) T = 8, C = 4;
- 7) T = 3, C = 6; 14) T = 9, C = 5.

Исследовав затем эти 14 (всего лишь!) вариантов, можно найти решение ребуса. Предлагаем читателям сделать это. Приславшие правильный ответ будут награждены дипломами.

Задача “Необычный маршрут”

Напомним условие: “Митя вышел из точки А плоской равнины и прошел 1 м на юг, 2 м — на запад, 4 м — на север, 6 м — на восток, 7 м — на юг, 8 м — на запад, 10 м — на север, 12 м — на восток, 13 м — на юг, 14 м — на запад, 16 м — на север, 18 м — на восток и т.д.

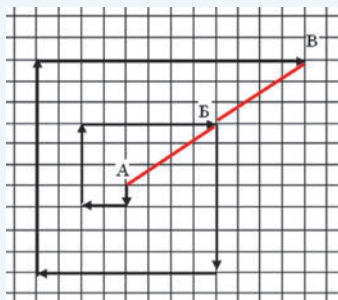
На каком расстоянии от точки А будет Митя после 100 “шагов” (после прохождения 100 отрезков)?

Какова длина последнего, 100-го отрезка пути?

Какой общий путь пройдет Митя за 100 “шагов”?”.

Благодарим Анастасию Журавлеву, г. Воронеж, лицей № 2 (учитель **Комбарова С.И.**), приславшую правильные ответы, и приводим начало анализа.

Подготовим схему маршрута “на клетчатой бумаге”:



Видно, что за первые четыре “шага” Митя окажется в точке **Б**. Анализ показывает, что отрезок **АВ** представляет собой гипотенузу прямоугольного треугольника, катеты которого равны 3 и 4 м. По теореме Пифагора длина такой гипотенузы равна 5 м. После восьми шагов (в точке **В**) мальчик будет находиться от точки **Б** на расстоянии 5 м. Аналогичное смещение имеет место за каждые следующие четыре “шага”. Значит, после 100 шагов смещение от точки **А** будет равно

$$\frac{100}{4} \times 5 = 125 \text{ м.}$$

Ответы на два оставшихся вопроса найдите самостоятельно. Рекомендации по выполнению:

1) проанализируйте значения 4-го, 6-го, 12-го и т.д. отрезков пути;

2) проанализируйте значения общего пути за каждые четыре “шага” маршрута.

Можно также использовать электронную таблицу или разработать компьютерную программу.

Задача “Поход”

Напомним условие: “Группа туристов во время похода частенько мокла под дождем. Пятнадцать дней то утром, то вечером небо заволакивали тучи и полдня лил жесточайший ливень. Но участники похода подметили закономерность: каждый раз, когда дождило с утра, к полудню ветер обязательно разгонял тучи, и наступала чудесная погода. Шестнадцать раз было безоблачное утро. Не могут забыть походники и семнадцать тихих, ясных вечеров. Сколько дней длился поход?”

Благодарим Александра Андрющенко, Валерию Остроухову, Анастасию Пономаренко и Ксению Уткину, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6 (учитель **Рябченко Н.Р.**), и Тараса Леженникова, Краснодарский край, г. Приморско-Ахтарск, школа № 22 (учитель **Корнеева М.В.**), приславших правильный ответ, и приводим решение задачи.

Из наблюдений участников похода следует, что дней, когда дождь был и утром, и вечером, не было, а были дни, когда утром был дождь, а вечером — ясно, и дни, когда дождь шел вечером, а утром его не было. Обозначим число таких дней, соответственно, N_1 и N_2 (согласно условию, $N_1 + N_2 = 15$). Кроме того, были дни, когда дождя не было ни утром, ни вечером. Количество таких дней обозначим N_3 . Тогда общее число дней с ясным утром равно $N_3 + N_2$, и это число по условию равно 16, а общее число дней с “недождливым” вечером — $N_3 + N_1$, и это число по условию равно 17. Итак, можем записать:

$$N_3 + N_2 = 16 \tag{1}$$

$$N_3 + N_1 = 17 \tag{2}$$

Складывая левые и правые части этих равенств, получим: $2N_3 + N_2 + N_1 = 33$, или $2N_3 + 15 = 33$, откуда $N_3 = 9$.

Имея значение N_3 , из равенства (1) получаем $N_2 = 7$, а из равенства (2) — $N_1 = 8$.

Значит, поход длился $N_3 + N_2 + N_1 = 24$ дня.

Числовой ребус со звездочками

Напомним, что требовалось решить числовой ребус:

$$\begin{array}{r} \times \quad * \quad * \quad * \quad * \quad * \quad * \quad * \quad * \quad \text{К} \\ \hline \text{А} \text{ А} \text{ А} \text{ А} \text{ А} \text{ А} \text{ А} \text{ А} \text{ А} \end{array}$$

в котором **А** и **К** — различные числа, а звездочкой (“*”) может быть любая цифра.

