

ИНФОРМАТИКА А

4

Хоть залейся!
Наш КуМир —
Водолей

14

**Графы —
короли ЕГЭ**
Все, что мы хотели
знать и хотели
спросить о графах

47

Внимание! Конкурс
На разработку заданий
ГИА и ЕГЭ

```
if a[i] > a[i+1]
then swap(a[i], a[i+1])
```





НА ОБЛОЖКЕ

► Пузырек — легендарная личность в мире информатики. Мало кто из коллег не шутил с детьми про алгоритм, названный именем видного ученого Пузырька. Кстати, не так мало ребят сначала не воспринимают эти слова как шутку. А что — есть сортировка фон Неймана, есть сортировка Пузырька ☺. Алгоритм пузырьковой сортировки очень узнаваем по характерному фрагменту кода, в котором заключена основная идея алгоритма, — если два соседних элемента расположены не в требуемом порядке, их нужно поменять местами. Вот, собственно, и вся идея. Зато какая изящная и понятная!

В НОМЕРЕ

- 3** ПАРА СЛОВ
 - А какое такое динамическое программирование?
- 4** БАЗОВЫЙ КУРС
 - Водолей + КуМир + практикум
- 14** ГИА + ЕГЭ
 - Просто графы
- 22** ЕГЭ
 - Задачи С2 из ЕГЭ по информатике
 - Родственники и прочие реляции (новые задачи с базами данных)
 - Приглашаем к участию в конкурсе
- 48** ЗАНИМАТЕЛЬНЫЕ МАТЕРИАЛЫ ДЛЯ ПЫТЛИВЫХ УЧЕНИКОВ И ИХ ТАЛАНТЛИВЫХ УЧИТЕЛЕЙ
 - "В мир информатики" № 174

НА ДИСКЕ



ЭЛЕКТРОННЫЕ МАТЕРИАЛЫ:

- Практикумы в среде КуМир и презентация к статье про исполнитель "Водолей"
- Исходные коды программ к статье о задаче ЕГЭ С2
- Презентации к остальным материалам номера, включая раздел "В мир информатики"

ИНФОРМАТИКА

ПОДПИСНЫЕ ИНДЕКСЫ: по каталогу "Роспечати": 32291 (бумажная версия), 19179 (электронная версия); "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 15509 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
Главный художник:
Иван Лукьянов
Педагогический университет:
Валерия Арсланьян (ректор)

ГАЗЕТА ИЗДАТЕЛЬСКОГО ДОМА

Первое сентября – Е.Бирюкова
ЖУРНАЛЫ ИЗДАТЕЛЬСКОГО ДОМА
Английский язык – А.Громушкина
Библиотека в школе – О.Громова
Биология – Н.Иванова
География – О.Коротова
Дошкольное образование – Д.Тюттерин
Здоровье детей – Н.Сёмина
Информатика – С.Островский
Искусство – М.Сартан
История – А.Савельев
Классное руководство и воспитание школьников – О.Леонтьева
Литература – С.Волков
Математика – Л.Рослова
Начальная школа – М.Соловейчик
Немецкий язык – М.Бузоева
Русский язык – Л.Гончар
Спорт в школе – О.Леонтьева
Управление школой – Е.Рачевский
Физика – Н.Козлова
Французский язык – Г.Чесновицкая
Химия – О.Блохина
Школьный психолог – И.Вачков

УЧРЕДИТЕЛЬ:
ООО "ЧИСТЫЕ ПРУДЫ"

Зарегистрировано ПИ № ФС77-44341 от 22.03.2011 в Министерстве РФ по делам печати
Подписано в печать: по графику 10.02.2012, фактически 10.02.2012
Заказ №
Отпечатано в ОАО "Чеховский полиграфический комбинат" ул. Полиграфистов, д. 1, Московская область, г. Чехов, 142300
Адрес Издателя:
ул. Киевская, д. 24, Москва, 121165
Тел./факс: (499) 249-31-38
Отдел рекламы:
(499) 249-98-70
<http://1september.ru>

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

Документооборот Издательского дома "Первое сентября" защищен антивирусной программой Dr.Web

**28 мая —
основной
день сдачи ЕГЭ
по информатике**



А какое такое динамическое программирование?

**Уважаемые коллеги!
Обратите внимание на объявление о конкурсе на с. 47**

Посредством формы "Письмо в редакцию", которая имеется в электронной версии "Информатики" в Личном кабинете, вы можете отправить письмо с вопросом/мнением/комментарием/предложением непосредственно председателю Федеральной комиссии ЕГЭ по информатике М.А. Ройтбергу.

► В конце февраля, когда до "основного" для сдачи ЕГЭ по информатике оставалось чуть больше трех месяцев, в редакции "Информатики" мы встретились с Михаилом Абрамовичем Ройтбергом — председателем Федеральной предметной комиссии ЕГЭ по информатике. У нас состоялся весьма обстоятельный разговор и о ЕГЭ в целом, и о некоторых конкретных задачах, являющихся предметом дискуссий в профессиональной среде.

К моменту этой встречи (как-то "клавиатура не поворачивается" назвать состоявшийся предметный и деловой разговор интервью) мартовский номер журнала был уже полностью готов к сдаче в печать. Поэтому подробности — в следующем, апрельском номере "Информатики". А здесь (для того чтобы можно было составить впечатление о содержании и стиле беседы) — короткий фрагмент — ответ на один из важных вопросов.

Сергей Островский: Ну а динамическое программирование-то вы зачем туда вставили?

Михаил Ройтберг: А какое такое динамическое программирование?

С.О.: Вы хотите сказать, что задач на динамическое программирование в ЕГЭ нет?

М.Р.: Я могу сказать, что там есть. Параметры задач на так называемое "динамическое программирование" подобраны так, что эти задачи можно решить прямым перебором вариантов, эта тема и раньше была в ЕГЭ. Возьмем, к примеру, задачу В9. Давайте решим ее полным перебором. Засечете время?

С.О.: Засекаю ☺. Но тогда уж полностью и аккуратно все выписывайте.

М.Р.: Хорошо.

...прошло две минуты. Решение председателя Федеральной предметной комиссии совпало с приведенным на с. 20 этого номера "Информатики" (см. "Способ 2").

М.Р.: Ну, вот, вроде все. Эта задача с такими исходными данными быстро и надежно решается полным перебором. Другие задачи "на динамическое программирование" тоже. Но они устроены так, что вызывают вопросы. Об этом можно подумать, об этом можно поговорить со школьниками — как можно экономить время работы алгоритма? как можно уходить от полного перебора? Динамическое программирование — нормальная тема. В ЕГЭ она сейчас отражена очень аккуратно.

С.О.: Какие интернет-ресурсы вы можете порекомендовать для подготовки к ЕГЭ?

М.Р.: Если узко говорить именно об интернет-ресурсах и не использовать слово "порекомендовать" — я все же лицо официальное, — а заменить его, скажем, на "посоветовать", то я бы посоветовал обратить внимание на два сайта. Первый хорошо известен — это сайт Константина Юрьевича Полякова *kpolyakov.narod.ru*. Второй, напротив, известен мало, но потому я и хочу о нем упомянуть — *ege-go.ru*. Если говорить не о подготовке, а о самоконтроле, можно посоветовать еще *ege.yandex.ru*.



Водолей + КуМир + практикум

Введение

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

► В последнее время наблюдается усиление роли раздела “Алгоритмизация и программирование” в школьном курсе информатики. Этому способствует, например, повышение сложности алгоритмических задач в ЕГЭ-2012 [1].

Сейчас практически все учителя согласны с тем, что начинать изучение основ алгоритмизации лучше всего на основе исполнителей. Однако, с одной стороны, исполнитель служит тут только средством, а с другой стороны, содержание такого курса определяется возможностями исполнителя и среды управления им.

Чаще всего в курсах информатики используется исполнитель Робот, имеющий достаточно развитую систему команд, в том числе и команды обратной связи. Другие исполнители, например Водолей (для решения классических задач на переливания), применяются значительно реже, и поэтому нельзя считать, что их возможности до конца раскрыты.

В настоящей статье предлагается небольшой вводный курс алгоритмизации на основе исполнителя Водолей, работающего совместно со средой КуМир. Такой подход позволяет не только управлять исполнителем с пульта, но и программировать его, в том числе составлять программы, работающие для различных начальных условий.

Курс рассчитан на 6–7 учебных часов и успешно используется автором на уроках в 7-х классах общеобразовательной (не специализированной) школы. За это время школьники осваивают

- 1) ручное управление исполнителем с помощью пульта;
- 2) программное управление исполнителем (линейные алгоритмы);
- 3) цикл “N раз”;
- 4) цикл с условием (цикл “пока”);
- 5) понятие переменной;
- 6) ветвления (условные операторы “если–то–иначе”).

В варианте, который использует автор, курс не предусматривает домашних заданий.

Большое преимущество последних версий КуМира — возможность автоматизированной проверки решения задач с помощью концепции практикумов [9]. В последнем разделе статьи будет дан обзор основных возможностей практику-

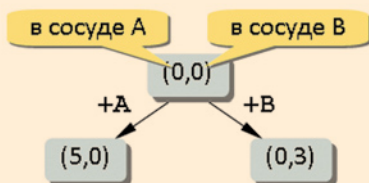
мов, а на диске-приложении размещены сам курс и рабочая тетрадь, содержащая решения всех задач. Некоторые задачи заимствованы из упомянутого выше опубликованного курса Д.П. Кириенко [9].

Немного теории

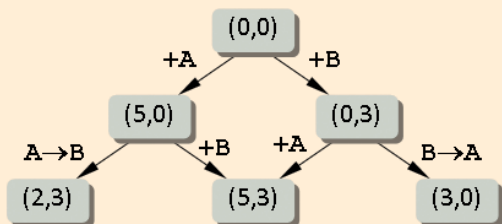
Итак, как же решать задачи с Водолеем? Нас будут интересовать не интуитивные, а систематические методы, позволяющие гарантированно решить задачу (если она разрешима) и, желательно, сделать это за наименьшее число шагов.

Перебор

Простейший способ — перебирать все возможные варианты, начиная с исходного состояния. Рассмотрим задачу с сосудами 5 л и 3 л, в которой нужно набрать 1 л. Будем обозначать состояние сосудов двумя числами, например, состояние (3,2) обозначает, что в сосуде А налито 3 л, а в сосуде В — 2 л. За один шаг можно заполнить один из сосудов (на схеме “+А” обозначает “наполни А”, а “+В” — “наполни В”):

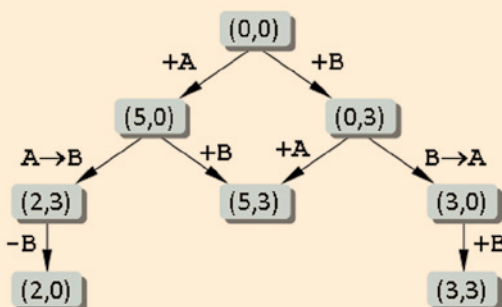


Таким образом, получить 1 л за один шаг не удастся. Проверяем второй шаг. Здесь и далее не будем рассматривать варианты, которые возвращают ситуацию в одно из предыдущих состояний, например, команду “вылей А”:



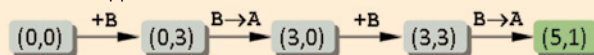
Здесь “А→В” и “В→А” обозначают команды “перелей из А в В” и “перелей из В в А”. За два шага нужный объем воды (1 л) также получить нельзя.

Находим все новые варианты, которые можно получить на третьем ходу:



Состояние (5,3) не позволяет получить никаких новых вариантов. Отметим, что построенная таким образом схема — это ориентированный граф.

За три шага требуемый объем 1 л не получен, а вот на следующем шаге его удастся получить: если в состоянии (3,3) перелить воду из сосуда В в сосуд А, в сосуде В останется 1 л, который не войдет в А. Таким образом, решение задачи — цепочку действий из начального состояния — можно записать в виде схемы:



или в виде последовательности команд исполнителя Водолей в среде КуМир:

1. **наполни В**
2. **перелей из В в А**
3. **наполни В**
4. **перелей из В в А**

Недостаток такого способа состоит в том, что решение получается довольно громоздким.

Чтобы не рисовать схему, можно использовать таблицу, в которой первый столбец — это номер состояния; второй и третий — номер состояния, из которого получено данное, и соответствующая команда; четвертый — количество воды в сосудах, а в пятом будем отмечать состояния, “последствия” которых уже рассмотрены. После первого шага получаем:

№	Предыдущее	Команда	Состояние	Обработано
1	–	–	(0,0)	✓
2	1	+А	(5,0)	
3	1	+В	(0,3)	

Далее смотрим, какие новые состояния можно получить из второго:

№	Предыдущее	Команда	Состояние	Обработано
1	–	–	(0,0)	✓
2	1	+А	(5,0)	✓
3	1	+В	(0,3)	
4	2	А→В	(2,3)	
5	2	+В	(5,3)	

и так далее. Приведем полную таблицу:

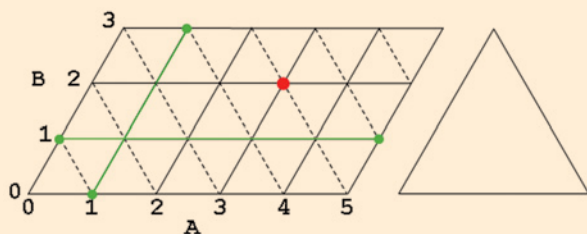
№	Предыдущее	Команда	Состояние	Обработано
1	–	–	(0,0)	✓
2	1	+А	(5,0)	✓
3	1	+В	(0,3)	✓
4	2	А→В	(2,3)	✓
5	2	+В	(5,3)	✓
6	3	В→А	(3,0)	✓
7	4	-В	(2,0)	✓
8	6	+В	(3,3)	✓
9	8	В→А	(5,1)	

Обратите внимание, что все состояния, которые уже встречались в таблице, мы повторно не записывали.

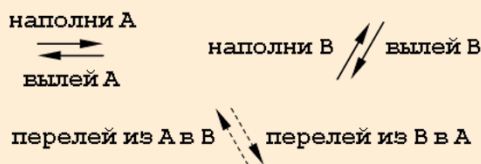
Для того чтобы найти нужную последовательность команд, используем второй столбец, двигаясь “с конца”. Нужный объем получился в состоянии 9, которое получено из состояния 8 командой “перелей из В в А”. В свою очередь, состояние 8 получено из 6 с помощью команды “наполни В” и так далее. В результате получим то же решение, что и раньше.

Бильярд

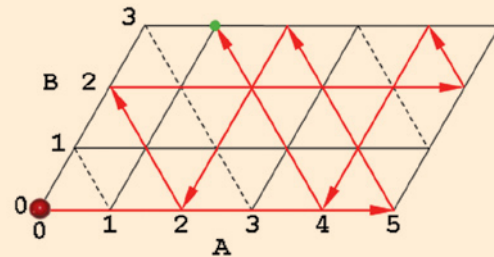
Более интересный способ решения задач на переливания, который предложил М.К.К. Твиди [10], использует закон отражения бильярдного шара от бортов. Только стол здесь не прямоугольный, как обычно, а имеет форму параллелограмма с углом 60°. Одна сторона параллелограмма равна (в некоторых единицах) размеру первого сосуда, а вторая — размеру второго. В рассматриваемой задаче он выглядит так (рядом со “столом” показан равносторонний треугольник, все углы которого равны 60°):



Каждая горизонтальная линия соответствует некоторому объему воды в сосуде В, а сплошные наклонные отрезки — различным объемам воды в сосуде А. Например, красной точкой отмечено состояние (3,2), когда в сосуде А налито 3 л воды, а в сосуде В — 2 л. Задача сводится к тому, чтобы перейти из начального состояния (0,0) на одну из зеленых линий, то есть в любое состояние, в котором один из сосудов содержит 1 л воды. Для этого можно использовать следующие команды (стрелки показывают направление движения на “бильярдном столе”):



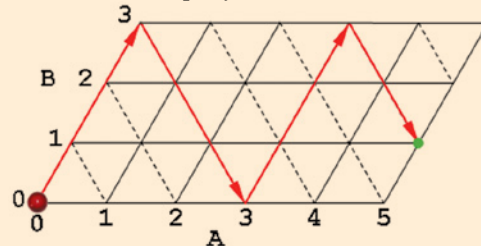
Решение определяется траекторией движения бильярдного шара, запущенного из начального состояния вдоль одной из линий. Например, для начального состояния (0,0) возможно движение вдоль горизонтальной линии (начать с наполнения сосуда А) или вдоль сплошной наклонной линии (начать с наполнения сосуда В). Построим траекторию движения шара с учетом отскока от бортов для первого случая:



По этой траектории, состоящей из 8 отрезков, можно восстановить цепочку команд Водолея:

- 1. наполни А
- 2. перелей из А в В
- 3. вылей В
- 4. перелей из А в В
- 5. наполни А
- 6. перелей из А в В
- 7. вылей В
- 8. перелей из А в В

Итак, полученное нами решение содержит 8 шагов. Впору задуматься, потому что ранее (методом перебора) мы нашли более короткое решение (из четырех шагов). Оказывается, его можно получить, “запустив” шарик вдоль наклонной стороны стола (наполняя сначала сосуд В). Траектория для этого случая показана на рисунке:



Легко проверить, что это то же самое решение, которое получено методом перебора. К сожалению, автору неизвестно, как заранее определить, какой именно из двух вариантов “запуска” бильярдного шара даст более короткое решение.

Обратите внимание, что в каждом из двух случаев переливания из сосуда в сосуд идут только в одном направлении: или все время из А в В, или все время из В в А. Таким образом, “бильярдный” подход приводит к двум универсальным алгоритмам (они записаны на школьном алгоритмическом языке системы КуМир):

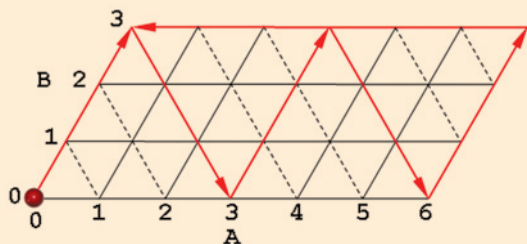
```

наполни А
нц пока не достигли цели
  перелей из А в В
  если сосуд В полон
    то вылей В
  все
  если сосуд А пуст
    то наполни А
  все
кц
    
```

```

наполни В
нц пока не достигли цели
  перелей из В в А
  если сосуд А полон
    то вылей А
  все
  если сосуд В пуст
    то наполни В
  все
кц
    
```

Математики знают, что не каждая задача для Водолея имеет решение. Например, пусть сосуды имеют размеры 6 л и 3 л, и в начальный момент они пусты. При этом траектория движения шара зацикливается:



Таким образом, мы можем получить только 0, 3 или 6 л. Заметим, что если в сосудах есть вода, ситуация может измениться. Например, если в любом сосуде налито 2 л воды, мы сможем также отмерить 2 или 5 л, однако траектория все равно зациклится.

Защикливание траектории связано с тем, что размеры сосудов имеют наибольший общий делитель (НОД), отличный от 1: в данном случае это 3. Если размеры сосудов — взаимно простые числа (их НОД равен 1), то можно отмерить любой объем воды от 1 л до размера наибольшего из сосудов.

Диофантовы уравнения

Перейдем еще на один уровень выше, к “чистой” математике. Дело в том, что задачи на переливания сводятся к решению так называемых *линейных диофантовых уравнений* — уравнений вида

$$c_1x_1 + c_2x_2 + \dots + c_nx_n = d,$$

в которых все коэффициенты $c_i (i = 1, \dots, n)$, свободный член d и все неизвестные величины $x_i (i = 1, \dots, n)$ — целые числа. Например, для задачи с сосудами 5 л и 3 л, в которой нужно получить 1 л, соответствующее уравнение выглядит так:

$$5a + 3b = 1, \quad (*)$$

где a и b — целые числа, обозначающие, сколько раз нужно наполнить или вылить каждый из сосудов. Предположим, что мы хотим переливать воду из сосуда А в сосуд В. Это значит, что величина a должна быть положительной (мы *наливаем* воду в сосуд А из крана), а b — отрицательной (выливаем из сосуда В в раковину).

Попробуем решить это уравнение методом подбора. Будем перебирать различные значения a , начиная с 1, пытаюсь найти целое b , удовлетворяющее уравнению (*). Выразим b из (*):

$$b = \frac{1 - 5a}{3}.$$

При $a = 1$ целое значение b не получается, а при $a = 2$ находим $b = -3$. Это значит, что нужно два раза наполнить сосуд А и три раза опорожнить сосуд В. После этого в сосуде А останется 1 л воды. Как видим, мы получили приведенное выше решение из 8 шагов.

Теперь найдем второе решение, при котором наполняем сосуд В и переливаем из сосуда В в сосуд А. Выразим a из (*):

$$a = \frac{1 - 3b}{5}$$

и будем подбирать целые положительные значения b так, чтобы a тоже оказалось целым. При $b = 1$ целое значение a не получается, а при $b = 2$ находим $a = -1$. Это означает, что нужно два раза наполнить сосуд В и один раз вылить воду из сосуда А — мы получили короткое решение из четырех шагов. Мы увидели, что решение уравнения (*) неединственно. Известно, что множество всех решений уравнения (*) описывается формулами:

$$a = a_0 + 3t, \quad b = b_0 - 5t,$$

где пара (a_0, b_0) — любое решение уравнения, а параметр t — произвольное целое число.

В более сложных случаях диофантовы уравнения решаются с помощью расширенного алгоритма Евклида или конечных цепных дробей [11–12].

Теперь перейдем к содержанию уроков, составляющих вводный курс алгоритмизации на основе исполнителя Водoley.

Урок 0. Ручное управление

Главное на этом вводном уроке — знакомство с исполнителем Водoley, его средой и пультом управления. Для начала можно ограничиться двумя сосудами (которые в среде КуМир называются латинскими буквами А и В) и командами

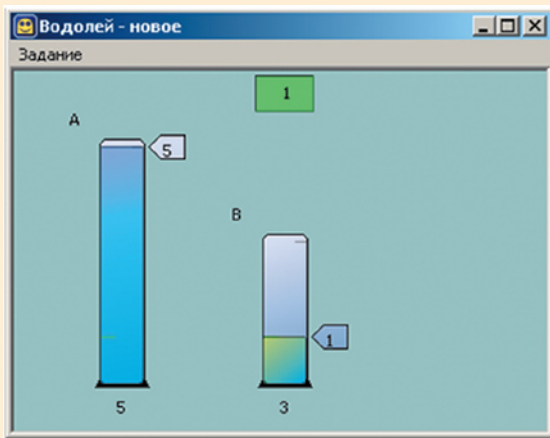
- | | |
|-------------------------|-------------------------|
| наполни А | вылей А |
| наполни В | вылей В |
| перелей из А в В | перелей из В в А |

Для практической работы удобно запустить исполнителя Водoley из среды КуМир, выбрав пункты меню *Миры — Водoley* и *Миры — Водoley Пульт* в любой последовательности. Для решения предлагаются 4 задачи, которые содержатся на диске:

- 0-1: сосуды 5 л и 3 л, набрать 1 л.
- 0-2: сосуды 6 л и 10 л, в первом налито 3 л; набрать 1 л.
- 0-3: сосуды 8 л и 5 л, набрать 1 л.
- 0-4: сосуды 8 л и 5 л, набрать 4 л.

Для загрузки задачи в меню окна Водoley нужно выбрать пункт *Задание — Загрузить*. После щелчка на кнопках пульта выполняется соответствующая команда, которая сразу записывается в окно в верхней части пульта. Кнопка позволяет восстановить исходное состояние сосудов и удалить программу.





Первые две задачи простые, они решаются за 4 шага, последние две — посложнее. Как правило, в течение практической части урока (около 20 минут) большинство школьников решает 3 задачи.

Урок 1. Управление из программы

На этом уроке школьники должны понять, что

- 1) существуют систематические способы решения задач с Водолеем, которые позволяют гарантированно найти решение, если оно существует;
- 2) некоторые задачи не имеют решения;
- 3) Водолеем можно управлять из программы.

Как следует из раздела “Немного теории”, существуют систематические методы, которые позволяют гарантированно решить задачу на переливания, если она разрешима. Если размеры сосудов — взаимно простые числа, с их помощью можно отмерить любой объем от 1 л до размера наибольшего из сосудов. Если НОД этих размеров равен $d \neq 1$ и сосуды пусты в начальный момент, то можно отмерить только те объемы воды, которые кратны d .

Теперь можно перейти к программированию исполнителя. Зачем это нужно? Дело в том, что в реальных ситуациях часто нет возможности использовать пульт для управления роботом. Например, летательный аппарат некоторое время работает в автономном режиме без связи с центром управления. Или нужно выдавать команды управления несколько раз в секунду без права на ошибку — человек просто не справится с таким темпом. В таких случаях заранее *спланировать* действия исполнителя, то есть написать программу, содержащую команды управления для каждой возможной ситуации.

Реализация Водолея, с которой мы работаем, напрямую связана с оболочкой системы КуМир. Это дает следующие возможности:

- 1) управлять исполнителем из программы на алгоритмическом языке;
- 2) передавать последовательность команд, записанную с помощью пульта, в программу (используя кнопку **К** на пульте);
- 3) организовать автоматическую проверку правильности решения задач.

Программа для решения 0-1.vod (сосуды 5 л и 3 л, набрать 1 л) выглядит так:

```
использовать Водолей
алг Отмерить 1 литр
нач
    наполни В
    перелей из В в А
    наполни В
    перелей из В в А
кон
```

При нажатии клавиши **F9** она выполняется полностью для задачи, загруженной в окно Водолея. Клавиша **F8** позволяет выполнять программу в пошаговом режиме, контролируя состояние Водолея после каждой выполненной команды.

Кроме четырех задач, которые уже решались в ручном режиме на предыдущем уроке, в практической части урока предлагаются еще две задачи с тремя сосудами [0]:

- 1-5: сосуды 6 л, 10 л и 15 л, набрать 1 л.
- 1-6: сосуды 6 л, 10 л и 15 л, набрать 7 л.

Большинство учащихся скорее всего решат 4 задачи из 6.

Урок 2. Цикл “N раз”

На этом уроке нужно научиться составлять программы с циклами “N раз”, в которых число шагов заранее известно или может быть рассчитано до выполнения цикла. При этом удастся

- 1) значительно сократить текст программы, заменив циклом повторяющиеся последовательно команды;
- 2) вычислять необходимое количество шагов и таким образом составлять программы, решающие задачу для различных начальных состояний исполнителя.

Цикл — это многократное выполнение одинаковых действий. В программе, которая рассматривалась на предыдущем уроке, последовательность команд

```
наполни В
перелей из В в А
выполни 2 раза
наполни В
перелей из В в А
кц
```

выполнялась 2 раза, поэтому ее можно заменить циклом

В такой записи “**нц**” означает начало цикла, а “**кц**” — конец цикла.

Рассмотрим еще одну задачу: с помощью сосудов 7 л и 2 л набрать 1 л воды. Решается она очень просто:

	А	В
наполни А	7	0
перелей из А в В	5	2
вылей В	5	0
перелей из А в В	3	2
вылей В	3	0
перелей из А в В	1	2

Справа от каждой команды показано, сколько литров воды окажется в каждом сосуде после ее выполнения. Если добавить в конец программы еще одну команду “**вылей В**”, задача по-прежнему будет решена правильно (в сосуде А набран 1 л воды). Однако при этом замечаем, что группа команд

```
перелей из А в В
вылей В
```

выполняется 3 раза, поэтому можно записать программу с помощью цикла:

```
наполни А
нц 3 раз
  перелей из А в В
  вылей В
кц
```

Что изменится, если размер сосуда А — не 7, а другое нечетное число? Почти ничего, только количество шагов цикла:

размер А	1	3	5	7	9	11	13	15
число шагов	0	1	2	3	4	5	6	7

Легко заметить, что для вычисления количества шагов нужно разделить размер сосуда А на 2 (то есть на размер сосуда В), причем остаток от деления нужно отбросить. Эту операцию в школьном алгоритмическом языке выполняет функция **div**: выражение **div(a, b)** означает результат деления **a** на **b** нацело, с отбрасыванием остатка.

В КуМире можно из программы определять размеры сосудов с помощью команд “**размер А**” и “**размер В**”. Тогда универсальная программа, которая позволяет отмерить 1 л воды при любом нечетном размере сосуда А и размере сосуда В, равном 2 л, запишется так:

```
наполни А
нц div(размер А, 2) раз
  перелей из А в В
  вылей В
кц
```

Для практической работы предлагаются следующие задачи, которые нужно решить с помощью цикла “N раз”:

- 2-1: сосуды 7 л и 2 л, набрать 1 л.
- 2-2: сосуды 5 л и 2 л, набрать 1 л.
- 2-3: сосуды 11 л и 2 л, набрать 1 л.
- 2-4: размер А — произвольное нечетное число, размер В — 2 л, набрать 1 л.
- 2-5: размер А — 3 л, размер В = $3k + 1$ (k — натуральное число или 0), набрать 1 л.
- 2-6: размер А = (размер В) · $k + 1$ (k — натуральное число или 0), набрать 1 л.
- 2-7: размер А = $3k + 1$ (k — натуральное число или 0), размер В — 3 л, набрать 1 л.
- 2-8: размер А = (размер В) · $k - 1$ (k — натуральное число), набрать 1 л.

Первые три задачи — разминочные, большинство учеников решит 6 задач.

Урок 3. Цикл “пока”

На этом уроке представляется еще один вид цикла — цикл “пока” или цикл с условием, который выполняется до тех пор, пока условие в заголовке цикла не станет ложно.

Удобно сравнить два вида циклов на примере уже разбиравшейся ранее задачи: набрать 1 л с помощью сосудов размером 7 л и 2 л. Новое решение выглядит так:

```
наполни А
нц пока в сосуде А <> 1
  перелей из А в В
  вылей В
кц
```

Два знака “<” и “>”, записанные рядом, означают “не равно”, а запись “**в сосуде А**” — это команда обратной связи исполнителя Водолей, с помощью которой мы можем узнать объем воды в сосуде в данный момент. Таким образом, цикл будет выполняться, пока в сосуде А не останется 1 л воды. Важно, что эта программа работает для любой задачи, в которой размер А — нечетное число, размер В — 2 л, и требуется набрать 1 л воды. Вычислять количество шагов цикла не нужно.

Однако неверное использование цикла с условием может привести к неприятным ошибкам. Например, если использовать приведенную выше программу для сосудов размером 12 л и 2 л, программа никогда не остановится, потому что в сосуде А никогда не получится 1 л воды. Это называется “зацикливание”. Чтобы зацикливания в любом случае не было, можно изменить условие так:

```
нц пока в сосуде А > 1
  перелей из А в В
  вылей В
кц
```

Такой цикл останавливается в любом случае (хотя бы тогда, когда из сосуда А вылита вся вода).

Задания для практической работы:

- 3-1: размер А — нечетное число, размер В — 2 л, набрать 1 л.
- 3-2: размер А — 3 л, размер В = $3k + 1$ (k — натуральное число или 0), набрать 1 л.
- 3-3: размер А — 3 л, размер В = $3k + 2$ (k — натуральное число или 0), набрать 1 л.
- 3-4: размер А — 4 л, размер В = $4k$ (k — натуральное число, $k > 2$); в сосуд А налито 2 л воды, в сосуд В — 1 л; набрать 10 л.
- 3-5: размер А = $4k$ (k — натуральное число, $k > 2$), размер В — 4 л; в сосуд А налито 2 л воды, в сосуд В — 1 л; набрать 9 л.
- 3-6: размер А = $4k$ (k — натуральное число, $k > 2$), размер В — 4 л; в сосуд А налито 2 л воды, в сосуд В — 1 л; набрать 11 л.

Урок 4. Переменные

На предыдущем уроке мы использовали команды “**в сосуде А**” и “**в сосуде В**”, с помощью которых определяли объем воды в сосудах после очередного шага цикла. В реальных системах для определения уровня воды в сосудах используются измерительные приборы (датчики). Многие датчики — это довольно дорогие элементы, и в простых случаях можно обойтись без них. Действительно, если мы знаем размеры обоих сосудов и знаем, что сосуд А был наполнен, а затем воду из него перелили в сосуд В, можно вычислить, сколько воды осталось в сосуде А.

На этом уроке мы не будем использовать команды “**в сосуде А**” и “**в сосуде В**”, считая, что датчиков нет (или они неисправны).

Для того чтобы запоминать данные во время работы программы, используется оперативная память. Память разбита на ячейки, в каждой из которых записано некоторое значение (число). Для того чтобы обращаться к ячейкам памяти, в языках программирования используют понятие переменной.

Переменная — это величина, которая имеет имя, тип и значение. Имя переменной в школьном алгоритмическом языке может содержать русские и латинские буквы, цифры и даже пробелы, но начинается обязательно с буквы. Тип переменной определяет размер отведенной ей ячейки памяти, диапазон возможных значений и операции, которые с ней можно выполнять. Значение переменной — это данные, записанные в соответствующей ячейке памяти; значение можно изменять в ходе выполнения программы.

Для того чтобы использовать переменную, ее нужно *объявить*: указать тип и имя переменной. Например, целая переменная *N* объявляется так:

цел N

Значение такой переменной не определено, и использовать ее нельзя (будет ошибка при выполнении программы). Оператор присваивания “:=” изменяет значение переменной, например,

N := 25

Очень важный прием — увеличение (или уменьшение) значения переменной на какую-то величину. Например, оператор

N := N + 1

означает “прочитать текущее значение *N*, прибавить к нему 1 и записать результат обратно в переменную *N*”.

Рассмотрим уже известную задачу — набрать 1 л с помощью сосудов размером 7 л и 2 л. Будем считать, что датчиков нет, и нужно запоминать, сколько воды остается в сосуде А после очередного переливания. Сначала наполним сосуд А, объявим целую переменную *N*, запишем в нее размер сосуда А

наполни А

цел N

N := размер А

Предполагая, что в переменной *N* будет храниться объем воды в сосуде А, можно записать цикл так:

```
нц пока N > 1
    перелей из А в В
    вылей В
    N := N - 2
кц
```

Здесь условие “**в сосуде А > 1**” мы заменили на “**N > 1**”, кроме того, после каждого переливания значение переменной *N* уменьшается на 2, то есть на размер В.

Задания для практической работы:

4-1: размер А — нечетное число, размер В — 2 л, набрать 1 л.

4-2: размер А — 2 л, размер В — нечетное число, набрать 1 л.

4-3: размер А — 3 л, размер В = $3k + 1$ (*k* — натуральное число или 0), набрать 1 л.

4-4: размер А = $3k + 2$ (*k* — натуральное число или 0), размер В — 3 л, набрать 1 л.

4-5: размер А = (размер В) · *k* + 1 (*k* — натуральное число или 0), набрать 1 л.

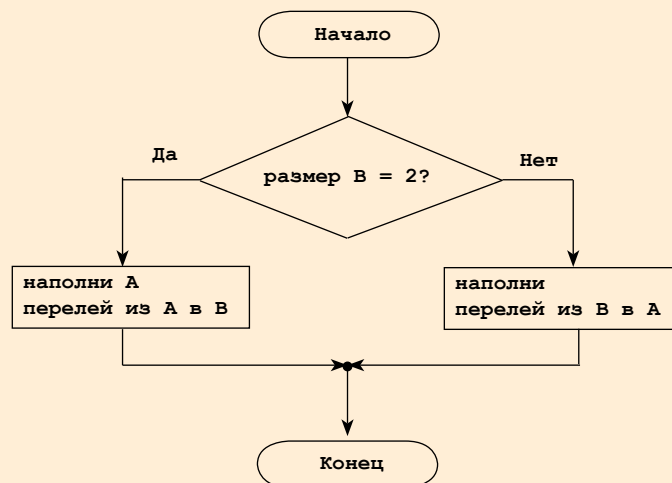
4-6: размер А = (размер В) · *k* - 1 (*k* — натуральное число), набрать 1 л.

Урок 5. Ветвления

При управлении реальными роботами невозможно заранее точно знать, какую последовательность команд нужно будет выполнить исполнителю — это зависит от ситуации, в которой он оказался. Поэтому в программе нужно предусмотреть различные варианты действий. В результате получаются *разветвляющиеся алгоритмы*, в которых выполняемая последовательность команд зависит от того, истинно или ложно некоторое условие. Если условие истинно, выполняется одна группа команд, если ложно — другая.

Рассмотрим простую задачу — нужно набрать 1 л воды с помощью сосудов, размеры которых 3 л и 2 л. При этом неизвестно, какой именно из сосудов (**А** или **В**) имеет размер 3 л, а какой — 2 л. В то же время, мы можем уже во время работы программы определять размеры сосудов, используя команды “**размер А**” и “**размер В**”. Поэтому алгоритм решения этой задачи на русском языке (для человека) может выглядеть так: “Если размер сосуда В — 2 л, то наполни сосуд А и перелей из А в В. Иначе наполни В и перелей из В в А”. Почти так же записывается алгоритм и на школьном алгоритмическом языке:

```
если размер В = 2
    то
        наполни А
        перелей из А в В
    иначе
        наполни В
        перелей из В в А
все
```



Для понимания работы разветвляющихся алгоритмов удобно использовать блок-схемы. В данном случае блок-схема представлена на рисунке вверху.

Задания для практической работы:

5-1: размер одного из сосудов (неизвестно какого) — 1 л, размер другого неизвестен, набрать 1 л.

5-2: размер одного из сосудов (неизвестно какого) — 2 л, размер другого — 3 л, набрать 1 л.

5-3: размер одного из сосудов (неизвестно какого) — 2 л, размер другого — 11 л, набрать 1 л (используйте цикл “N раз”).

5-4: размер одного из сосудов (неизвестно какого) — 4 л, размер другого — 15 л, набрать 1 л (используйте цикл “N раз”).

5-5: размер одного из сосудов (неизвестно какого) — 2 л, размер другого — нечетное число, набрать 1 л (используйте цикл “пока”).

5-6: размер одного из сосудов (неизвестно какого) — 2 л, размер другого — нечетное число, набрать 1 л (используйте цикл “N раз”).

Практикумы в КуМире

В последних версиях КуМира [13] появилась возможность создавать практикумы (курсы). Первые практикумы — курсы по исполнителям Водoley и Робот — были разработаны Д.П. Кириенко [9].

Практикум — это набор задач, которые можно группировать по темам. Каждая задача включает условие, шаблон программы и набор обстановок для исполнителя, на которых проводится проверка решения.

Зачем нужен шаблон программы? Прежде всего, на начальном этапе ученику бывает достаточно сложно написать всю программу целиком за короткое время. Для ускорения процесса можно включить в шаблон строки, которые не относятся к задачам данного урока, и защитить их от изменений, оставив ученику ограниченное “поле деятельности” — строки, в которых он должен что-то изменить или добавить. Например, заготовка программы для решения “разминочной” задачи 2-1 может выглядеть так:

использовать Водoley
алг Отмерить 1 литр
нач

наполни А
нц раз
перелей из А в В
вылей В
кц
кон

Строки, выделенные серо-зеленым фоном, заблокированы, и ученик не может изменить их. В его распоряжении — только одна строка, в которую нужно вписать количество шагов цикла. При решении следующих задач шаблон постепенно “освобождается” и ученику нужно уже полностью написать программу:

использовать Водoley
алг Отмерить 1 литр
нач

кон

Защитить строки шаблона от изменения можно в режиме учителя. Для этого нужно при запуске КуМира указать в командной строке параметр “-t”.

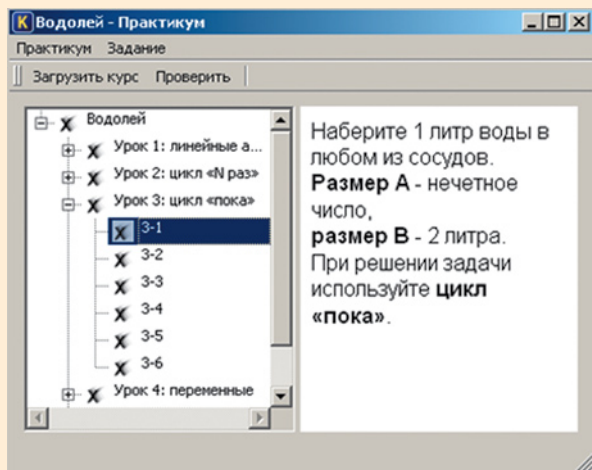
Кроме того, в режиме учителя в шаблон можно добавить скрытые строки, которые ученик не видит. Они нужны для программы тестирования, с помощью которой выполняется автоматическая проверка правильности выполнения задания. Для задач, которые выполняет Водoley, программа тестирования выглядит так [9]:

алг цел @тестирование
нач
Отмерить 1 литр
если @решено то
знач := 10
вывод "Задание выполнено успешно", нс
иначе
вывод "Задание выполнено неверно", нс
знач := 0
все
кон

Тестирующий алгоритм должен называться @**тестирование** и возвращать, например, число 10, если задача решена успешно, и 0, если решение неверно. Приведенный выше алгоритм сначала вызывает алгоритм **Отмерить 1 литр**, написанный учеником. После этого вызывается алгоритм @**решено**, который возвращает логическое значение, определяющее правильность выполнения задания. Программа тестирования запускается при автоматической проверке или при нажатии клавиш **Ctrl** + **T**.