Благодарим читателей, приславших правильный ответ (правда — без обоснования):

— Загидуллину Алсу, Рахматуллину Альфиру и Файзуллину Алину, Адельшинская средняя школа, Чистопольский р-н Республики Татарстан, учитель **Фатхутдинова А.А.**;

— Леженникова Тараса, Краснодарский край, г. Приморско-Ахтарск, школа № 22, учитель **Корнеева М.В.**, и приводим начало анализа.

Заметим, что цифра **A** меньше, чем **K**, иначе частное от деления произведения на **K** имело бы не 8, а 9 знаков. Значит, нужно найти такие однозначные числа (цифры **K**), квадрат которых оканчивается на цифру, меньшую этих чисел.

Ответы ко всем задачам, пожалуйста, присылайте в редакцию.

ЯПОНСКИЙ УГОЛОК

Два sudoku

Решите, пожалуйста, две японские головоломки "судоку": 1) простую:

	7			3	5	4	2	
4							7	9
3				4			1	
		3	4	2				
	4	8				5	3	
				5	8	7		
	8		9					1
1	3							7
	2	5	8	7			4	



2) сложную:

			1	3		5		9
1		9						4
	8	3		7				
	7		5	2	8		3	1
5	9				1			7
								3
9								5
	2	5		8	3			6

Ответы (можно не на все головоломки) присылайте в редакцию.

ШКОЛА ПРОГРАММИРОВАНИЯ

Задачи с палиндромами

"Палиндромом" называется слово, предложение или стихотворная строка, одинаково читаемые слева направо и справа налево. Например:

ИДИ

ШАЛАШ

АНИКИНА

АРГЕНТИНА МАНИТ НЕГРА

Рассмотрим ряд задач по программированию, связанных с палиндромами. Как принято в разделе "В мир информатики", используем школьный язык программирования (система КуМир).

Задача 1. Дано слово. Проверить, является ли оно палиндромом.

В соответствии с определением палиндрома задачу можно решить, сравнив заданное слово и слово, образованное из его букв, прочитанных справа налево. Соответствующие две величины назовем *слово* и *слово_наоборот*. Прежде чем представлять соответствующую программу, заметим, что в школьном языке программирования величина строкового (в терминах языка Паскаль) типа рассматривается как массив, элементами которого являются отдельные символы. Это позволяет обратиться к тому или иному символу строки как к элементу массива — по имени строковой величины и его (символа) номеру⁵. Рассмотреть же все символы строки можно с помощью стандартной функции *длин*, возвращающей общее количество символов в строковой величине.

⁵ Так же можно обратиться к отдельному символу строковой величины и в программе на языке Паскаль. В языке Бейсик для этого используется функция MID\$.

Итак, программа решения задачи:

```

алг Проверка
нач лит слово, слово_наоборот, цел i
. вывод нс, "Введите слово"
. ввод слово
. слово_наоборот := "" | Начальное значение
  | величины
. нц для i от длин(слово) до 1 шаг -1
. . | Рассматриваем все символы, начиная
  | с последнего, и добавляем их "старому"
  | значению слово_наоборот
. . слово_наоборот := слово_наоборот +
  слово[i]
. кц
. | Сравниваем слова
. если слово = слово_наоборот
. . то
. . . вывод нс, "Введенное слово - палиндром"
. . иначе
. . . вывод нс, "Введенное слово палиндромом
  не является"
. все
кон
  
```

Можно также решить задачу, не формируя величину *слово_наоборот*, а сравнивая попарно первую и последнюю, вторую и предпоследнюю и т.д. буквы заданного слова. Если встретится хоть одна пара различающихся букв, то заданное слово палиндромом не является. Сравнения букв можно проводить до середины слова. Прежде чем представлять соответствующую программу, установим, с какой буквой сравнивается буква, номер которой (при счете слева направо) — *лев*. Для этого составим такую таблицу (в ней *n* — общее число букв в слове):

Номер буквы	С какой буквой она сравнивается
1	С последней (<i>n</i> -й)
2	С предпоследней, (<i>n</i> - 1)-й
3	С (<i>n</i> - 2)-й
...	...
$n \div 2 - 1$	С ($n - n \div 2 + 2$)-й
$n \div 2$	С ($n - n \div 2 + 1$)-й

— где *div* — знак операции целочисленного деления.

Внимание! Индекс в двух последних строках таблицы нельзя рассчитывать с использованием выражения $n/2$, так как значение номера буквы может быть только целым.