С каждой задачей можно связать несколько обстановок исполнителя. Например, в задаче 2–4 алгоритм должен работать при любом нечетном размере сосуда А, поэтому правильность решения нужно проверить для разных вариантов, например, 1 и 2 л, 7 и 2 л, 15 и 2 л и т.д.

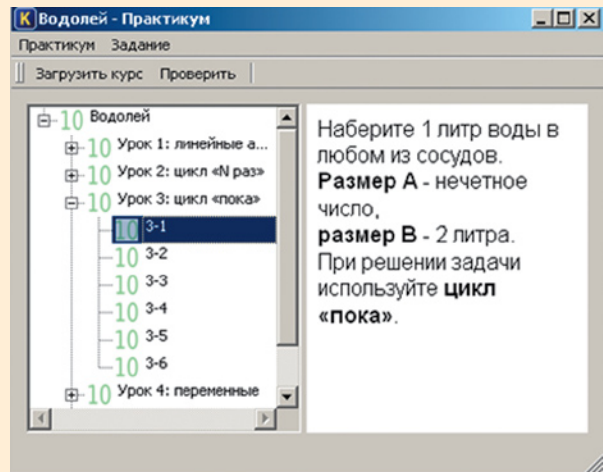
Начиная работу с практикумом, ученик открывает окно практикума (меню *Инструменты* — *Практикум* в КуМир) и с помощью кнопки *Загрузить курс* загружает основной XML-файл курса, в данном случае это файл **Водолей.kurs.xml**.



В левой части этого окна нужно выбрать урок и задачу, тогда справа появляется условие задачи, в окно исполнителя загружается одна из заданных обстановок, а в основное окно КуМира — шаблон для этой задачи.

Когда решение построено, можно проверить его на загруженной обстановке (используя клавиши **F9** — запуск, и **F8** — пошаговый режим). Затем нужно щелкнуть по кнопке *Проверить* в окне практикума. В первый раз будет предложено сохранить рабочую тетрадь — файл с двойным расширением ***.work.xml**, в которой будут сохраняться все программы ученика и отмечаться правильно решенные задачи. В следующий раз нужно будет загружать уже не файл курса, а свою рабочую тетрадь.

При проверке тестирующий алгоритм (который записан в шаблоне задачи) вызывается для каждой из обстановок исполнителя, связанных с этой задачей. Если на всех обстановках программа отработала верно, слева от номера задачи вместо крестика появляется число 10:



Выводы

1. Исполнитель Водолей может успешно использоваться для начального курса алгоритмизации, в ходе которого изучаются циклы, условные операторы и переменные.
2. Технология практикумов среды КуМир оказывается весьма полезной для автоматизированной проверки правильности решения задач.

Литература

1. Демонстрационный вариант КИМ для проведения ЕГЭ 2012 года по информатике (<http://www.fipi.ru/view/sections/222/docs/578.html>).
2. *Пейперт С.* Переворот в сознании: дети, компьютеры и плодотворные идеи. М.: Педагогика, 1989.
3. *Звенигородский Г.А.* Первые уроки программирования. Библиотечка “Квант”, вып. 41. М.: Наука, 1985.
4. *Гольцман А.М., Дуванов А.А., Зайдельман Я.Н., Первин Ю.А.* Исполнители // “Информатика и образование” № 4, 1990.
5. *Кушницренко А.Г., Леонов А.Г.* Методика преподавания основ алгоритмизации на базе системы КуМир (<http://edu.1september.ru/courses/07/010/>).
6. *Кушницренко А.Г., Лебедев Г.В., Сворень Р.А.* Основы информатики и вычислительной техники: Пробный учебник для средних учебных заведений. М.: Просвещение, 1990.
7. *Звонкин А.К., Ландо С.К., Семенов А.Л.* Информатика. Алгоритмика. Учебник для 6-х классов общеобразовательных учреждений. М.: Просвещение, 2006.
8. *Поляков К.Ю.* Программная среда “Исполнители” (<http://kpolyakov.narod.ru/school/robots/robots.htm>).
9. *Кириенко Д.П.* Курс алгоритмизации с использованием исполнителей системы КуМир и автоматического тестирования (<http://server.179.ru/wiki/?page=DenisKirienko/Kumir>).
10. *Гарднер М.* Математические досуги. М.: Мир, 1972.
11. *Гельфонд А.О.* Решение уравнений в целых числах. М.: Наука, 1978.
12. *Серпинский В.* О решении уравнений в целых числах. М.: Физматлит, 1961.
13. <http://lpm.org.ru/kumir2/>.

РАСПИСАНИЕ ДНЕЙ ПЕДАГОГИЧЕСКОГО МАРАФОНА

20 марта	Открытие День классного руководителя	29 марта	День учителя биологии
21 марта	День школьного психолога	30 марта	День учителя информатики
22 марта	День здоровья детей, коррекционной педагогики, логопеда, инклюзивного образования и лечебной физической культуры День учителя технологии (ЦО № 293)	31 марта	День учителя физики
23 марта	День учителя начальной школы (день первый)	1 апреля	День учителя математики
24 марта	День учителя начальной школы (день второй)	3 апреля	День учителя истории и обществознания
25 марта	День дошкольного образования	4 апреля	День учителя МХК, музыки и ИЗО
27 марта	День учителя географии	5 апреля	День школьного и детского библиотекаря
28 марта	День учителя химии День учителя ОБЖ (Кадетская школа № 1784)	6 апреля	День учителя литературы
		7 апреля	День учителя русского языка
		8 апреля	День учителя английского языка
		10 апреля	День учителя французского языка
		11 апреля	День учителя немецкого языка
		12 апреля	День учителя физической культуры
		13 апреля	День школьной администрации Заккрытие

marathon.1september.ru

! Обязательная предварительная регистрация на все дни Марафона откроется 20 февраля 2012 года на сайте marathon.1september.ru.

! Каждый участник Марафона, посетивший три мероприятия одного дня, получает официальный именной сертификат (6 часов)

В дни Марафона ведущие издательства страны представляют книги для учителей

Начало работы – 9.00. Завершение работы – 15.00

УЧАСТИЕ БЕСПЛАТНОЕ. ВХОД ПО БИЛЕТАМ

РЕГИСТРИРУЙТЕСЬ, РАСПЕЧАТЫВАЙТЕ СВОЙ БИЛЕТ И ПРИХОДИТЕ!

Место проведения Марафона: лицей № 1535, ул. Усачева, дом 50 (в 3 минутах ходьбы от станции метро «Спортивная»)

Место проведения Дня учителя технологии: ЦО № 293, ул. Ярославская, д. 27 (ст. метро «ВДНХ»)

Место проведения Дня учителя ОБЖ: Кадетская школа № 1784, ул. Скаковая, д. 20 (ст. метро «Динамо», «Белорусская»)

По всем вопросам обращайтесь по телефону **8-499-249-3138** или по электронной почте: marathon@1september.ru

ПОДРОБНАЯ ПРОГРАММА КАЖДОГО ДНЯ МАРАФОНА:

- на сайте marathon.1september.ru
- в электронных приложениях
- на вложенном диске



Просто графы

1. Введение

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

В последние годы в школьный курс информатики стремительно вошла тема “Графы”, что связано прежде всего с включением соответствующих задач в варианты ЕГЭ и ГИА. Тема эта глубокая и интересная, имеет много серьезных практических приложений (транспортные перевозки, проектирование сетей коммуникаций, маршрутизация в Интернете, разводка печатных плат).

С другой стороны, изучение графов с помощью классических вузовских учебников [1–3] в течение нескольких уроков — дело достаточно бессмысленное. Потому что за короткое время удастся только познакомиться с довольно сложной (и не до конца устоявшейся) терминологией и очертить круг задач, которые решаются с помощью графов. Чтобы изучить методы решения этих задач, многие из которых NP-полные (которые пока решаются только перебором всех вариантов), нужно потратить не один десяток часов и написать не одну программу, которая значительно длиннее “школьных” 30–40 строчек.

Таким образом, необходимо сделать выбор — изучать графы за несколько уроков на уровне общих понятий или по-

святивать глубокому изучению графов достаточно большое количество времени (в объеме семестрового курса вуза).

Кроме того, имеет смысл разделять два варианта значения слова “граф”:

1) удобная форма описания структур типа дорожной сети или сети передачи данных;

2) математический объект $G := (V, E)$, где V — это непустое множество вершин, а E — множество ребер (пар вершин).

Очевидно, что в школьном курсе информатики (по крайней мере на базовом уровне) “математическое” понятие графа не требуется (так же, как, например, в базовом курсе не изучается строгое определение термина “алгоритм”, использующее понятие универсального исполнителя типа машины Тьюринга).

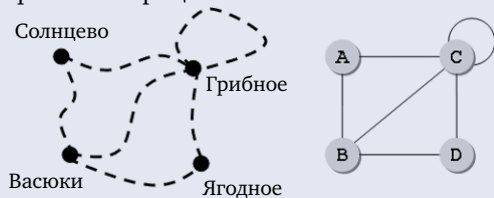
Задача настоящей статьи — представить минимальный материал, необходимый для решения задач ЕГЭ и ГИА на графы (задачи А2, В9 в демоварианте ЕГЭ-2012 [4], задачи 3 и 11 в демоварианте ГИА-2012 [5]), для освоения которого достаточно 2–3 уроков. Приводятся несколько способов решения этих задач, так что каждый может выбрать то, что ему по душе. Кроме того, рассмотрение объекта с разных сторон способствует пониманию его сущности.

Начнем с основных понятий, связанных с графами.

2. Описание графа

Итак, граф — это (непустое) множество вершин и множество соединяющих их ребер. Эта структура естественно появляется тогда, когда в задаче есть несколько однотипных объектов и каждый из них может быть связан с произвольным количеством других объектов. В качестве объектов могут быть железнодорожные станции, маршрутизаторы локальных и глобальных сетей и т.п. В первом случае ребра графа — это дороги между станциями, во втором — каналы связи (кабельные, оптоволоконные, спутниковые и т.д.).

Для человека наиболее естественно использовать изображения графов (рисунки, схемы) для объяснения связей между элементами какой-то системы. Поскольку информатика занимается автоматической обработкой данных с помощью компьютеров, сразу возникает вопрос: “Как представить информацию о графе в памяти компьютера?” Хранить ее в виде рисунка (растрового или векторного) неэффективно, потому что рисунок предназначен для восприятия человеком, а не компьютером. Компьютеру удобнее всего хранить информацию в виде таблиц (массив тоже можно считать простейшей таблицей). Для описания графа часто используют квадратную таблицу, которая описывает все возможные связи между узлами (без учета дублирования). Если, например, на пересечении строки **A** и столбца **B** записано число 1, это означает, что есть ребро, соединяющее вершины **A** и **B**; число 0 в этой ячейке означает, что такого ребра нет. Такую таблицу называют *матрицей смежности*. На рисунке показаны схема дорог, соответствующий ей граф и его матрица смежности:



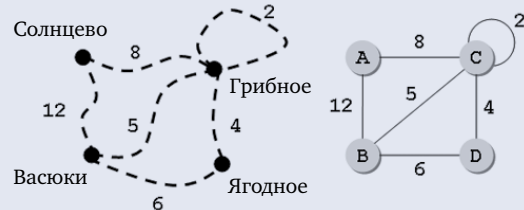
	A	B	C	D
A	0	1	1	0
B	1	0	1	1
C	1	1	1	1
D	0	1	1	0

Единица на главной диагонали (выделенной серым цветом) показывает, что в графе есть петля — ребро, которое начинается и заканчивается в одной и той же вершине.

Обратите внимание, что матрица смежности симметрична относительно главной диагонали, то есть если существует ребро из вершины **A** в вершину **B**, то существует и ребро из **B** в **A**. Такой граф называют *неориентированным* — ребра не имеют направления и каждое из них учтено в матрице смежности дважды.

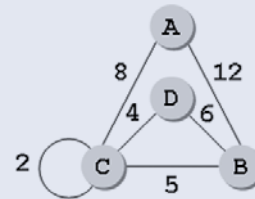
Во многих практических задачах (например, при определении кратчайшего пути из одной вершины

в другую) важную роль играет не только наличие связей между вершинами, но и “длины” этих связей, которые в теории графов называют “весами” (от слова “вес”). Весом может быть, например, длина дороги или стоимость авиаперелета. В этом случае вместо матрицы смежности используют *весовую матрицу*, в клетках которой записывают веса ребер, а если ребра нет, то клетку оставляют пустой. На рисунке показана схема, на которой указаны длины дорог, соответствующий ей граф и его весовая матрица:



	A	B	C	D
A		12	8	
B	12		5	6
C	8	5	2	4
D		6	4	

Заметим, что весовая матрица никак не определяет взаимное расположение вершин. Например, рассмотренный выше граф можно нарисовать совсем иначе, например, так:



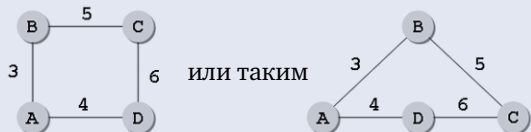
Однако с точки зрения теории графов это будет тот же самый граф, поскольку весовая матрица не изменилась. По аналогии можно вспомнить, что в математике записи 0,5, 1/2, 3/6 и 5/10 обозначают одно и то же число.

Что можно выяснить с помощью весовой матрицы? Во-первых, определить, есть ли ребро между двумя заданными вершинами, и если есть, какова его длина (вес). Для этого достаточно посмотреть в соответствующую ячейку. Например, значение, выделенное кружком на рисунке, показывает, что между вершинами **B** и **C** есть ребро с весом 5:

	A	B	C	D
A		3		4
B	3		5	
C		5		6
D	4		6	

Предполагая, что веса ребер обозначают расстояния между вершинами, можно определить длину пути, проходящего через заданные вершины, — сумму длин ребер, составляющих этот путь. В данном случае длина замкнутого пути **ABCD** складывается из длин ребер **AB**, **BC**, **CD** и **DA**, которые определяют по таблице. Получаем: 3 + 5 + 6 + 4 = 18.

Наконец, полезно научиться рисовать граф по заданной весовой матрице. Нужно только помнить, что это можно сделать разными способами. Для последней приведенной матрицы рисунок может быть, например, таким:



После того как освоены основные понятия, можно переходить непосредственно к задачам, которые включены в ЕГЭ и ГИА.

Вопросы и задачи:

1) Как по весовой матрице графа определить количество ребер (количество петель)?

2) Как можно обозначить отсутствие связи между вершинами при хранении весовой матрицы в памяти реального компьютера (рассмотрите разные варианты)?

3) Для каждой приведенной ниже весовой матрицы:

- определите вес ребра между вершинами **В** и **Д** (если оно есть);

- предполагая, что веса ребер обозначают расстояния между вершинами, определите длину пути **ABDCEA**;

- укажите, какой из трех путей — **ABDC**, **ADEC** или **AEBC** — самый короткий, а какой самый длинный:

а)

	А	В	С	Д	Е
А		4	3	8	7
В	4		4	2	2
С	3	4		6	3
Д	2	2	6		1
Е	7	8	3	1	

б)

	А	В	С	Д	Е
А		2	5	7	6
В	2		1	3	4
С	5	1		9	8
Д	7	3	9		1
Е	6	4	8	1	

в)

	А	В	С	Д	Е
А		1	2	3	9
В	1		5	4	1
С	2	5		5	6
Д	3	4	5		7
Е	9	1	6	7	

г)

	А	В	С	Д	Е
А		5	2	1	6
В	5		3	7	2
С	2	3		4	9
Д	1	7	4		8
Е	6	2	9	8	

3. Кратчайший путь

Одна из самых известных задач, связанных с графами, — определение длины кратчайшего пути из одной вершины в другую. Человек, знакомый с теорией графов, сразу скажет, что нужно использовать алгоритм Дейкстры [1–3, 6], однако на самом деле это не обязательно. Алгоритм Дейкстры действительно хорош, но он предназначен для автоматического решения задач этого типа. Автоматическое решение требуется, когда

- исходные данные заранее неизвестны (поступают из другого источника, например, весовая матрица передается в виде файла);

- необходимо решать множество однотипных задач;

- необходимо решать сложные задачи (для матриц с большим числом узлов и ребер).

Если же нужно решить одну простую задачу, применение универсального, но непростого алгоритма Дейкстры — это, как говорят, “из пушки по воробьям”. Кроме того, большинство школьников не помнят наизусть алгоритм Дейкстры, даже если он изучался в профильном курсе информатики. Поэтому мы будем говорить о простых методах решения, которые доступны всем, кто изучает информатику даже на базовом уровне.

Рассмотрим задачу № 3 из демонстрационного варианта ГИА [5].

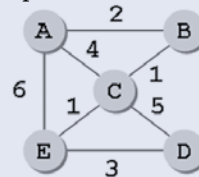
Между населенными пунктами **А**, **В**, **С**, **Д**, **Е** построены дороги, протяженность которых приведена в таблице. Определите кратчайший путь между пунктами **А** и **Д** (при условии, что передвигаться можно только по построенным дорогам).

	А	В	С	Д	Е
А		2	4		6
В	2		1		
С	4	1		5	1
Д		1	5		3
Е	6		1	3	

Уточним, что на самом деле в задаче требуется определить не сам кратчайший путь (последовательность ребер), а его длину. В условии дано еще 4 варианта ответа, но в данном случае они только мешают, поэтому мы их не приводим. Хотя задача сформулирована как практическая, она сводится к поиску кратчайшего пути в графе.

Способ 1. Построение графа.

По заданной таблице (весовой матрице графа) определяем, что граф содержит 7 ребер: **AB** (длиной 2), **AC** (4), **AE** (6), **BC** (1), **CD** (5), **CE** (1) и **DE** (3). Нарисуем граф, соответствующий этим данным (вершины можно располагать как угодно):

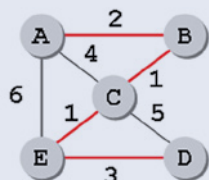


Теперь перебираем все возможные пути из вершины **А** в вершину **Д**, не проходящие дважды через одну и ту же вершину, и считаем их длины:

- ABCD:** 2 + 1 + 5 = 8
- ABCED:** 2 + 1 + 1 + 3 = 7
- ACD:** 4 + 5 = 9
- ACED:** 4 + 1 + 3 = 8
- AECD:** 6 + 1 + 5 = 12
- AED:** 6 + 3 = 9

Чтобы не запутаться и не потерять какой-то путь, мы сначала рассмотрели все пути, начинающиеся с ребра **AB**, затем — все пути, начинающиеся с ребра **AC**, и т.д. Более того, пути перебираются в так называемом *лексикографическом порядке*, то есть в таком порядке, в каком они были бы расположены в словаре (в данном случае — в английском). Такой порядок делает перебор систематическим и поэтому уменьшает вероятность пропуска какого-то пути.

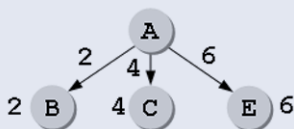
Сравнивая полученные длины, находим, что кратчайший путь **ABCED** (он выделен красным цветом) имеет длину 7:



Прелесть этого способа состоит в том, что в простых задачах можно сразу увидеть ответ, перебрав все варианты в уме и таким образом сэкономив время. Однако при этом есть риск пропустить какой-то вариант (на это и рассчитаны отвлекающие варианты ответов — *дистракторы*).

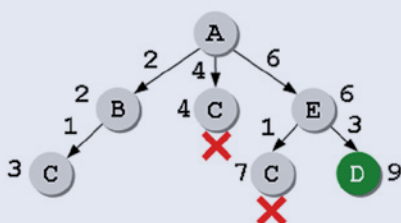
Способ 2. Построение дерева возможных путей.

Можно использовать другой способ, при котором явно строится схема возможных путей в графе (структура типа «дерево»). Сначала по таблице определяем, что из исходной вершины **A** можно ехать только в **B**, **C** и **E**:



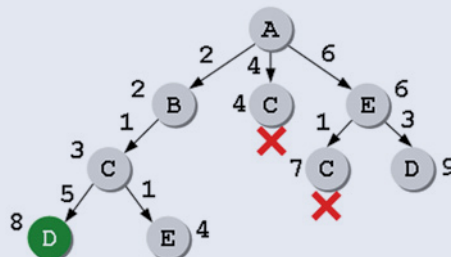
Числа около стрелок обозначают длины дорог (веса соответствующих ребер), а числа около вершин — расстояния от начальной вершины **A** по этому пути.