Анализ таблицы показывает, что лев-я буква должна сравниваться с $(n - \text{лев} + 1)$ -й. Здесь возникает также вопрос: а что будет, если n — нечетное число? Ответ — в этом случае значения в таблице не изменятся, а средний элемент (его индекс $n \div 2 + 1$) меняться не будет (убедитесь в этом, рассмотрев конкретные значения n).

Вся программа имеет вид:

```
алг Проверка2
нач лит слово, цел лев, лог палин
. вывод нс, "Введите слово"
. ввод слово
. |Сначала принимаем, что слово - палиндром
. палин := да
. нц для лев от 1 до div(длин(слово), 2)
. . если слово[i] <> слово[длин(слово) - лев + 1]
. . . то
. . . . палин := нет
. . . все
. . кц
. если палин
. . то
. . . вывод нс, "Введенное слово - палиндром"
. . . иначе
. . . вывод нс, "Введенное слово палиндромом
. . . . не является"
. все
кон
```

— где div — функция, возвращающая целую часть частного от деления своего первого аргумента на второй (в других языках программирования для этого используется не функция, а специальная операция).

Можно также прекратить проверки, как только встретятся разные буквы (применим оператор цикла с условием):

```
алг Проверка3
нач лит слово, цел лев, лог палин
. вывод нс, "Введите слово"
. ввод слово
. |Сначала принимаем, что слово - палиндром
. палин := да
. лев := 1
. нц пока палин и лев <= div(длин(слово), 2)
. . если слово[i] <> слово[длин(слово) - лев + 1]
. . . то
. . . . палин := нет
. . . все
. . кц
. если палин
. . то
. . . вывод нс, "Введенное слово - палиндром"
. . . иначе
. . . вывод нс, "Введенное слово палиндромом
. . . . не является"
. все
кон
```

Задача 2. Дано предложение. Проверить, является ли оно палиндромом без учета имеющихся в нем пробелов. Начальных и конечных пробелов и других символов, кроме букв, в предложении нет.

Задачу можно решить в два основных этапа:

- 1) получить слово (скорее всего оно будет бессмысленным), состоящее только из букв заданного предложения (без пробелов);
- 2) проверить полученное слово по методике, описанной применительно к задаче 1.

Следовательно, начало программы решения задачи может быть таким:

```
алг Задача_2
нач лит предложение, слово, цел i
. вывод нс, "Введите предложение"
. ввод предложение
. слово := ""
. нц для i от 1 до длин(предложение)
. . если предложение[i] <> " "
. . . |Если очередной символ - не пробел
. . . . то
. . . . . |добавляем его к "старому"
. . . . . значению слова
. . . . слово := слово + предложение[i]
. . все
. кц
...
```

Определение понятия “палиндром” можно применить и к числам. Например, палиндромами являются числа 7, 121, 42 024.

Является ли число палиндромом, человек может заметить сразу. Но было бы интересно узнать, является ли конкретное десятичное число палиндромом в какой-либо другой системе счисления. Например, число 45 является палиндромом в двоичной и в восьмеричной системах (проверьте!).

Задача 3. Дано натуральное десятичное число. Найти все его палиндромы в системах счисления с основанием от 2 до 9.

Обозначим заданное натуральное десятичное число — n , а основание систем счисления, в которую будет переводиться это число, — *основание*. Идея решения первой части задачи — получить все цифры записи числа n в системе с основанием *основание* и записать каждую из них в массив. Затем исследовать отдельные элементы массива с цифрами числа аналогично тому, как мы исследовали отдельные символы заданного слова при решении задачи 1 (лучше всего методом, описанным в программе Проверка3).

Напомним методику перевода целых чисел из десятичной системы счисления в систему с другим основанием. Необходимо определять остаток от деления заданного числа и всех промежуточных целочисленных частных на *основание* и делать это до тех пор, пока частное не станет равно нулю.

В приведенном ниже фрагменте программы, в котором происходит заполнение массива цифр, кроме величин n и *основание*, использованы также следующие основные переменные величины:

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

— *кол_цифр* — фактическое количество цифр в новой записи числа.

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

```

...
кол_цифр := 0
нц пока n > 0
  |Увеличиваем значение кол_цифр
  кол_цифр := кол_цифр + 1
  |Определяем очередную цифру
  |и записываем ее в массив
  цифры[кол_цифр] := mod(n, основание)
  |Определяем целочисленное частное
  n := div(n, основание)

```

кц

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

Прежде чем представлять всю программу решения задачи, заметим, что в ней используется величина *копия_n* — так сказать, “копия значения заданного числа *n*”, поскольку при выделении цифр обрабатываемое число меняется, а исходное значение *n* понадобится при различных системах счисления.