Далее рассматриваем все пути, состоящие из двух ребер: из **B** можно ехать в **C** (или вернуться в **A**, но такой вариант нас не интересует), из **C** — в **B**, **D** и **E**, а из **E** — в **C** и **D**. Однако первый же из рассмотренных путей, **ABC**, имеет длину 3, что меньше, чем длина ребра **AC** (4). Поэтому все пути, начинающиеся с ребра **AC**, можно далее вообще не рассматривать — в любом случае переезд из **A** в **C** через **B** будет на 1 короче, чем напрямую. По этой же причине не нужно рассматривать пути, начинающиеся с **AEC** (длина этого пути равна 7, что больше уже известного пути в **C** длиной 3):

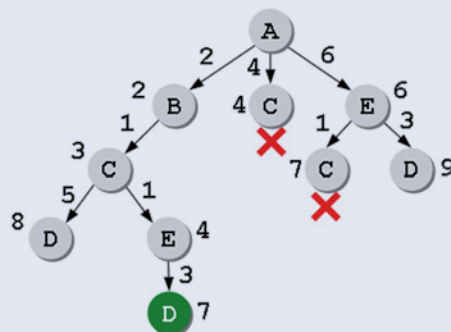


Итак, мы нашли один путь (**AED**) из **A** в **D**, который имеет длину 9. Пока запоминаем его, однако нужно рассмотреть еще ветку **ABC**, которая может дать более короткий путь.

По таблице находим, что из **C** можно ехать в **B**, **D** и **E**. Возвращаться в **B** нет смысла, поэтому рассматриваем последние два варианта:



Получаем еще один (лучший на данный момент!) путь **ABCD** из **A** в **D** длиной 8. С другой стороны, длина пути **ABCE** меньше, чем длина предыдущего известного пути из **A** в **E** (путь **AE**, длина 6), поэтому на этой ветке, возможно, удастся еще улучшить результат. И действительно, из **E** имеет смысл ехать только в **D** (возвращаться в **A** или в **C** не стоит), и этот путь имеет длину 7:



Таким образом, мы построили *дерево*, которое учитывает все возможные пути. Некоторые ветки дерева (**AC** и **AEC**) отсечены, потому что эти пути заведомо не улучшают результат. Длина кратчайшего пути **ABCED** равна 7.

Способ 3. Перебор возможных путей без построения дерева.

Сначала выпишем все ребра, соединяющие начальную вершину **A** с другими вершинами, и их длины:

- AB:** 2
- AC:** 4
- AE:** 6

Теперь рассмотрим все пути, состоящие из двух ребер. Получив путь **ABC** длиной 3, сразу отбрасываем все пути, проходящие через ребро **AC** (так же, как в способе 2):

- ABC:** 2 + 1 = 3
- AEC:** 6 + 1 = 7
- AED:** 6 + 3 = 9 (цель достигнута)

Видим, что все пути, проходящие через **AEC**, тоже можно отбросить. Остается проверить ветку **ABC**:

- ABCD:** 3 + 5 = 8 (цель достигнута)
- ABCE:** 3 + 1 = 4

Продолжая последнюю оставшуюся ветку **АВСЕ**, находим:

АВСЕД: $4 + 3 = 7$ (цель достигнута)

Нерассмотренных путей, которые могли бы улучшить решение, больше не осталось, поэтому выбираем лучший путь: **АВСЕД** длиной 7.

Способ 4. Использование алгоритма Дейкстры [1–3, 6].

Здесь мы не будем приводить детальное описание алгоритма Дейкстры, которое можно найти в указанной литературе, а просто покажем, как с его помощью можно решить предложенную задачу.

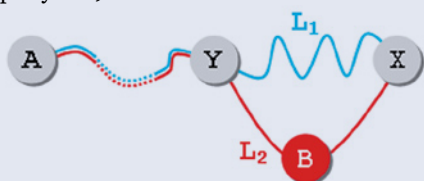
Составляем таблицу, в первой строке выписываем все вершины, кроме начальной вершины **А**; во вторую строку заносим длины ребер, соединяющих начальную вершину с данной (если такое ребро есть), а в третью записываем имя начальной вершины:

В	С	Д	Е
2	4		6
А	А	А	А

Первая строка меняться не будет, поэтому она выделена цветом. Во второй строке будет храниться длина кратчайшего (на данный момент) пути из **А** в соответствующую вершину. Нижняя строка хранит предпоследнюю вершину на этом пути. Пока мы учли только пути, состоящие из одного ребра, поэтому в нижней строке везде указана вершина **А**. Постепенно мы будем расширять множество “учтенных” путей и в конце концов найдем длину действительно кратчайшего пути в каждую вершину.

На каждом этапе работы алгоритма Дейкстры окончательно определяется длина кратчайшего пути до одной из вершин. Сначала ищем минимальное значение во всей второй строке таблицы — это число 2 для вершины **В**. Можно доказать, что это и есть истинная длина кратчайшего пути из **А** в **В**.

Проверяем, не удастся ли сократить путь, если ехать в остальные вершины через вершину **В**. Здесь используется очень простая идея: дорога из некоторой вершины **У** в другую вершину **Х** через **В** может оказаться короче, чем напрямую. Например, предположим, что среди всех рассмотренных ранее путей из **А** в **Х** лучшим является путь, включающий ребро **УХ**, и его длина равна L_1 (синяя линия на рисунке).



Теперь “подключаем” вершину **В**, и может оказаться, что сумма длин дорог **УВ** и **ВХ** будет меньше, чем длина дороги **УХ**, поэтому длина L_2 пути через **В** (красная линия) будет меньше, чем L_1 .

В нашем случае, “подключая” вершину **В**, мы рассматриваем пути, в которых последние звенья —

это **ВС**, **ВД** и **ВЕ**. В данном графе ребер **ВД** и **ВЕ** нет, поэтому нужно лишь проверить, не будет ли путь **АВС** короче, чем **АС**. Оказывается, это действительно так: вместо того чтобы ехать по прямой дороге из **А** в **С** (ее длина 4), лучше ехать из **А** в **В**, а затем из **В** в **С** (общая длина пути 3). Поэтому длина кратчайшего пути во второй строчке меняется на 3, а предпоследняя вершина в третьей строке — на **В**:

В	С	Д	Е
2	3		6
А	В	А	А

На следующем этапе вершину **В** уже не рассматриваем (серый фон означает, что этот столбец далее не будет изменяться):

В	С	Д	Е
2	3		6
А	В	А	А

Среди незакрашенных клеток второй строки ищем минимальное значение — это 3 для вершины **С**. Проверяем, не позволяет ли объезд через **С** сократить путь. Во-первых, появляется путь из **АСД**, его длина равна $4 + 5 = 9$:

В	С	Д	Е
2	3	9	6
А	В	С	А

Кроме того, вместо пути **АЕ** (длиной 6) можно ехать сначала в **С** (длина 3), а затем — в **Е** (длина 1), то есть фактически использовать путь **АВСЕ** длиной $3 + 1 = 4$:

В	С	Д	Е
2	3	9	4
А	В	С	С

Теперь вершина **С** тоже исключается из рассмотрения, из оставшихся вершин значение во второй строке минимально для вершины **Е**:

В	С	Д	Е
2	3	9	4
А	В	С	С

Если ехать из **А** сначала в **Е** (длина 4), а затем по дороге **ЕД** (длина 3), то общая длина пути оказывается равна 7, и это позволяет улучшить результат: кратчайший путь из **А** в **Д** имеет длину 7:

В	С	Д	Е
2	3	7	4
А	В	Е	С

В этой задаче не требуется определять сам путь (важна только его длина), но в третьей строке таблицы есть все данные для этого. Путь “раскручивается” с конечной вершины **Д**: в третьей строке видим, что в нее мы попадаем из **Е**. Смотрим далее: в **Е** приезжаем из **С**, в **С** — из **В**, а в **В** — из **А**. Поэтому кратчайший путь — **АВСЕД** длиной 7.

Задачи для тренировки

Между населенными пунктами **А**, **В**, **С**, **Д**, **Е**, **Ф** построены дороги, протяженность которых приведена в таблице. Отсутствие числа в ячейке таблицы означает, что прямой дороги между пунктами нет. Определите длину кратчайшего пути между пунк-

тами **A** и **F** (при условии, что передвигаться можно только по построенным дорогам).

а) демонстрационный вариант ЕГЭ, ответ: 9)

	A	B	C	D	E	F
A		2	4			
B	2		1		7	
C	4	1		3	4	
D			3		3	
E		7	4	3		2
F					2	

б) (ответ: 17)

	A	B	C	D	E	F
A		5				
B	5		9	3	8	
C		9			4	
D		3			2	
E		8	4	2		7
F					7	

в) (ответ: 14)

	A	B	C	D	E	F
A		4				
B	4		6	3	6	
C		6			4	
D		3			2	
E		6	4	2		5
F					5	

г) (источник — <http://ege.yandex.ru>, ответ: 15)

	A	B	C	D	E	F
A			3			
B			9		4	
C	3	9		3	8	
D			3		2	
E		4	8	2		7
F					7	

д) (ответ: 18)

	A	B	C	D	E	F
A			2			
B			6		3	
C	2	6		5	7	
D			5		4	
E		3	7	4		9
F					9	

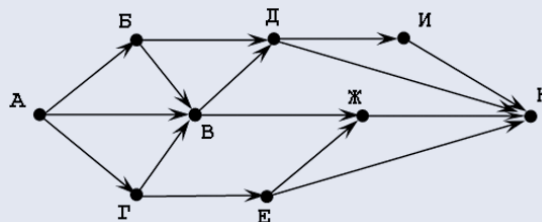
3. Количество путей

В этом году в ЕГЭ и ГИА введена совсем новая задача, в которой требуется определить количество различных путей из одной вершины в другую. Как показали первые обсуждения, она представляет

сложность даже для многих учителей информатики, не говоря уже про учеников.

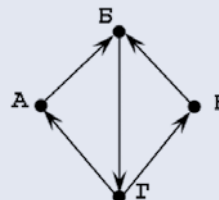
Рассмотрим задачу В9 из демонстрационного варианта ЕГЭ [4]:

На рисунке — схема дорог, связывающих города **A**, **B**, **В**, **Г**, **Д**, **Е**, **Ж**, **И**, **К**. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города **A** в город **К**?



Если говорить на языке теории графов, в этой задаче используется *ориентированный граф* (или кратко *орграф*) — по каждому ребру можно двигаться только в одном направлении, указанном стрелкой (аналогия из жизни — дорога с односторонним движением). Отметим, что если предположить, что движение возможно в обоих направлениях, количество путей будет бесконечно (можно сколько угодно ездить туда и обратно между соседними городами).

Тогда возникает вопрос — а для любого ли ориентированного графа количество путей из одной вершины в другую конечно? Простой пример показывает, что это не так:



Действительно, здесь есть циклы **АБГ** и **БГВ**, в которых можно “крутиться” бесконечно долго. Таким образом, для разрешимости задачи необходимо, чтобы в графе не было циклов.

Теперь займемся решением. Число вершин в таких задачах достаточно велико (здесь — 9), поэтому матрица смежности размером 9×9 при ручном решении нам вряд ли поможет.

Способ 1. Перебор всех возможных путей “методом наблюдения”.

Попробуем найти все возможные пути, глядя на схему. Сначала рассмотрим пути, начинающиеся с отрезка **АВ**. Чтобы не запутаться, будем начинать с верхней части графа, постепенно “спускаясь” ниже и ниже:

АВДИК, АВДК, АВВДИК, АВВДК, АВВЖК

Теперь выпишем пути, которые начинаются с ребра **АВ** (также сверху вниз):

АВДИК, АВДК, АВЖК

и, наконец, пути, начинающиеся с участка **АГ**:

АГВДИК, АГВДК, АГВЖК, АГЕЖК, АГЕК.

Всего получилось 13 путей. Это наиболее простой метод, но он нередко приводит к неправильному результату, потому что легко пропустить какой-то путь.

Способ 2. Пошаговое построение всех возможных путей.

Сначала рассмотрим все вершины, куда можно попасть из начальной вершины **А** за один шаг:

- АБ**
- АВ**
- АГ**

Теперь выписываем все пути из вершины **А**, содержащие два ребра (продолжения уже построенных путей на одно ребро):

- АБД АБВ**
- АВД АВЖ**
- АГВ АГЕ**

Следующий шаг (в рамку обведены пути, дошедшие до конечного пункта **К**):

- АБДИ** АБДК **АБВД** **АБВЖ**
- АВДИ** АВДК АВЖК
- АГВД** **АГВЖ** **АГЕЖ** АГЕК

Итак, четыре пути из **А** в **К** мы уже нашли. Продолжая движение по остальным маршрутам, найдем семь путей, состоящих из четырех ребер:

- АБДИК АБВДИ АБВДК АБВЖК
- АВДИК
- АГВДИ** АГВДК АГВЖК АГЕЖК

и далее два пути из пяти ребер:

- АБВДИК
- АГВДИК

Таким образом, всего найдено $4 + 7 + 2 = 13$ путей, причем по построению других путей нет (в списке не осталось путей, не доведенных до конечной точки).

На взгляд автора, этот способ более надежен, чем предыдущий, поскольку на каждом шаге нас интересуют только ребра, исходящие из одной вершины. Это значит, что нужно анализировать не весь граф, а только его небольшую часть, поэтому труднее ошибиться.

К сожалению, оба уже рассмотренных способа плохо работают, если граф сложный и количество путей велико (несколько десятков). В этом случае приходится применять “тяжелую артиллерию” — методы, которые не требуют перечисления всех возможных путей, но позволяют найти их количество “малой кровью”. В их основе лежит следующая довольно простая и очевидная идея: если в некоторую вершину **Х** можно попасть только из вершин Y_1, Y_2, \dots, Y_m , то количество путей из начальной точки **А** в **Х** вычисляется как

$$N_x = N_{y_1} + N_{y_2} + \dots + N_{y_m}, \quad (*)$$

где N_z обозначает количество путей из вершины **А** в некоторую вершину **z**.

Способ 3. Подстановка.

Для начала заметим, что для вершин, в которые можно приехать только из **А**:

$$N_B = N_A = 1.$$

Применим формулу (*) для конечной вершины **К**, учитывая, что в **К** можно попасть только из **И**, **Д**, **Ж** и **Е**:

$$N_K = N_I + N_D + N_J + N_E.$$

Теперь нужно записать аналогичные выражения для слагаемых в правой части и подставить их в предыдущую формулу. Повторяя эту процедуру несколько раз, мы сведем правую часть к арифметическому выражению, включающему только известные величины N_B и N_A .

Применяем формулу (*) для вершин **И**, **Д**, **Ж** и **Е**:

$$N_I = N_D, N_D = N_B + N_B, N_J = N_B + N_E, N_E = N_A.$$

Подставляем эти выражения в формулу для N_K :

$$N_K = N_D + N_D + N_J + N_E = 2(N_D + N_B) + N_B + 2N_E = 2N_B + 3N_B + 2N_A.$$

Поскольку $N_B = N_A = 1$, сразу получаем

$$N_K = 4 + 3N_B.$$

Вычисляем N_B , снова применяя формулу (*) и учитывая прямой путь из **А**:

$$N_B = 1 + N_A + N_A = 3,$$

поэтому $N_K = 4 + 3 \cdot 3 = 13$.

Обратите внимание, что использованный здесь подход (подстановка) фактически сводит решение исходной задачи (вычисление N_K) к решению нескольких более простых задач того же типа (вычислению N_I, N_D, N_J и N_E) до тех пор, пока мы не придем к простейшим задачам с известными решениями ($N_B = N_A = 1$). Такой прием, называемый *динамическим программированием* [6], применяется также при решении задачи С3 из демонстрационного варианта ЕГЭ-2012 [4].

Способ 4. Подстановка-2.

Попробуем систематизировать применение подстановки и тем самым уменьшить вероятность случайной ошибки. Для этого выпишем для каждой вершины список всех вершин, из которых можно в нее попасть:

- Б** ← **А** **В** ← **АБГ** **Г** ← **А**
- Д** ← **БВ** **Е** ← **Г** **Ж** ← **ВЕ**
- И** ← **Д** **К** ← **ИДЖЕ**

Теперь составляем таблицу: в первом столбце будем записывать имя вершины, во втором — из каких вершин можно в нее попасть, и в третьем — количество путей из начальной вершины в данную. Очевидно, что в третьем столбце можно сразу поставить единицы для тех вершин, в которые можно попасть только из **А** (это **Б** и **Г**):

Вершина	Предыдущие	N
Б	А	1
В	АБГ	
Г	А	1
Д	БВ	
Е	Г	
Ж	ВЕ	
И	Д	
К	ИДЖЕ	

Далее ищем все вершины, куда можно доехать через вершины с уже известным количеством путей (**А**, **Б** и **Г**), — это **В** и **Е**:

Вершина	Предыдущие	N
Б	А	1
В	АБГ	3
Г	А	1
Д	БВ	
Е	Г	1
Ж	ВЕ	
И	Д	
К	ИДЖЕ	

Теперь можно определить количество путей для вершин д и ж

Вершина	Предыдущие	N
Б	А	1
В	АБГ	3
Г	А	1
Д	БВ	4
Е	Г	1
Ж	ВЕ	4
И	Д	
К	ИДЖЕ	

затем — для вершины и и, наконец, для конечной вершины к:

Вершина	Предыдущие	N
Б	А	1
В	АБГ	3
Г	А	1
Д	БВ	4
Е	Г	1
Ж	ВЕ	4
И	Д	4
К	ИДЖЕ	13

Таким образом, правильный ответ — 13 путей.

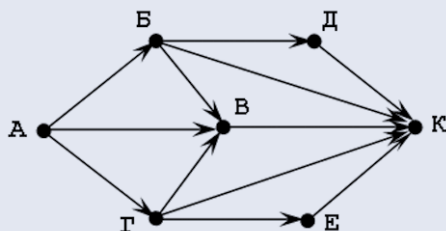
Мы заполняли таблицу в определенном порядке, каждый раз “выискивая” вершины, для которых мы уже можем подсчитать количество путей (то есть для всех предыдущих вершин количество путей уже известно). Фактически при этом выполняется *топологическая сортировка* вершин графа [7], но изучать ее на школьном уровне совершенно необязательно.

Отметим еще раз, что нам не потребовалось выписывать сами пути, поэтому этот метод хорошо работает и для сложных графов с большим количеством путей.

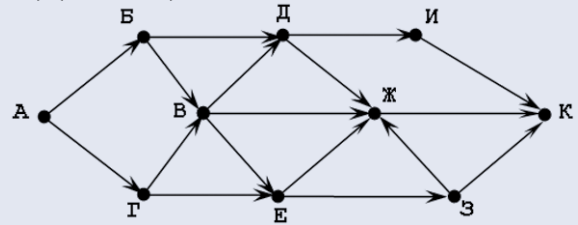
Задачи для тренировки

На рисунке показана схема дорог. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города **а** в город **к**?

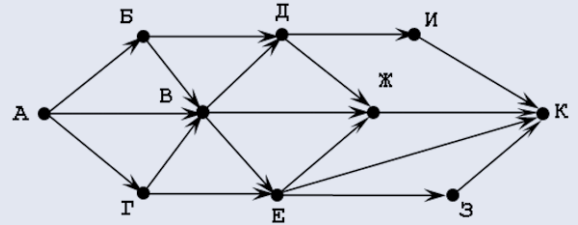
а) (демонстрационный вариант ГИА, ответ: 7)



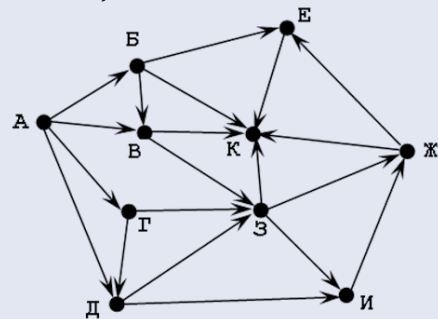
б) (ответ: 17)



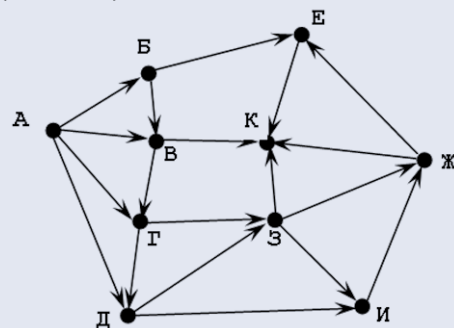
в) (ответ: 23)



г) (ответ: 33)



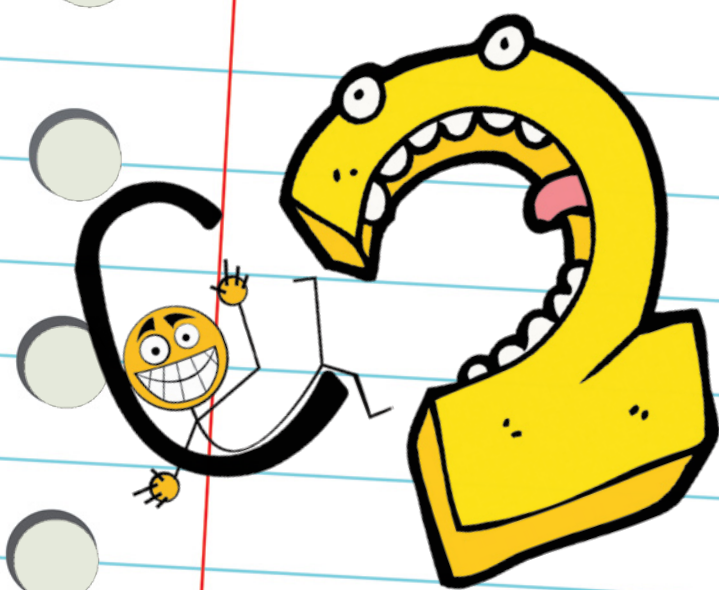
д) (ответ: 46)



Автор благодарит д. ф.-м. н. М.А. Ройтберга за полезное обсуждение данного материала и ценные замечания.

Литература

1. Кристофидес Н. Теория графов. Алгоритмический подход. М.: Мир, 1978.
2. Оре О. Теория графов. М.: Наука, 1968.
3. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. Часть VI. Алгоритмы для работы с графами. М.: Вильямс, 2006.
4. Демонстрационный вариант КИМ для проведения ЕГЭ 2012 года по информатике (<http://www.fipi.ru/view/sections/222/docs/578.html>).
5. Демонстрационный вариант КИМ для проведения ГИА 2012 года по информатике (<http://www.fipi.ru/view/sections/223/docs/579.html>).
6. Поляков К.Ю., Шестаков А.П., Еремин Е.А. Алгоритмизация и программирование // Информатика, № 17, 2011, с. 4–33.
7. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. Глава 22.4. Топологическая сортировка. М.: Вильямс, 2006.



Задачи С2 из ЕГЭ по информатике

Д.М. Златопольский,
Москва

► В демонстрационных вариантах ЕГЭ по информатике и ИКТ последних лет [1–6], а также в книгах от разработчиков ЕГЭ [7–8] традиционно представлена задача, обозначенная как С2. В ней, как указывается в [9], проверяются “умения написать короткую (10–15 строк) простую программу (например, обработки массива) на языке программирования или записать алгоритм на естественном языке”. В данной статье приведена методика решения задач С2 из указанных источников. При анализе используется школьный алгоритмический язык. Соответствующие фрагменты программ на языке Паскаль представлены на диске к данному номеру “Информатики”.

Обратим внимание на то, что, начиная с демонстрационного варианта ЕГЭ 2010 года, в условии задачи С2 приводится перечень переменных величин, используемых в программе (“Исходные данные объявлены так, как показано ниже”), и требование, касающееся их использования (“Запрещается использовать переменные, не описанные

ниже, но разрешается не использовать часть из них”). Пример:

Естественный язык

Объявляем массив A из 30 элементов.

Объявляем целочисленные переменные I, X, Y .

Объявляем вещественную переменную S .

В цикле от 1 до N вводим элементы массива A с 1-го по 30-й.

...

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например, Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учетом синтаксиса и особенностей используемого вами языка программирования.

Учитывая это, мы не будем при решении приводить этапы, связанные с описанием величин, заполнением массива и выводом его на экран.

1. Задача из [4]

Условие

Опишите на русском языке или на одном из языков программирования алгоритм получения из заданного целочисленного массива размером 30 элементов другого массива, который будет содержать модули значений элементов первого массива (не используя специальной функции, вычисляющей модуль числа).

Анализ решения

Ясно, что необходимо использовать два массива размером 30 элементов: заданный и тот, который требуется получить. Пусть их имена, соответственно, $m1$ и $m2$.

Задача “Изменение значений элементов массива с заданными свойствами” рассмотрена, например, в [10]. В ней изменения происходили в заданном массиве (для некоторых элементов). В данном случае необходимо значения элементов массива $m1$, удовлетворяющего условию (они должны быть отрицательными), менять на положительные, но новые значения записывать на соответствующее место в массив $m2$. Другие значения (неотрицательные) остаются без изменения и также записываются на соответствующее место во втором массиве.

Описание фрагмента алгоритма на русском языке

В цикле с параметром i от первого элемента до последнего сравниваем элементы заданного массива с нулем и в зависимости от результата сравнения получаем значения элементов второго массива:

— для отрицательных элементов меняем знак (умножая их значения на -1 или записывая значения со знаком “-”);

— для неотрицательных элементов — значение не меняем.

Каждое i -е значение (измененное или исходное) записываем в элемент второго массива с тем же индексом.

Фрагмент программы

```
нц для i от 1 до n
  если m1[i] < 0
    то
      m2[i] := -m1[i]
  иначе
    m2[i] := m1[i]
все
кц
```

где n — число элементов массивов $m1$ и $m2$.

2. Задача варианта 8 из [7]

Условие

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать произвольные значения. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит номер элемента массива, наименее отличающегося от среднего арифметического всех его элементов.

Анализ решения

Задача представляет собой “сумму” двух частных задач:

1) расчета среднего арифметического всех элементов массива;

2) поиска индекса элемента, значение которого наименее отличается от найденного среднего арифметического.

Вторая частная задача аналогична “классической” задаче поиска индекса минимума (см., например, [11]), с той разницей, что рассматриваться должны не значения элементов массива, а их “отклонения” от среднего арифметического. При этом, поскольку эти “отклонения” могут быть как положительными, так и отрицательными, сравнения должны проводиться с использованием функции, возвращающей модуль числа.

Перечень величин

n — размер массива (константа, равная 40);

m — заданный массив (из n элементов целого типа);

$сумма$ — сумма всех значений элементов массива;
 $сред$ — их среднее арифметическое (величина вещественного типа);

i — индексы элементов;

$номер$ — искомый номер элемента.

Описание фрагмента алгоритма на естественном языке¹

1. Записываем² в переменную $сумма$ нулевое значение.

2. В цикле с параметром i перебираем все элементы от первого до последнего и значение переменной $сумма$ увеличиваем на значение текущего (i -го) элемента.

3. Рассчитываем значение $сред$, деля значение $сумма$ на количество элементов n .

4. Переменной $номер$ присваиваем начальное значение, равное 1.

5. В цикле с параметром i перебираем все элементы массива от второго до n -го. Если модуль разности текущего (i -го) элемента и значения $сред$ оказывается меньше модуля разности значения элемента с индексом $номер$ и значения $сред$, то в переменную $номер$ записываем значение i .

6. Выводим значение переменной $номер$.

Фрагмент программы

```
| Расчет среднего арифметического
сумма := 0
```

```
нц для i от 1 до n
  сумма := сумма + m[i]
```

```
кц
сред := сумма/n
```

```
| Поиск номера элемента с минимальным
| отклонением от среднего арифметического
номер := 1
```

¹ Так записано в [7] и в других источниках.

² Мы используем стиль изложения алгоритма, аналогичный применяемому в [1–8].

```

нц для i от 2 до n
  если abs(m[i] - сред) <
    abs(m[номер] - сред)
  то
    номер := i
  все
кц
|Вывод результата
вывод нс, номер

```

Примечания

1. В решении, приведенном в [7], величина *сумма* используется также для хранения значения среднего арифметического ($сумма := сумма/n$).

2. Аналогично можно найти не номер, а значение соответствующего элемента.

3. Задача из [2]

Условие

Опишите на русском языке или на одном из языков программирования алгоритм поиска номера первого из двух последовательных элементов в целочисленном массиве из 30 элементов, сумма которых максимальна (если таких пар несколько, то можно выбрать любую из них).

Анализ решения

Здесь для каждой пары последовательных элементов необходимо рассчитать сумму значений элементов и найти индекс такого “левого” элемента пары, для которого эта сумма максимальна (последняя задача аналогична задаче поиска индекса максимума). В ходе расчетов переменную величину, хранящую значение суммы для каждой пары, можно не применять, а использовать выражение суммы значений двух последовательных элементов ($m[i] + m[i + 1]$).

Перечень величин

n — размер массива (константа, равная 30);

t — заданный массив (из *n* элементов целого типа);

макс_сумма — максимальное значение суммы двух последовательных элементов среди рассмотренных пар;

i — индексы “левых” элементов в каждой паре;

номер_макс — искомый индекс.

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменную *номер_макс* значение, равное 1, а в переменную *макс_сумма* — сумму значений первого и второго элементов (первая пара).

2. В цикле с параметром *i* перебираем все элементы от второго до предпоследнего и сравниваем сумму *i*-го и (*i* + 1)-го элементов со значением *макс_сумма*. Если сумма двух элементов оказывается больше, то в переменную *номер_макс* записываем значение *i*, а в переменную *макс_сумма* — сумму значений *i*-го и (*i* + 1)-го элементов.

3. Выводим значение переменной *номер_макс*.

Фрагмент программы

```

номер_макс := 1           | Начальные
макс_сумма := m[1] + m[2] | значения
нц для i от 2 до n - 1
  если m[i] + m[i + 1] > макс_сумма
  то
    номер_макс := i
    макс_сумма := m[i] + m[i + 1]
  все
кц
вывод нс, номер_макс

```

Примечание. Аналогично можно найти не индекс, а значение соответствующего элемента.

4. Задача варианта 10 из [7]

Условие

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать произвольные значения. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит номера двух элементов массива, наименее отличающихся друг от друга.

Анализ решения

В данной задаче надо найти не одну, а две величины (два индекса). Рассмотреть все возможные пары элементов с индексами *i* и *j* можно с помощью такого вложенного цикла:

```

нц для i от 1 до n - 1
  нц для j от i + 1 до n

```

```
кц
```

```
кц
```

Для каждой пары индексов *i* и *j* необходимо рассчитать разность значений их элементов и найти пару индексов, для которых эта разность минимальна (последняя задача аналогична задаче поиска индекса минимального элемента [10–11]). Ясно, что здесь, как и при решении задачи 2, сравнения должны проводиться с использованием функции, возвращающей модуль числа. В качестве начальных значений (рассчитываемых до вложенного цикла) можно принять значения, соответствующие индексам 1 и 2.

Перечень величин (кроме индексов *i* и *j*)

n — размер массива (константа, равная 40);

t — заданный массив (из *n* элементов целого типа);

инд1 и *инд2* — искомые номера элементов;

мин_разн — модуль минимальной разности значений пар элементов.

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменную *инд1* значение, равное 1, в переменную *инд2* — равное 2, а в переменную *мин_разн* — равное абсолютному значению разности между $t[инд1]$ и $t[инд2]$.

2. В цикле с параметром *i* перебираем все элементы от первого до предпоследнего, а внутри него в цикле с параметром *j* перебираем все элементы

от $(i + 1)$ -го до последнего. Внутри второго цикла сравниваем:

- 1) абсолютное значение разности между элементами с индексами *инд1* и *инд2*;
- 2) значение *мин_разн*.

Если первое значение оказывается меньше второго, то в переменную *мин_разн* записываем первое значение, в переменную *инд1* — значение, равное *i*, в переменную *инд2* — равное *j*.

3. Выводим значения переменных *инд1* и *инд2*.

Фрагмент программы

```
инд1 := 1
инд2 := 2
|Начальное значение величины мин_разн
мин_разн := abs(m[инд1] - m[инд2])
нц для i от 1 до n - 1
  нц для j от i + 1 до n
    если abs(m[i] - m[j]) < мин_разн
      то
        мин_разн := abs(m[i] - m[j])
        инд1 := i
        инд2 := j
      все
    кц
  кц
вывод нс, инд1, " ", инд2
```

Примечание. Аналогично можно найти не индексы, а значения соответствующих элементов.

5. Задача варианта 9 из [7]

Условие

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать произвольные значения. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит номера двух элементов массива, сумма которых минимальна.

Анализ решения

Данную задачу можно решить следующим образом: рассмотреть все возможные пары элементов массива так, как это делалось в задаче 4, и из всех пар найти такую пару, для которой сумма значений элементов минимальна (последняя задача также аналогична решавшейся в предыдущей задаче). Конечно, функцию *abs* здесь использовать необходимости нет.

Фрагмент соответствующей программы

```
|Начальные значения величин
инд1 := 1
инд2 := 2
мин_сумма := m[1] + m[2]
нц для i от 1 до n - 1
  нц для j от i + 1 до n
    если m[i] + m[j] < мин_сумма
      то
        мин_сумма := m[i] + m[j]
        инд1 := i
        инд2 := j
      все
    кц
  кц
вывод нс, инд1, " ", инд2
```

При использовании описанного метода решения общее число сравнений в программе примерно равно $n^2/2$. Это число можно значительно сократить, если понять, что искомая минимальная сумма будет у двух элементов, которые стояли бы на двух первых местах, если массив был упорядочен по неубыванию. Иными словами, мы пришли к задаче нахождения индексов первого и второго минимумов (при первом толковании понятия “второй минимум” — см. [10–11]). Именно такой вариант решения задачи представлен в [7]. Поэтому приведем его (см. также задачу 21 ниже).

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменную *инд1* значение, равное 1, в переменную *инд2* — равное 2.

2. Сравниваем значения первого и второго элементов массива. Если значение второго элемента массива меньше, чем первого, то записываем в переменную *инд1* значение, равное 2, в переменную *инд2* — равное 1.

3. В цикле с параметром *i* от 3-го элемента до последнего сравниваем текущие элементы с элементом массива с индексом *инд1*:

— если текущий (*i*-й) элемент массива окажется меньше, то в переменную *инд2* записываем значение переменной *инд1*, а в переменную *инд1* — значение индекса текущего элемента *i*;

— в противном случае сравниваем значение текущего элемента с элементом массива с индексом *инд2*; если *i*-й элемент окажется меньше, то в переменную *инд2* записываем значение индекса текущего элемента *i*.

4. Выводим результат (значения *инд1* и *инд2*).

Фрагмент программы

```
|Начальные значения величин
инд1 := 1 |Индекс первого минимума
инд2 := 2 |Индекс второго минимума
если m[2] < m[1]
  то
    инд1 := 2
    инд2 := 1
  все
нц для i от 3 до n
  если m[i] < m[инд1]
    то
      инд2 := инд1;
      инд1 := i
    иначе
      если m[i] < m[инд2]
        то
          инд2 := i
      все
    кц
вывод нс, инд1, " ", инд2
```

Обратим внимание на то, что в [10–11] определялись значения двух первых минимумов, а не их индексов, как в обсуждаемой задаче.

6. Задача варианта 4 из [7]

Условие

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать произвольные значения. С клавиатуры вводится целое число X . Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит наименьший номер элемента массива, равного X , или сообщение, что такого элемента нет.

Анализ решения

Проще всего решить задачу, реализуя поиск элемента, начиная с конца массива:

```
| Поиск искомого номера
номер := 0
нц для i от n до 1 шаг -1
  если m[i] = X
    то
      номер := i
все
кц
| Вывод результата
если номер > 0
  то
    вывод нс, номер
иначе
  вывод нс, "Такого числа в массиве нет"

```

— где:

n — размер массива (константа, равная 30);
 t — заданный массив (из n элементов целого типа);
 X — число, вводимое с клавиатуры;
 i — индексы элементов;
номер — искомый номер элемента.

Однако целесообразно прекратить обработку массива после нахождения (возможного) искомого элемента.

Описание фрагмента соответствующего алгоритма на естественном языке

1. Записываем в переменную *номер* значение, равное 0.

2. В цикле с предусловием, пока значение *номер* меньше или равно n и пока значение $t[\text{номер}]$ не равно X , увеличиваем значение переменной *номер* на 1.

3. Если значение *номер* оказалось равно нулю, то выводим сообщение об отсутствии в массиве чисел, равных X , в противном случае выводим значение переменной *номер*.

Фрагмент соответствующей программы

```
номер := 0
нц пока номер <= n и m[номер] <> X
  номер := номер + 1
кц
| Вывод результата
если номер <= n
  то
    вывод нс, номер
```

иначе

```
вывод нс, "Такого числа
в массиве нет"
```

все

Во фрагменте на языке Паскаль применим вариант, учитывающий возможный выход за пределы массива (см. [10]):

```
номер := 1;
while (номер < n) and
      (m[номер] <> X) do
  номер := номер + 1
{Вывод результата}
if m[номер] <> X
then write('Такого числа
в массиве нет')
else write(номер);
```

7. Задача варианта 2 из [7]

Условие

Дан вещественный массив из 50 элементов. Элементы массива могут принимать произвольные значения. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит наименьший номер отрицательного элемента массива или сообщение, что такого элемента нет.

Задача решается аналогично предыдущей.

8. Задача варианта 1 из [8]

Условие

Опишите на русском языке или на одном из языков программирования алгоритм подсчета суммы всех отрицательных элементов заданного целочисленного массива размером из 30 элементов. Если отрицательных элементов нет, сообщите об этом.

Анализ решения

Задача нахождения суммы элементов массива с заданными свойствами (удовлетворяющих некоторому условию) рассматривалась в [10]. Обратим внимание на возможность отсутствия в массиве отрицательных элементов.

Перечень величин

n — размер массива (константа, равная 30);
 t — заданный массив (из n элементов целого типа);
сумма — сумма всех отрицательных элементов массива;
кол — количество таких элементов;
 i — индексы элементов.

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменные *сумма* и *кол* нулевые значения.

2. В цикле с параметром i от первого элемента до последнего сравниваем элементы заданного массива с нулем и для отрицательных элементов:

- 1) прибавляем к текущему значению переменной *сумма* значение рассматриваемого (i -го) элемента;
- 2) увеличиваем значение переменной *кол* на 1.

3. Сравним значение *кол* с нулем (проверка наличия в массиве отрицательных элементов). Если *кол* больше нуля, то выводим значение *сумма*, в противном случае выводим сообщение “Отрицательных элементов в массиве нет”.

Фрагмент программы

|Расчет значений *сумма* и *кол*

сумма := 0

кол := 0

нц для *i* от 1 до *n*

если *m*[*i*] < 0

то

сумма := *сумма* + *m*[*i*]

кол := *кол* + 1

все

кц

|Вывод результата

если *кол* > 0

то

вывод *нс*, *сумма*

иначе

вывод *нс*, "Отрицательных элементов в массиве нет"

все

Примечание. Величину *кол* можно не использовать. В этом случае сообщение об отсутствии в массиве отрицательных элементов выводится тогда, когда значение величины *сумма* равно нулю.

9. Задача варианта 2 из [8]

Условие

Опишите на русском языке или на одном из языков программирования алгоритм подсчета произведения всех отрицательных элементов заданного целочисленного массива размером 30 элементов в предположении, что в массиве есть хотя бы один отрицательный элемент.

Анализ решения

Данная задача отличается от предыдущей тем, что рассчитывается не сумма, а произведение, а также тем, что случай, когда в массиве нет отрицательных элементов, рассматриваться не должен. Обратим внимание на то, что начальное значение произведения должно быть равно 1.

10. Задача варианта 3 из [8]

Условие

В целочисленном массиве размером 30 элементов задан рост учащихся выпускного класса (в сантиметрах). Опишите на русском языке или на одном из языков программирования алгоритм подсчета количества учащихся, чей рост превосходит 175 см. Если таких учащихся нет, сообщите об этом.

Анализ решения

Это задача типа “Нахождение количества элементов массива с заданными свойствами” [10] с учетом возможности отсутствия в массиве таких элементов.

Перечень величин

n — размер массива (константа, равная 30);

t — заданный массив (из *n* элементов целого типа);

кол — количество элементов, больших 175;

i — индексы элементов.

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменную *кол* нулевое значение.

2. В цикле с параметром *i* от первого элемента до последнего сравниваем элементы заданного массива с числом 175 и для элементов, больших 175, увеличиваем значение переменной *кол* на 1.