Итак, программа:

```

алг Задача_3
нач цел n, основание, копия_n, кол_цифр, лев,
i, цел таб цифры[1:20], лог палин
. вывод нс, "Введите натуральное число"
. ввод n
. |Рассматриваем различные основания
. нц для основание от 2 до 9
. . копия_n := n
. . |Заполняем массив цифры
. . кол_цифр := 0
. . нц пока копия_n > 0
. . . кол_цифр := кол_цифр + 1
. . . цифры[кол_цифр] := mod(копия_n, основание)
. . . копия_n := div(копия_n, основание)
. . кц
. . |Проверяем на "палиндромичность"
. . палин := да
. . лев := 1

```

```

. . нц пока палин и лев <= div(длин(слово), 2)
. . . если слово[i] <> слово[длин(слово) -
. . . . лев + 1]
. . . . то
. . . . . палин := нет
. . . . все
. . . кц
. . если палин
. . . то
. . . . |Выводим число в новой системе
. . . . |счисления
. . . . | (цифры в обратном порядке)
. . . . нц для i от кол_цифр до 1 шаг -1
. . . . . вывод цифры[i]
. . . . . кц
. . . . вывод " ", основание
. . все
. кц
кон

```

Примечание. В программе после числа-палиндрома выводится также основание системы счисления, при котором такое число получается.

Задания для самостоятельной работы

1. Разработав любую из приведенных программ решения задачи 1 (на языке программирования, который вы изучаете), установите, является ли палиндромом слово “Аникина”. Объясните полученный результат.

2. Разработайте программу решения задачи “Дано предложение. Проверить, является ли оно палиндромом без учета имеющихся в нем пробелов, цифр, русских букв и других символов, кроме латинских букв”.

3. Определите максимальное количество систем счисления (с основанием от 2 до 9), в которых одно из чисел от 2 до 10 000 является палиндромом.

Результаты присылайте в редакцию. Фамилии всех приславших ответы будут опубликованы. Можно выполнять не все задания.

МЫСЛИ

“Если хочешь иметь досуг, не теряй времени даром”.

Бенджамин Франклин

ВНИМАНИЕ! КОНКУРС!

Конкурс № 111 “Переправы”

Напомним, что задания данного конкурса представляют собой задачи на переправы. Конкурс будет проводиться в несколько туров, а его итоги будут подводиться с учетом всех туров в целом.

Тур 3

3.1. Две семьи (в каждой папа, мама и дочь) хотят переправиться через реку. Есть двухместная лодка. Грести могут только мужчины. Дочери могут быть

на берегу или в лодке только вместе с кем-нибудь из своих родителей. Как им всем переправиться на другой берег?

3.2. То же, но, кроме того, никакую из женщин нельзя оставлять на берегу одну.

Ответы с обоснованием отправьте в редакцию до 1 декабря по адресу: 121165, Москва, ул. Киевская, д. 24, “Первое сентября”, “Информатика” или по электронной почте: vmi@1september.ru. Пожалуйста, четко укажите в ответах свои фамилию и имя, населенный пункт, номер и адрес школы, фамилию, имя и отчество учителя информатики.



Общероссийский проект **Школа цифрового века**

Издательский дом «ПЕРВОЕ СЕНТЯБРЯ» • Издательство «ПРОСВЕЩЕНИЕ»

Каждый педагогический работник образовательной организации, вошедшей в проект «Школа цифрового века», получает доступ ко всем материалам проекта по принципу «все включено» (без дополнительной платы)

МАТЕРИАЛЫ ПРОЕКТА

- **23 предметно-методических журнала** по всем предметам и направлениям школьной жизни плюс журнал для родителей
- **Модульные дистанционные курсы*** из циклов «Навыки профессиональной и личной эффективности педагога» и «Инклюзивный подход в образовании»
- **Дистанционные 36-часовые курсы**** повышения квалификации с выдачей удостоверения установленного образца
- **Методические брошюры** по всем школьным предметам

Стоимость участия образовательной организации в проекте – **6 тысяч рублей за весь учебный год**. Стоимость участия не зависит от количества педагогических работников в образовательной организации

Участие образовательной организации и педагогических работников в проекте удостоверяется соответствующими документами. Для дошкольных организаций предусмотрен свой набор удостоверяющих документов

Срок действия проекта в 2014/15 учебном году: с 1 августа 2014 года по 30 июня 2015 года

Подробности и прием заявок
от образовательных организаций
на сайте

digital.1september.ru

* В течение указанного срока предоставляются без ограничения количества курсов.

** Предоставляется по одному курсу для одного педагогического работника в течение одного учебного года (выбор конкретного курса – на усмотрение педагога).