3. Сравним значение *кол* с нулем. Если *кол* больше нуля, то выводим соответствующее значение, в противном случае выводим сообщение “Таких учащихся нет”.

Фрагмент программы

Школьный алгоритмический язык

|Расчет значения *кол*

кол := 0

нц для *i* от 1 до *n*

если *m*[*i*] > 175

то

кол := *кол* + 1

все

кц

|Вывод результата

если *кол* > 0

то

вывод *нс*, *кол*

иначе

вывод *нс*, "Таких учащихся нет"

все

11. Задача варианта 4 из [8]

Условие

В вещественном массиве размером 30 элементов задан вес спортсменок одной команды (в килограммах с округлением до десятых). Опишите на русском языке или на одном из языков программирования алгоритм подсчета количества спортсменок, чей вес превышает 50 кг, но не более 57 кг. Если таких спортсменок нет, сообщите об этом.

Анализ решения

Решение аналогично решению предыдущей задачи, с той разницей, что условие для подсчета — сложное (составное):

вес[*i*] > 50 **и** *вес*[*i*] <= 57

12. Задача варианта 6 из [7]

Условие

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать произвольные значения. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит номер третьего положительного элемента массива (если из массива вычеркнуть все

неположительные элементы, этот элемент стоял бы в получившемся массиве на третьем месте). Если в массиве меньше, чем три положительных элемента, вывести об этом сообщение.

Анализ решения

Решать задачу за два прохода по массиву:

1) сначала подсчитать количество положительных элементов;

2) потом в зависимости от этого количества либо вывести сообщение об отсутствии трех положительных элементов, либо на втором проходе найти соответствующий элемент и вывести его — нерационально.

Лучше за один проход подсчитывать количество положительных элементов, а когда это количество станет равно 3 (этого может и не быть) — запомнить номер соответствующего элемента в качестве искомого значения.

Перечень величин

n — размер массива (константа, равная 40);

m — заданный массив (из n элементов вещественного типа);

$кол$ — количество положительных элементов;

i — индексы элементов;

$номер$ — искомый номер элемента.

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменную $кол$ нулевое значение.

2. В цикле с параметром i от первого элемента до последнего сравниваем элементы заданного массива с нулем. Если текущий элемент больше нуля, то:

1) увеличиваем значение переменной $кол$ на 1;

2) сравниваем значение переменной $кол$ с тремя; если оно равно трем, то записываем в переменную $номер$ индекс i текущего элемента.

3. Сравниваем значение $кол$ с числом 3. Если это значение больше либо равно трем, то выводим значение $номер$, в противном случае выводим сообщение "В массиве нет такого элемента".

Фрагмент программы

```
кол := 0
нц для i от 1 до n
  если m[i] > 0
    то
      кол := кол + 1
      если кол = 3
        то
          номер := i
      все
    все
кц
|Вывод результата
если кол >= 3
  то
    вывод нс, номер
  иначе
    вывод нс, "В массиве нет такого элемента"
все
```

13. Задача варианта 5 из [8]

Условие

Опишите на русском языке или на одном из языков программирования алгоритм подсчета среднего значения отрицательных элементов в целочисленном массиве из 30 элементов в предположении, что в массиве есть хотя бы один отрицательный элемент.

Анализ решения

Для расчета среднего арифметического значения элементов массива с заданными свойствами необходимо найти сумму и количество таких элементов (эти две задачи рассмотрены в [10]). Случай, когда в массиве нет отрицательных элементов, рассматривать не нужно.

Перечень величин

n — размер массива (константа, равная 30);

m — заданный массив (из n элементов целого типа);

$сумма$ — сумма всех отрицательных элементов массива;

$кол$ — количество таких элементов;

$сред$ — среднее арифметическое значение отрицательных элементов массива (величина вещественного типа);

i — индексы элементов.

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменные $сумма$ и $кол$ нулевые значения.

2. В цикле с параметром i от первого элемента до последнего сравниваем элементы заданного массива с нулем и для отрицательных элементов:

1) прибавляем к текущему значению переменной $сумма$ значение рассматриваемого (i -го) элемента;

2) увеличиваем значение переменной $кол$ на 1.

3. Рассчитываем значение $сред$ как частное от деления $сумма$ на $кол$.

4. Выводим результат (значение $сред$).

Фрагмент программы

```
|Расчет значений сумма и кол
сумма := 0
кол := 0
нц для i от 1 до n
  если m[i] < 0
    то
      сумма := сумма + m[i]
      кол := кол + 1
    все
кц
|Расчет значения сред
сред := сумма/кол
|Вывод результата
вывод нс, сред
```

Примечание. Величину $сред$ можно не использовать. В этом случае в качестве результата выводится выражение $сумма/кол$.

14. Задача из [5]

Условие

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать значения от 0 до 1000. Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет подсчитать и вывести среднее арифметическое элементов массива, имеющих нечетное значение.

Гарантируется, что в исходном массиве хотя бы один элемент имеет нечетное значение.

Анализ решения

Задача решается аналогично предыдущей. Обратим также внимание на важное обстоятельство, касающееся программ на языке Паскаль, — если бы в массиве могли присутствовать и отрицательные значения, то проверку элемента массива на “нечетность” следующим образом:

$m[i] \bmod 2 = 1$

— проводить было бы нельзя, так как в этом языке остаток от деления нечетного отрицательного числа на 2 равен -1 (правильное условие:

$m[i] \bmod 2 <> 0$).

В языке Си проблема аналогична.

15. Задача варианта 10 из [8]

Условие

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать значения от -20 до 20 — сведения о температуре за каждый день ноября. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит максимальную температуру среди дней, когда были заморозки (то есть температура опускалась ниже нуля). Гарантируется, что хотя бы в один день ноября была отрицательная температура.

Анализ решения

Данная задача относится к типу “Определение максимального значения среди тех элементов массива, которые удовлетворяют некоторому условию”. При этом диапазон значений элементов массива известен. Такая задача обсуждалась в [10].

Перечень величин

n — размер массива (константа, равная 30);

t — заданный массив (из n элементов целого типа);

$макс$ — максимальное значение среди отрицательных элементов массива;

i — индексы элементов.

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменную $макс$ значение, равное -20 (оно не больше минимального из отрицательных элементов массива).

2. В цикле с параметром i от первого элемента до последнего сравниваем элементы заданного массива с нулем и для отрицательных элементов дополнительно сравниваем их значение со значением

переменной $макс$ ³. Если текущий элемент массива больше $макс$, то в качестве нового значения переменной $макс$ принимаем значение этого элемента.

3. Выводим результат (значение $макс$).

Фрагмент программы

```
макс := -20
нц для i от 1 до n
  если m[i] < 0
    то
      если m[i] > макс
        то
          макс := m[i]
      все
    все
кц
вывод нс, макс
```

16. Задача варианта 9 из [8]

Условие

Опишите на русском языке или на одном из языков программирования алгоритм вычисления разности максимального среди элементов, имеющих четные значения, и максимального среди элементов, имеющих нечетные значения, в заданном целочисленном массиве из 30 положительных элементов (в предположении, что в массиве есть и четные, и нечетные элементы).

Анализ решения

В данной задаче по сравнению с предыдущей добавляется также вторая частная задача того же типа, но с другим условием для учета элементов. Причем отдельно второе условие можно не проверять, а рассматривать его как ветвь “иначе” (“if”) полного условного оператора. Конечно, после вычисления двух максимальных значений в качестве ответа нужно вывести их разность.

Перечень величин

n — размер массива (константа, равная 30);

t — заданный массив (из n элементов целого типа);

$макс_чет$ — максимальное значение среди четных элементов массива;

$макс_нечет$ — то же среди нечетных элементов;

i — индексы элементов.

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменные $макс_чет$ и $макс_нечет$ значения, равные нулю (согласно условию, все значения в массиве — положительные, и 0 не больше минимального из четных и не больше минимального из нечетных элементов массива).

2. В цикле с параметром i от первого элемента до последнего проверяем, является ли очередной (i -й) элемент заданного массива четным:

³ Можно также сразу записать сложное условие: $m[i] > 0$ и $m[i] > макс$, однако при этом в программах на некоторых языках будет проводиться полная проверка условия даже в случаях, когда очередной элемент — неотрицательный. Сложное условие можно записывать и в нескольких следующих задачах.

1) если является, то дополнительно сравниваем значение i -го элемента со значением переменной *макс_чет*. Если он больше ($m[i] > \text{макс_чет}$), то в качестве нового значения переменной *макс_чет* принимаем значение этого элемента $m[i]$;

2) если не является, то дополнительно сравниваем значение i -го элемента со значением переменной *макс_нечет*. Если он больше ($m[i] > \text{макс_нечет}$), то в качестве нового значения переменной *макс_нечет* принимаем значение этого элемента.

3. Выводим результат (значение *макс_чет* – *макс_нечет*).

Фрагмент программы

```
макс_чет := 0
макс_нечет := 0
нц для i от 1 до n
  если mod(m[i], 2) = 0
    то
      если m[i] > макс_чет
        то
          макс_чет := m[i]
      все
    иначе
      если m[i] > макс_нечет
        то
          макс_нечет := m[i]
      все
  все
кц
|Вывод результата
вывод нс, макс_чет - макс_нечет
```

17. Задача из [4]

Условие

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 100 — баллы учащихся выпускного класса за итоговый тест по информатике. Для получения положительной оценки за тест требовалось набрать не менее 20 баллов. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит минимальный балл среди учащихся, получивших за тест положительную оценку. Известно, что в классе хотя бы один учащийся получил за тест положительную оценку.

Анализ решения

Данная задача относится к типу “Определение минимального значения среди тех элементов массива, которые удовлетворяют некоторому условию”. При этом известен диапазон значений элементов массива, а также тот факт, что соответствующие элементы в массиве имеются (см. [10]).

Перечень величин

n — размер массива (константа, равная 30);

m — заданный массив (из n элементов целого типа);

мин — минимальное значение среди элементов массива, не меньших 20 (искомая величина);

i — индексы элементов.

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменную *мин* значение, равное 100 (оно не меньше максимального из элементов массива).

2. В цикле с параметром i от первого элемента до последнего сравниваем элементы заданного массива с 20. Если текущий (i -й) элемент массива больше или равен 20, то дополнительно сравниваем его значение со значением переменной *мин*. Если он меньше *мин*, то в качестве нового значения переменной *мин* принимаем значение этого элемента.

3. Выводим результат (значение *мин*).

Фрагмент программы

```
мин := 100
нц для i от 1 до n
  если m[i] >= 20
    то
      если m[i] < мин
        то
          мин := m[i]
      все
    все
кц
вывод нс, мин
```

18. Задача варианта 1 из [7]

Условие

Дан целочисленный массив из 28 элементов. Элементы массива могут принимать значения от 0 до 100 — процент выполнения учащимися домашних заданий по информатике. Для получения положительной оценки за год требовалось набрать не менее 40 баллов. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит минимальный балл среди учащихся, получивших за год положительную оценку. Гарантируется, что в классе хотя бы один учащийся получил за год положительную оценку.

Задача решается аналогично предыдущей задаче.

19. Задача из [6]

Условие

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от -1000 до 1000 . Опишите на русском языке или на одном из языков программирования алгоритм, позволяющий найти и вывести минимальное значение среди элементов массива, которые имеют четное значение и не делятся на три. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого четно и не кратно трем.

Анализ решения

Здесь также требуется определить минимальное значение среди тех элементов массива, которые удовлетворяют некоторому условию. Особенность в том, что это условие — сложное:

$\text{mod}(m[i], 2) = 0$ и $\text{mod}(m[i], 3) \neq 0$

20. Задача варианта 3 из [6]

Условие

Дан вещественный массив из 40 элементов. Элементы массива могут принимать произвольные значения. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит минимальный положительный элемент массива или сообщение, что такого элемента нет.

Анализ решения

Особенность (и сложность!) данной задачи в том, что диапазон значений элементов массива неизвестен и что положительных элементов в массиве может не быть. Такая задача также рассмотрена в [10].

Напомним два основных этапа ее решения:

1. Ищем (пробуем найти) индекс первого положительного элемента в массиве.

2. Если он найден:

то

1) принимаем значение этого элемента в качестве искомого минимального значения;

2) рассматриваем все следующие элементы и положительные из них:

— сравниваем со “старым” минимальным значением. Если текущий положительный элемент меньше минимального, то принимаем его в качестве нового значения искомого значения;

3) выводим найденное минимальное значение иначе выводим сообщение о том, что положительных элементов в массиве нет.

Перечень величин

n — размер массива (константа, равная 40);

m — заданный массив (из n элементов вещественного типа);

$мин$ — минимальное значение среди положительных элементов массива (искомая величина);

i — индекс (возможный) первого (при просмотре массива от его начала) положительного элемента;

j — индексы элементов, большие i .

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменную i значение, равное 1 (начинаем поиск с первого элемента массива).

2. В цикле с предусловием, пока i не больше n и пока i -й элемент — неположительный, увеличиваем значение i на 1.

3. Если после окончания цикла окажется, что i меньше либо равно n , то

1) записываем в переменную $мин$ значение i -го элемента;

2) в цикле с параметром j рассматриваем элементы от $(i + 1)$ -го до последнего и сравниваем их с нулем. Если текущий элемент больше нуля, то сравниваем его со значением $мин$. Если он меньше минимального, то принимаем текущий положительный элемент ($m[j]$) его в качестве нового значения величины $мин$;

3) выводим результат (значение $мин$) иначе выводим сообщение “Таких чисел в массиве нет”.

Фрагмент программы

|Ищем (пробуем найти)

|первый положительный элемент

$i := 1$

нц пока $i \leq n$ **и не** $m[i] > 0$

$i := i + 1$

кц

|Если такой элемент найден

если $i \leq n$

то

|Его индекс равен i

|Принимаем этот элемент в качестве

|минимального

$мин := m[i]$

|Рассматриваем оставшиеся элементы

нц для j **от** $i + 1$ **до** n

|и положительные из них сравниваем

|с минимальным среди уже рассмотренных

если $m[j] > 0$

то

если $m[j] < мин$

то

$мин := m[j]$

все

все

кц

|Выводим ответ

вывод нс, мин

иначе |Положительных элементов

|в массиве нет

|Выводим соответствующее сообщение

вывод нс, "Положительных чисел в массиве нет"

все

Приведем также компактный вариант решения:

$индекс_мин := 0$ |Условно

|Рассматриваем все элементы

нц для i **от** 1 **до** n

если $m[i] > 0$

то |Встретился положительный элемент

|Если это произошло впервые

|или он меньше "старого" минимума

если $индекс_мин = 0$ **или**

$m[i] < m[индекс_мин]$

то

|Запоминаем его индекс

|в качестве значения $индекс_мин$

$индекс_мин := i$

все

все

кц

|Выводим ответ

если $индекс_мин > 0$

то

вывод нс, $m[индекс_мин]$

```

иначе
    вывод нс, "Таких чисел в массиве нет"
все

```

21. Задача варианта 5 из [7]

Условие

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать произвольные значения. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит значение второго максимума (элемента, который в отсортированном по неубыванию массиве стоял бы вторым).

Анализ решения

Задача нахождения второго максимума анализировалась в [10–11]. Здесь принимается первое из двух возможных толкований этого понятия. Еще одна особенность — диапазон значений элементов массива неизвестен.

Перечень величин

n — размер массива (константа, равная 40);
 t — заданный массив (из n элементов целого типа);

макс1 — максимальный элемент массива (первый максимум);

макс2 — второй максимум (искомое значение);

i — индексы элементов.

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменную макс1 значение 1-го элемента массива, в переменную макс2 — 2-го элемента.

2. Сравниваем значения 1-го и 2-го элементов массива. Если 2-й элемент больше, то в переменную макс1 записываем значение 2-го элемента массива, в переменную макс2 — 1-го элемента.

3. В цикле с параметром i от 3-го элемента до последнего сравниваем элементы массива со значением макс1 :

— если текущий (i -й) элемент массива окажется больше макс1 , то в переменную макс2 записываем значение переменной макс1 , а в переменную макс1 — значение текущего элемента массива;

— в противном случае сравниваем значение текущего элемента массива со значением макс2 — если i -й элемент окажется больше макс2 , то в переменную макс2 записываем значение текущего элемента массива.

4. Выводим результат (значение макс2).

Фрагмент программы

```

макс1 := m[1]
макс2 := m[2]
если m[2] > m[1]
    то
        макс1 := m[2]
        макс2 := m[1]
все

```

```

нц для  $i$  от 3 до  $n$ 
    если m[i] > макс1
        то
            макс2 := макс1 |Именно в таком
            макс1 := m[i] |порядке
    иначе
        если m[i] > макс2
            то
                макс2 := m[i]
    все
кц
вывод нс, макс2

```

22. Задача варианта 6 из [8]

Условие

Опишите на русском языке или на одном из языков программирования алгоритм подсчета максимального количества подряд идущих отрицательных элементов в целочисленном массиве длины 30.

Анализ решения

Задача такого типа рассмотрена в статье, представленной на диске к данному номеру. Здесь также для краткости будем называть участок массива с подряд идущими отрицательными элементами “подмассивом”.

Перечень величин

n — размер массива (константа, равная 30);
 t — заданный массив (из n элементов целого типа);

кол_отр — количество отрицательных элементов в текущем подмассиве;

макс_отр — максимальное количество подряд идущих отрицательных элементов в подмассивах (искомая величина);

i — индексы элементов.

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменные кол_отр и макс_отр нулевые значения.

2. В цикле с параметром i рассматриваем элементы от первого до последнего и сравниваем их с нулем. Если текущий элемент меньше нуля, то увеличиваем значение кол_отр на 1, в противном случае:

1) сравниваем значение величин кол_отр и макс_отр . Если первое значение больше второго, в переменную макс_отр записываем значение кол_отр ;

2) в переменную кол_отр записываем нулевое значение.

3. По окончании цикла дополнительно сравниваем значение величин кол_отр и макс_отр . Если $\text{кол_отр} > \text{макс_отр}$, то в переменную макс_отр записываем значение кол_отр .

4. Выводим значение величины макс_отр .

Фрагмент программы

```

кол_отр := 0

```



```

макс_отр := 0
нц для i от 1 до n
  если m[i] < 0
    то
      |Подмассив отрицательных
      |элементов продолжается
      |или начался.
      |Увеличиваем его длину на 1
      кол_отр := кол_отр + 1
  иначе |Встретился неотрицательный
      |элемент.
      |Подмассив отрицательных
      |элементов кончился.
      |Сравниваем его длину
      |со значением макс_отр
      если кол_отр > макс_отр
        то
          макс_отр := кол_отр
      все
      кол_отр := 0 |Новое значение
    все
кц
|Проверяем длину последнего
| (возможного)
|подмассива
если кол_отр > макс_отр
  то
    макс_отр := кол_отр
все
|Выводим ответ
вывод нс, макс_отр

```

Обратим внимание на то, что после окончания текущего подмассива величине *кол_отр* присваивается нулевое значение независимо от результата сравнения величин *кол_отр* и *макс_отр*.

23. Задача варианта 8 из [8]

Условие

Опишите на русском языке или на одном из языков программирования алгоритм подсчета максимального количества подряд идущих четных элементов в целочисленном массиве длины 30.

Задача решается аналогично предыдущей.

24. Задача варианта 7 из [8]

Условие

Опишите на русском языке или на одном из языков программирования алгоритм подсчета максимального количества подряд идущих элементов, каждый из которых больше предыдущего, в целочисленном массиве длины 30.

Анализ решения

Решение данной задачи также во многом аналогично двум предыдущим. Отличия:

- 1) параметр цикла *i* должен меняться от 2 до *n* (или от 1 до *n* - 1);
- 2) условие для подсчета длины подмассива: $m[i] > m[i - 1]$ (или $m[i] < m[i + 1]$).

25. Задача из [2]

Условие

Опишите на русском языке или на одном из языков программирования алгоритм подсчета максимального количества подряд идущих совпадающих элементов в целочисленном массиве длины 30.

Анализ решения

Особенность данной задачи по сравнению с тремя предыдущими в том, что начальные значения переменных величин, хранящих длину текущего подмассива и максимальную длину подмассивов, равны 1 (если во всех парах находящихся рядом элементов массива нет одинаковых, то искомая величина будет равна 1).

Перечень величин

n — размер массива (константа, равная 30);
t — заданный массив (из *n* элементов целого типа);

кол_совп — количество подряд идущих совпадающих элементов в текущем подмассиве;

макс_совп — максимальное количество подряд идущих совпадающих элементов в подмассивах, рассмотренных ранее (искомая величина);

i — индексы элементов.

Описание фрагмента алгоритма на естественном языке

1. Записываем в переменные *кол_совп* и *макс_совп* значения, равные 1.

2. В цикле с параметром *i* рассматриваем элементы от второго до последнего и каждый из них сравниваем с предыдущим. Если текущий (*i*-й) элемент равен предыдущему, то увеличиваем значение *кол_совп* на 1, в противном случае сравниваем значение величин *кол_совп* и *макс_совп*. Если первое значение больше второго, то:

1) в переменную *макс_совп* записываем значение *кол_совп*;

2) в переменную *кол_совп* записываем значение, равное 1.

3. По окончании цикла дополнительно сравниваем значение величин *кол_совп* и *макс_совп*. Если $кол_совп > макс_совп$, то в переменную *макс_совп* записываем значение *кол_совп*.

4. Выводим значение величины *макс_совп*.

Фрагмент программы

```

кол_совп := 1
макс_совп := 1
нц для i от 2 до n
  если m[i] = m[i - 1]
    то
      кол_совп := кол_совп + 1
  иначе
    если кол_совп > макс_совп
      то
        макс_совп := кол_совп
  все
  кол_совп := 1 |Новое значение
все
кц

```

```

если кол_совп > макс_совп
то
    макс_совп := кол_совп
все
вывод макс_совп

```

26. Задача варианта 7 из [7]

Условие

Дан целочисленный массив из 40 элементов. Элементы этого массива могут принимать произвольные значения. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит сумму элементов наибольшей возрастающей последовательности подряд идущих элементов массива.

Анализ решения

Здесь необходимо определить не максимальную длину⁴ (количество элементов) подмассива и не максимальную сумму его элементов, а сумму элементов в подмассиве максимальной длины. Значит, нужно в ходе поиска такого подмассива рассчитывать и запоминать также сумму его элементов. При этом в данном случае необходимо в длине подмассива учитывать и его первый элемент (при решении задачи 24 этот элемент не учитывался).

Перечень величин

n — размер массива (константа, равная 40);
 t — заданный массив (из n элементов целого типа);
 $кол_возр$ — количество элементов возрастающей последовательности в текущем подмассиве;
 $сумма_возр$ — сумма значений элементов в таком подмассиве;
 $макс_кол$ — максимальное количество элементов в возрастающих последовательностях элементов;
 $макс_сумма$ — максимальная сумма значений элементов в соответствующей последовательности;
 i — индексы элементов.

Фрагмент программы

```

кол_возр := 1           |Учитываем первый
сумма_возр := m[1]     |элемент
макс_кол := 1
нц для  $i$  от 2 до  $n$ 
    если  $m[i] > m[i - 1]$ 
        то
            кол_возр := кол_возр + 1
            сумма_возр := сумма_возр +  $m[i]$ 
        иначе
            если кол_возр > макс_кол
                то
                    макс_кол := кол_возр
                    макс_сумма := сумма_возр
            все
                |Новые значения
                сумма_возр :=  $m[i]$ 
                кол_возр := 1
    кц
если кол_возр > макс_кол
    то
        макс_сумма := сумма_возр

```

⁴ Хотя эту величину также придется рассчитывать, но как вспомогательную — см. далее.

|Здесь уточняем только
|значение макс_сумма

все

вывод $нс$, макс_сумма

Обратим внимание на то, что начальные значения величины $кол_возр$ принимаются равными 1. Заметим также, что в решении, приведенном в [7], начальное значение величины $макс_кол$ принимается равным 0. Очевидно, авторы допускают возможность того, что в массиве не будет двух и более элементов, образующих возрастающую последовательность.

В заключение приведем перечень типовых задач программирования, встречающихся в задачах С2 ЕГЭ по информатике (см. с. 27).

Литература

1. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена по информатике и ИКТ 2007 года. / <http://www.fipi.ru/binaries/395/inf%20dem.doc>.
2. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена по информатике и ИКТ 2008 года. / <http://www.fipi.ru/binaries/518/inf.rar>.
3. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена по информатике и ИКТ 2009 года. / <http://www.fipi.ru/binaries/731/infZIP%20-%20WinRAR.zip>.
4. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена по информатике и ИКТ 2010 года. / <http://www.fipi.ru/binaries/895/inf.zip>.
5. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2011 года по информатике и ИКТ. / http://www.fipi.ru/binaries/1079/inf_10_11_10.zip.
6. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2012 года по информатике и ИКТ (проект). / <http://www.fipi.ru/binaries/1232/infEGE2012.zip>.
7. Самое полное издание типовых вариантов реальных заданий ЕГЭ-2010: Информатика. / Авт.-сост. П.А. Якушкин, Д.М. Ушаков. М.: Астрель, 2010 (Федеральный институт педагогических измерений).
8. Якушкин П.А., Лещинер В.Р., Кириенко Д.П. ЕГЭ-2010. Информатика. Типовые тестовые задания. М.: Изд-во "Экзамен", 2010.
9. Спецификация контрольных измерительных материалов единого государственного экзамена 2012 года по информатике и ИКТ. / <http://www.fipi.ru/binaries/1232/infEGE2012.zip>.
10. Ушаков Д.М. Задачи на обработку массивов в ЕГЭ по информатике. / "Информатика" № 7/2010.
11. Златопольский Д.М. Что надо уметь для решения задач С4 из ЕГЭ по информатике. / "Информатика" № 16/2011.
12. Златопольский Д.М. Задачи по программированию в ЕГЭ по информатике. Избранное. / "Информатика" № 9/2011.

№	Типовая задача	Где встречается	Примечание
1	Изменение значений элементов массива с заданными свойствами	В [3]	Новые значения записываются в другой массив
2	Расчет среднего арифметического всех элементов массива	В [7] (вариант 8)	
3	Поиск индекса элемента массива, значение которого наименее отличается от некоторого значения	Там же	Аналог задачи поиска индекса минимального элемента. Используется функция, возвращающая абсолютное значение аргумента
4	Поиск индекса первого из двух последовательных элементов массива, сумма которых максимальна	В [1]	Аналог задачи поиска индекса максимального элемента (сравниваются суммы значений пар соседних элементов)
5	Поиск индексов двух элементов массива, наименее отличающихся друг от друга	В [7] (вариант 10)	Аналог задачи поиска индекса минимального элемента. Искомых значений — 2. Используется функция, возвращающая абсолютное значение аргумента
6	Поиск индексов двух элементов массива, сумма которых минимальна	В [7] (вариант 9)	Аналог задачи поиска индекса минимального элемента (сравниваются суммы значений пар соседних элементов). Искомых значений — 2. Задача может быть также решена как задача поиска индексов двух первых минимумов (в первом толковании термина “второй минимум”)
7	Поиск индекса первого элемента массива, равного некоторому числу	В [7] (варианты 2, 4)	Искомый элемент в массиве может не быть
8	Нахождение суммы элементов массива с заданными свойствами (удовлетворяющих некоторому условию)	В [8] (вариант 1)	Элементов с заданными свойствами в массиве может не быть
9	Нахождение суммы элементов массива с заданными свойствами	В [8] (вариант 2)	
10	Нахождение количества элементов массива с заданными свойствами	В [8] (варианты 3–4)	Элементов с заданными свойствами в массиве может не быть. В задаче варианта 4 условие для подсчета — сложное (составное)
11	Поиск индекса третьего положительного элемента массива (если из массива вычеркнуть все неположительные элементы, этот элемент стоял бы в получившемся массиве на третьем месте)	В [7] (вариант 6)	Используется подсчет количества элементов массива с заданными свойствами (см. предыдущий пункт таблицы)
12	Нахождение среднего арифметического значения элементов массива с заданными свойствами	В [8] (вариант 5), в [5]	Случай, когда в массиве нет отрицательных элементов, рассматриваться не должен
13	Определение максимального значения среди тех элементов массива, которые удовлетворяют некоторому условию	В [8] (варианты 9–10)	Диапазон значений элементов массива известен
14	Определение минимального значения среди тех элементов массива, которые удовлетворяют некоторому условию	В [4], в [7] (вариант 1), в [6]	Известен диапазон значений элементов массива, а также тот факт, что соответствующие элементы в массиве имеются
15	Та же	В [7] (вариант 3)	Диапазон значений элементов массива неизвестен. Соответствующие элементы в массиве могут не быть
16	Нахождение второго максимума (в первом толковании термина)	В [7] (вариант 5)	
17	Нахождение максимального количества подряд идущих элементов массива, обладающих заданными свойствами	В [2], в [8] (варианты 6–8), в [7] (вариант 7)	В задаче варианта 7 из [7] находится также сумма элементов такого подмассива



Документ-камера

MimioView™

снимает объемные предметы

и плоские документы,

позволяя вывести на большой экран все,

что может понадобиться на уроке.

Добавьте картинки и видео на интерактивную доску в реальном времени!

Конструкция камеры позволяет легко выравнивать документы, получать стоп-кадры и записывать аннотации к видеоизображениям. Питание и передача данных осуществляются по единственному кабелю USB, что уменьшает количество лишних проводов в вашем классе! С помощью программы **MimioStudio™** можно обработать полученные статические изображения и снимать видео.

Демонстрируемые образцы подсвечиваются светодиодными лампами, обеспечивающими яркое освещение, низкую температуру и не требующими периодической замены. Гибкая регулировка обеспечивает оптимальную освещенность любых объектов, в том числе и со сложной структурой. Поставляемый с камерой адаптер микроскопа позволяет выводить на экран изображения микромира — статические и движущиеся.

Камера оснащена высококачественным объективом с 5,5-кратным оптическим и 16-кратным цифровым увеличением.

Имеется также скоростная автофокусировка. Документ-камера обеспечивает разрешение изображения 1600x1200 точек по стандарту UXGA и имеет автоматическую подстройку цветового баланса.

Документ-камера **MimioView™** является частью комплексного решения **MimioClassroom™**, интегрированной интерактивной системы, включающей также интерактивную приставку, систему автоматизации тестирования учащихся, копи-приставку и многое другое.



Узнайте больше, посетите наш сайт или позвоните:

<http://www.mimioclass.ru>

8 (800) 5555-33-0

Звонок по России бесплатный

ООО «Рене» — генеральный дистрибьютор Mimio



Mimio®

Модульные курсы предоставляют уникальную возможность:

- начать обучение в любой момент;
- выбирать удобный график освоения материалов и самостоятельно определять срок окончания изучения модуля (минимальный срок обучения – 1 месяц);
- выполнять контрольную работу в режиме онлайн;
- осваивать знания из психологии, менеджмента, экономики, которые позволят: лучше понять себя и других людей; психологические причины возникновения стрессов и различных заболеваний и сохранить свое здоровье; оптимизировать свою деятельность и др.

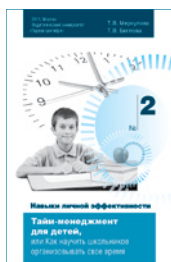
Нормативный срок освоения каждого модуля – 6 часов. Форма обучения – дистанционная. После успешного окончания модуля выдается сертификат.

Стоимость одного модульного курса – 200 руб.

ПЕРЕЧЕНЬ МОДУЛЬНЫХ КУРСОВ



Тайм-менеджмент, или Как эффективно организовать свое время.



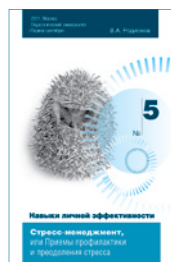
Тайм-менеджмент для детей, или Как научить школьников организовывать свое время.



Приемы конструктивного разрешения конфликтных ситуаций, или Конфликты в нашей жизни: способы решения.



Профессиональное выгорание, или Как сохранить здоровье и не «сгореть» на работе.



Стресс-менеджмент, или Приемы профилактики и преодоления стресса.



Управление имиджем, или Как создать свой стиль.



Искусство договариваться, или Как понять других людей и донести свою точку зрения.

ПОДАЙТЕ ЗАЯВКУ НА ОБУЧЕНИЕ НА САЙТЕ <http://edu.1september.ru>

Получить более подробную информацию можно на сайте, по электронной почте: module@1september.ru или по телефону (499) 249-47-82



Родственники и прочие реляции (новые задачи с базами данных)

О.Б. Богомолова,
д. п. н., учитель
информатики
и математики
ГОУ СОШ № 1360,
Восточный округ
г. Москвы

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

► На Едином государственном экзамене среди прочих заданий уже несколько лет предлагаются задачи на работу с базами данных. В предыдущие годы подобные задачи были сравнительно простыми: например, требовалось определить, сколько записей соответствует заданному запросу на выборку данных. Но в 2011 году задания на базы данных (**задание А6**) в ЕГЭ стали существенно сложнее и предполагают прослеживание реляционных связей между таблицами в многотабличной БД. “Типовая” формулировка некоторых из таких задач связана с поиском родственников заданного лица в некоей условной реляционной БД, где информация о людях и родственных связях между ними закодирована индивидуальными номерами. (Задачи, предлагавшиеся в тренировочной работе в октябре 2011 г., более просты, но смысл задания в целом тот же.)

Как выяснилось, подобные задачи вызывают у учащихся определенные трудности, причем, скорее, не на знание соб-

ственно принципов работы баз данных, а связанные с самим процессом нахождения взаимосвязей в предложенных БД. Именно поэтому авторы решили посвятить разбору таких задач отдельную статью.

Итак, начнем...

1. Демонстрационный вариант ЕГЭ 2012 г. (ФИПИ)

В фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведенных данных фамилию и инициалы бабушки Ивановой А.И.

- 1) Петренко А.И.
- 2) Черных И.А.
- 3) Цейс Т.Н.
- 4) Петренко Н.Н.

Таблица 1

ID	Фамилия_И.О.	Пол
71	Иванов Т.М.	М
85	Петренко И.Т.	М
13	Черных И.А.	Ж
42	Петренко А.И.	Ж
23	Иванова А.И.	Ж
96	Петренко Н.Н.	Ж
82	Черных А.Н.	М
95	Цейс Т.Н.	Ж
10	Цейс Н.А.	М
...		

Таблица 2

ID_родителя	ID_ребенка
23	71
13	23
85	23
82	13
95	13
85	42
82	10
95	10
...	...

Решение

Первое, на что здесь нужно обратить внимание: каждому человеку, чьи Ф.И.О. и пол отражены в этой базе данных в табл. 1, присвоен конкретный индивидуальный номер.

Вторая особенность: табл. 2 определяет родственные связи между людьми (закодированными их индивидуальными номерами) по принципу “родитель — ребенок”.

Начинаем решение.

Исходное лицо: Иванова А.И.

1) Ищем запись о ней в табл. 1 (проверяя, что значение поля “Пол” для нее должно быть равно “Ж”) и определяем ее индивидуальный номер: **23**.

Таблица 1

ID	Фамилия_И.О.	Пол
71	Иванов Т.М.	М
85	Петренко И.Т.	М
13	Черных И.А.	Ж
42	Петренко А.И.	Ж
23	Иванова А.И.	Ж
96	Петренко Н.Н.	Ж
82	Черных А.Н.	М
95	Цейс Т.Н.	Ж
10	Цейс Н.А.	М
...

Таблица 2

ID_родителя	ID_ребенка
23	71
13	23
85	23
82	13
95	13
85	42
82	10
95	10
...	...

2) Определяем по табл. 2, кто является родителем человека с найденным индивидуальным номером 23. Для этого ищем в табл. 2 все строки с номером 23 в поле “ID_ребенка” и определяем, какие индивидуальные номера соответствуют в найденных записях полю “ID_родителя”. Это номера **13** и **85**: очевидно, мать и отец Ивановой А.И. (Вернувшись к табл. 1, по найденным номерам можно опреде-

лить по значению поля “Пол”, кто есть кто, хотя для решения данной задачи это не требуется.)

Таблица 1

ID	Фамилия_И.О.	Пол
71	Иванов Т.М.	М
85	Петренко И.Т.	М
13	Черных И.А.	Ж
42	Петренко А.И.	Ж
23	Иванова А.И.	Ж
96	Петренко Н.Н.	Ж
82	Черных А.Н.	М
95	Цейс Т.Н.	Ж
10	Цейс Н.А.	М
...

Таблица 2

ID_родителя	ID_ребенка
23	71
13	23
85	23
82	13
95	13
85	42
82	10
95	10
...	...

3) Мы определили родителей Ивановой А.И., но по условию задачи требуется найти, кто является ее бабушкой. Бабушка — это (в контексте рассматриваемого задания) мать одного из родителей Ивановой А.И., поэтому вышеописанную операцию поиска ID_родителей по ID_детей надо повторить для найденных на предыдущем шаге матери и отца Ивановой А.И.

• Ищем в табл. 2 в поле “ID_ребенка” индивидуальный номер 85. Поскольку такое значение в указанном поле отсутствует, поиск по данной ветви генеалогического древа можно прекратить.

• Ищем в табл. 2 в поле “ID_ребенка” индивидуальный номер 13 и определяем все значения индивидуальных номеров в поле “ID_родителя”, соответствующих указанному значению поля “ID_ребенка”. Это номера **82** и **95** (мать и отец матери Ивановой А.И., т.е. ее дед и искомая бабушка).

Таблица 1

ID	Фамилия_И.О.	Пол
71	Иванов Т.М.	М
85	Петренко И.Т.	М
13	Черных И.А.	Ж
42	Петренко А.И.	Ж
23	Иванова А.И.	Ж
96	Петренко Н.Н.	Ж
82	Черных А.Н.	М
95	Цейс Т.Н.	Ж
10	Цейс Н.А.	М
...

Таблица 2

ID_родителя	ID_ребенка
23	71
13	23
85	23
82	13
95	13
85	42
82	10
95	10
...	...

4) Найдя ID деда и бабушки, возвращаемся к табл. 1, чтобы по этим номерам и значениям поля “Пол” определить, кто из них кто.

Таблица 1

ID	Фамилия_И.О.	Пол
71	Иванов Т.М.	М
85	Петренко И.Т.	М
13	Черных И.А.	Ж
42	Петренко А.И.	Ж
23	Иванова А.И.	Ж
96	Петренко Н.Н.	Ж
82	Черных А.Н.	М
95	Цейс Т.Н.	Ж
10	Цейс Н.А.	М

Таблица 2

ID_родителя	ID_ребенка
23	71
13	23
85	23
82	13
95	13
85	42
82	10
95	10
...	...

Очевидно, что бабушкой Ивановой А.И. является Цейс Т.Н.

Ответ: вариант 3.

2. Тренировочная работа № 1 по информатике, 11 октября 2011 г. (МИОО). Вариант 1

Дан фрагмент базы данных некоторого образовательного учреждения. Все объекты в этой базе имеют свой идентификационный код (4-значное число в 16-ричной системе счисления). Определите на основании приведенных данных номер группы, в которой учится Смирнова Ю.

Таблица 2

№ группы	ID
Группа 1	3D95
Группа 2	67BA
Группа 3	3668
Группа 4	5D6B
...	...

Таблица 1

Учащийся	ID
Бердыев А.	B8FE
Зинатулина А.	2969
Круглова С.	F719
Кузнецов Ю.	34F4
Лебедева А.	9829
Лобчиков В.	4BF1
Морозова А.	0118
Мохначева А.	BFCE
Петрова А.	E641
Смирнова Ю.	156D
Тамкова В.	DDC8
Храповский М.	F46C
Черткова Д.	F045
Шопша Н.	E8AC
Шубина Ж.	96F0
Щербакова Е.	E63D
...	...

Таблица 3

ID_группы	ID_учащегося
3668	F46C
3668	96F0
3668	E8AC
3D95	F719
3D95	34F4
3D95	BFCE
3D95	2969
3D95	DDC8
5D6B	B8FE
5D6B	156D
5D6B	E641
5D6B	0118
67BA	F045
67BA	4BF1
67BA	9829
67BA	E63D

- 1) 1 2) 2 3) 3 4) 4

Решение

Хотя здесь речь идет не о родственниках, принцип решения тот же.

Исходное лицо: Смирнова Ю.

1) По заданной фамилии и имени студентки определяем в табл. 1 ее уникальный идентификационный номер: **156D**.

2) Определив ID учащегося, ищем его в соответствующем поле табл. 3, отражающей взаимосвязь кодов учащихся и учебных групп (обратите внимание: одному коду группы может соответствовать много кодов учащихся, но каждому коду учащегося соответствует только один код группы; в предыдущей задаче одному коду ребенка соответствовали один или два кода его родителей). Искомый код группы равен **5D6B**.

3) Обратившись к табл. 2, где сопоставлены коды групп и их порядковые номера, определяем искомый номер группы. Он равен **4**.

Таблица 1

Учащийся	ID
Бердыев А.	B8FE
Зинатуллина А.	2969
Круглова С.	F719
Кузнецов Ю.	34F4
Лебедева А.	9829
Лобчиков В.	4BF1
Морозова А.	0118
Мохначева А.	BFCE
Петрова А.	E641
Смирнова Ю.	156D
Тамкова В.	DDC8
Храповский М.	F46C
Черткова Д.	F045
Шопша Н.	E8AC
Шубина Ж.	96F0
Щербакова Е.	E63D
...	...

Таблица 2

№ группы	ID
Группа 1	3D95
Группа 2	67BA
Группа 3	3668
Группа 4	5D6B
...	...

Таблица 3

ID_группы	ID_учащегося
3668	F46C
3668	96F0
3668	E8AC
3D95	F719
3D95	34F4
3D95	BFCE
3D95	2969
3D95	DDC8
5D6B	B8FE
5D6B	156D
5D6B	E641
5D6B	0118
67BA	F045
67BA	4BF1
67BA	9829
67BA	E63D

Ответ: вариант 4.

3. Тренировочная работа № 1 по информатике, 11 октября 2011 г. (МИОО). Вариант 2

Дан фрагмент базы данных некоторого образовательного учреждения. Все объекты в этой базе имеют свой идентификационный код (4-значное число в 16-ричной системе счисления). Определите на основании приведенных данных, какую отметку по физике имеет Кузнецов Ю.

Таблица 1

Учащийся	ID
Круглова С.	F719
Кузнецов Ю.	34F4
Лебедева А.	9829
Лобчиков В.	4BF1
Морозова А.	0118
Мохначева А.	BFCE
Петрова А.	E641
Смирнова Ю.	156D
...	...

Таблица 2

Предмет	ID
Русский язык	2969
Математика	DDC8
Физика	B8FE
Химия	156D
...	...

Таблица 3

ID_учащегося	ID_предмета	Отметка
9829	B8FE	5
BFCE	B8FE	3
F719	B8FE	4
E641	B8FE	5
E641	156D	5
4BF1	DDC8	2
0118	DDC8	4
E641	2969	5
BFCE	DDC8	3
34F4	B8FE	3
4BF1	2969	2
156D	2969	2
4BF1	B8FE	2
9829	2969	5
34F4	156D	3
4BF1	156D	2
156D	DDC8	2
0118	2969	4
156D	B8FE	2
F719	156D	4
F719	DDC8	4
BFCE	156D	3
0118	B8FE	4
E641	DDC8	5
156D	156D	2
0118	156D	4
9829	156D	5
F719	2969	4
BFCE	2969	3
9829	DDC8	5
34F4	2969	3
34F4	DDC8	3
...

- 1) 2 2) 3 3) 4 4) 5

Решение

Исходное лицо: Кузнецов Ю.

1) По табл. 1 определяем уникальный идентификационный номер Кузнецова Ю.: это код **34F4**.

2) Аналогично, по табл. 2 определяем идентификационный номер предмета "физика". Это код **B8FE**.

3) Зная коды учащегося и предмета, ищем в табл. 3 значение оценки. Обратите внимание: в этой таблице одному коду учащегося может соответствовать несколько кодов учебных предметов и наоборот, одному коду учебного предмета может соответствовать несколько кодов учащихся, но *каждое сочетание кода учащегося и кода*

учебного предмета является уникальным. Очевидно, ранее определенным кодам учащегося и предмета соответствует запись, содержащая в поле “Отметка” значение 3.

Ответ: вариант 2.

(Не запутайтесь! Если в предыдущей задаче номер варианта ответа совпадал с соответствующим числовым значением — искомым номером учебной группы, то в данном случае такого совпадения уже нет!)

Таблица 1

Учащийся	ID
Круглова С.	F719
Кузнецов Ю.	34F4
Лебедева А.	9829
Лобчиков В.	4BF1
Морозова А.	0118
Мохначева А.	BFCE
Петрова А.	E641
Смирнова Ю.	156D
...	...

Таблица 2

Предмет	ID
Русский язык	2969
Математика	DDC8
Физика	B8FE
Химия	156D
...	...

Таблица 3

ID_учащегося	ID_предмета	Отметка
9829	B8FE	5
BFCE	B8FE	3
F719	B8FE	4
E641	B8FE	5
E641	156D	5
4BF1	DDC8	2
0118	DDC8	4
E641	2969	5
BFCE	DDC8	3
34F4	B8FE	3
4BF1	2969	2
156D	2969	2
4BF1	B8FE	2
9829	2969	5
34F4	156D	3
4BF1	156D	2
156D	DDC8	2
0118	2969	4
156D	B8FE	2
F719	156D	4
F719	DDC8	4
BFCE	156D	3
0118	B8FE	4
E641	DDC8	5
156D	156D	2
0118	156D	4
9829	156D	5
F719	2969	4
BFCE	2969	3
9829	DDC8	5
34F4	2969	3
34F4	DDC8	3
...

4. Диагностическая работа № 1 по информатике, 19 декабря 2011 г. (МИОО). Вариант 1

В фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведенных данных фамилию и инициалы племянника Черных Н.И.

Примечание: племянник — сын сестры или брата.

Таблица 1

ID	Фамилия_И.О.	Пол
85	Гуревич И.Т.	М
82	Гуревич А.И.	М
42	Цейс А.Т.	Ж
71	Петров Т.М.	М
23	Петров А.Т.	М
13	Цейс И.И.	Ж
95	Черных Т.Н.	Ж
10	Черных Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
95	82
85	13
71	42
85	82
13	42
71	23
13	23
95	13
85	10
...	...

1) Петров А.Т. 2) Петров Т.М. 3) Гуревич А.И. 4) Гуревич И.Т.

Решение

Эта задача аналогична по формулировке и принципу решения ранее рассмотренной задаче из демонстрационного варианта ЕГЭ от ФИПИ. Однако она заметно сложнее, поскольку ее решение не так “прямолинейно”.

Исходное лицо: Черных Н.И.

1) По табл. 1 определяем уникальный идентификационный номер Черных Н.И. Он равен 10.

Таблица 1

ID	Фамилия_И.О.	Пол
85	Гуревич И.Т.	М
82	Гуревич А.И.	М
42	Цейс А.Т.	Ж
71	Петров Т.М.	М
23	Петров А.Т.	М
13	Цейс И.И.	Ж
95	Черных Т.Н.	Ж
10	Черных Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
95	82
85	13
71	42
85	82
13	42
71	23
13	23
95	13
85	10
...	...

2) Нам нужен племянник Черных Н.И. Но в табл. 2 отражены родственные связи только типа “родитель — ребенок”, а нам нужно выявить родственную связь “сын сестры/брата”. Как это сделать?

Нужно догадаться, что сестра или брат являются детьми тех же самых родителей, что и интересующего нас лица. Поэтому, зная ID исходного человека (10), мы ищем в табл. 2 все строки, для которых в поле “ID_ребенка” записан код 10. Такая запись в табл. 2 одна и содержит в поле “ID_родителя” идентификационный номер 85. (Вернувшись к табл. 1, можно по этому идентификатору определить, что речь идет об отце Черных Н.И. — Гуревиче И.Т.)

Таблица 1

ID	Фамилия_И.О.	Пол
85	Гуревич И.Т.	М
82	Гуревич А.И.	М
42	Цейс А.Т.	Ж
71	Петров Т.М.	М
23	Петров А.Т.	М
13	Цейс И.И.	Ж
95	Черных Т.Н.	Ж
10	Черных Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
95	82
85	13
71	42
85	82
13	42
71	23
13	23
95	13
85	10
...	...

3) Теперь нам нужно (опять-таки по табл. 2) определить для найденного ID_родителя идентификационные номера *всех* его детей. Для этого мы ищем в табл. 2 все записи (строки), где в поле “ID_родителя” записан код 85. Соответствующие ID детей равны: 13, 82 и 10.

Таблица 1

ID	Фамилия_И.О.	Пол
85	Гуревич И.Т.	М
82	Гуревич А.И.	М
42	Цейс А.Т.	Ж
71	Петров Т.М.	М
23	Петров А.Т.	М
13	Цейс И.И.	Ж
95	Черных Т.Н.	Ж
10	Черных Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
95	82
85	13
71	42
85	82
13	42
71	23
13	23
95	13
85	10
...	...

4) Вернемся к табл. 1 и для найденных идентификаторов найдем информацию о людях (Ф.И.О. и пол):

- код 13 — Цейс И.И. (пол “Ж”);
- код 82 — Гуревич А.И. (пол “М”);
- код 10 — Черных Н.И. (пол “М”).

Из найденных людей Черных Н.И. — исходное лицо, и его мы из рассмотрения исключаем.

Таблица 1

ID	Фамилия_И.О.	Пол
85	Гуревич И.Т.	М
82	Гуревич А.И.	М
42	Цейс А.Т.	Ж
71	Петров Т.М.	М
23	Петров А.Т.	М
13	Цейс И.И.	Ж
95	Черных Т.Н.	Ж
10	Черных Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
95	82
85	13
71	42
85	82
13	42
71	23
13	23
95	13
85	10
...	...

5) Итак, после проделанных операций мы определили брата и сестру Черных Н.И. Теперь нужно найти племянника Черных Н.И., зная, что это сын его брата или сестры. Для этого вновь обращаемся к табл. 2 и ищем в ней записи (строки), где в поле "ID_родителя" записано значение 13 или 82. (Нетрудно видеть, что записи для кода 82 в таблице отсутствуют, что облегчает нам задачу.) Таких записей две, в них в поле "ID_ребенка" записаны значения кодов 42 и 23.

Таблица 1

ID	Фамилия_И.О.	Пол
85	Гуревич И.Т.	М
82	Гуревич А.И.	М
42	Цейс А.Т.	Ж
71	Петров Т.М.	М
23	Петров А.Т.	М
13	Цейс И.И.	Ж
95	Черных Т.Н.	Ж
10	Черных Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
95	82
85	13
71	42
85	82
13	42
71	23
13	23
95	13
85	10
...	...

6) Наконец, снова вернувшись к табл. 1, для найденных кодов определяем Ф.И.О. и пол соответствующих лиц:

- код 42 — Цейс А.Т. (пол "Ж");
- код 23 — Петров А.Т. (пол "М").

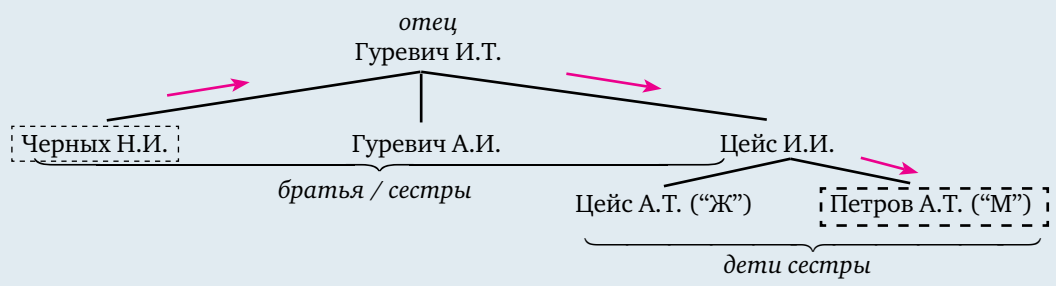
Таблица 1

ID	Фамилия_И.О.	Пол
85	Гуревич И.Т.	М
82	Гуревич А.И.	М
42	Цейс А.Т.	Ж
71	Петров Т.М.	М
23	Петров А.Т.	М
13	Цейс И.И.	Ж
95	Черных Т.Н.	Ж
10	Черных Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
95	82
85	13
71	42
85	82
13	42
71	23
13	23
95	13
85	10
...	...

Из указанных лиц Цейс А.Т. нам не подходит, поскольку по условию задачи нужно было определить племянника, а не племянницу. Поэтому единственный человек, подходящий в качестве решения задачи, — это Петров А.Т.
 Ответ: вариант 1.



Методический совет. Для наглядности мы изобразили “генеалогическое древо” всех рассмотренных в задаче лиц с указанием стрелками пути поиска племянника.

Такую схему рекомендуется продемонстрировать учащимся при помощи медиапроектора в ходе разбора решения задачи, однако уже после того, как они попытаются найти решение самостоятельно, либо после словесной подсказки учителя.

5. Диагностическая работа № 1 по информатике, 19 декабря 2011 г. (МИОО). Вариант 2

В фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведенных данных фамилию и инициалы племянника Симоняна Н.И.

Примечание: племянник — сын сестры или брата.

Таблица 1

ID	Фамилия_И.О.	Пол
86	Седых И.Т.	М
83	Седых А.И.	М
50	Силис А.Т.	Ж
79	Симонов Т.М.	М
23	Симонов А.Т.	М
13	Силис И.И.	Ж
98	Симонян Т.Н.	Ж
11	Симонян Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
98	83
86	13
79	50
86	83
13	50
79	23
13	23
98	13
86	11
...	...

- 1) Седых А.И. 2) Седых И.Т. 3) Симонов А.Т. 4) Симонов Т.М.

Решить эту задачу и изобразить соответствующее ей “генеалогическое древо” персонажей предлагаем самим читателям.

Решение задачи 5

Исходное лицо: Симонян Н.И.

- 1) По табл. 1 определяем уникальный идентификационный номер Симоняна Н.И. Он равен **11**.

Таблица 1

ID	Фамилия_И.О.	Пол
86	Седых И.Т.	М
83	Седых А.И.	М
50	Силис А.Т.	Ж
79	Симонов Т.М.	М
23	Симонов А.Т.	М
13	Силис И.И.	Ж
98	Симонян Т.Н.	Ж
11	Симонян Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
98	83
86	13
79	50
86	83
13	50
79	23
13	23
98	13
86	11
...	...

2) Для поиска сестры или брата Симоняна Н.И. в табл. 2 ищем записи о его родителях (где в поле “ID_ребенка” записан код 11). Такая запись в табл. 2 одна и содержит в поле “ID_родителя” идентификационный номер **86**. (Вернувшись к табл. 1, можно по этому идентификатору определить, что речь идет об отце Симоняна Н.И. — Седых И.Т.)

Таблица 1

ID	Фамилия_И.О.	Пол
86	Седых И.Т.	М
83	Седых А.И.	М
50	Силис А.Т.	Ж
79	Симонов Т.М.	М
23	Симонов А.Т.	М
13	Силис И.И.	Ж
98	Симонян Т.Н.	Ж
11	Симонян Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
98	83
86	13
79	50
86	83
13	50
79	23
13	23
98	13
86	11
...	...



3) Теперь (опять по табл. 2) определяем для найденного ID родителя идентификационные номера всех его детей. Для этого мы ищем в табл. 2 все записи (строки), где в поле “ID_родителя” записан код 86. Соответствующие ID детей равны: **13, 82 и 10**.

Таблица 1

ID	Фамилия_И.О.	Пол
86	Седых И.Т.	М
83	Седых А.И.	М
50	Силис А.Т.	Ж
79	Симонов Т.М.	М
23	Симонов А.Т.	М
13	Силис И.И.	Ж
98	Симонян Т.Н.	Ж
11	Симонян Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
98	83
86	13
79	50
86	83
13	50
79	23
13	23
98	13
86	11
...	...

4) Вернемся к табл. 1 и для найденных идентификаторов найдем информацию о людях (Ф.И.О. и пол), исключая из рассмотрения исходное лицо — Симоняна Н.И.:

- код 13 — Силис И.И. (пол “Ж”);
- код 83 — Седых А.И. (пол “М”).

Таблица 1

ID	Фамилия_И.О.	Пол
86	Седых И.Т.	М
83	Седых А.И.	М
50	Силис А.Т.	Ж
79	Симонов Т.М.	М
23	Симонов А.Т.	М
13	Силис И.И.	Ж
98	Симонян Т.Н.	Ж
11	Симонян Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
98	83
86	13
79	50
86	83
13	50
79	23
13	23
98	13
86	11
...	...

5) Для поиска племянника Симоняна Н.И., зная, что это сын его брата или сестры, обращаемся к табл. 2 и ищем в ней записи, где в поле “ID_родителя” записано значение 13 (поскольку записи для кода 83 в таблице отсутствуют). Таких записей две, в них в поле “ID_ребенка” записаны значения кодов **50 и 23**.

Таблица 1

ID	Фамилия_И.О.	Пол
86	Седых И.Т.	М
83	Седых А.И.	М
50	Силис А.Т.	Ж
79	Симонов Т.М.	М
23	Симонов А.Т.	М
13	Силис И.И.	Ж
98	Симонян Т.Н.	Ж
11	Симонян Н.И.	М
...

Таблица 2

ID_родителя	ID_ребенка
98	83
86	13
79	50
86	83
13	50
79	23
13	23
98	13
86	11
...	...

6) Вернувшись к табл. 1, для найденных кодов определяем Ф.И.О. и пол соответствующих лиц:

- код 50 — Силис А.Т. (пол “Ж”);
- код 23 — Симонов А.Т. (пол “М”).

Таблица 1

ID	Фамилия_И.О.	Пол
86	Седых И.Т.	М
83	Седых А.И.	М
50	Силис А.Т.	Ж
79	Симонов Т.М.	М
23	Симонов А.Т.	М
13	Силис И.И.	Ж
98	Симомян Т.Н.	Ж
11	Симомян Н.И.	М
	...	

Таблица 2

ID_родителя	ID_ребенка
98	83
86	13
79	50
86	83
13	50
79	23
13	23
98	13
86	11
...	...

Из указанных лиц Силис А.Т. нам не подходит (это — племянница). Поэтому единственный человек, подходящий в качестве решения задачи, — это Симонов А.Т.

Ответ: вариант 3.

ПРИГЛАШАЕМ К УЧАСТИЮ В КОНКУРСЕ

► Федеральный институт педагогических измерений (ФИПИ) приглашает читателей “Информатики” принять участие в конкурсе на разработку перспективных моделей экзаменационных заданий для проведения ЕГЭ и итоговой аттестации за курс основной школы (ГИА-9).

Объявляя этот конкурс, ФИПИ стремится начать серьезное и содержательное обсуждение путей совершенствования экзамена по информатике и ИКТ, причем речь совсем не идет о немедленной смене “правил игры”, а скорее о плавном движении в определенных направлениях.

Какие же направления кажутся наиболее актуальными?

Не секрет, что в нынешней модели ЕГЭ по информатике существенная часть заданий так или иначе связана с такими разделами курса, как логика, кодирование информации, алгоритмы, языки программирования.

С другой стороны, Федеральный компонент действующего государственного стандарта полного (среднего) общего образования по информатике и ИКТ включает требования, которые представляются более чем актуальными с точки зрения продолжения образования и будущей профессиональной деятельности по тем специальностям, на которые можно поступить, сдавая ЕГЭ по информатике и ИКТ.

Так, например, выпускник должен “уметь выделять информационный аспект в деятельности человека, выделять информаци-

онное взаимодействие в простейших социальных, биологических и технических системах, уметь выполнять требования техники безопасности, гигиены, эргономики и ресурсосбережения при работе со средствами информатизации”. Важнейшими представляются и требования “использовать приобретенные знания и умения для поиска и отбора информации, в частности, связанной с личными познавательными интересами, самообразованием и профессиональной ориентацией, для соблюдения требований информационной безопасности, информационной этики и права”.

Хотелось бы обратить внимание читателей и на то, что в качестве конкурсной заявки могут быть предложены модели экзаменационных заданий, использование которых возможно только на компьютере. Это направление конкурса напрямую связано с проработкой в настоящий момент вопроса о проведении ЕГЭ по информатике и ИКТ в компьютерной форме.

В заключение хочется пожелать читателям журнала успехов и пригласить к совместному творчеству.

О конкурсе

Заявки принимаются с 15 февраля до 30 марта 2012 г. Подведение итогов — 16 апреля 2012 г. Более подробная информация имеется на странице <http://www.fipi.ru/view/sections/68/docs/600.html> сайта ФИПИ.

**Сергей Владимирович Станченко,
заместитель директора ФИПИ**



ИСТОРИЯ ИНФОРМАТИКИ

► **От редакции.** В 2011 году в издательстве “ЭНАС” вышла книга В.В. Шилова “Удивительная история информатики и автоматки (О чем умолчали учебники)”. В ней собраны интересные сведения и малоизвестные факты из истории информатики, автоматки, робототехники, информационных и коммуникационных технологий. Предлагаем читателям ознакомиться с фрагментом из книги, подготовленным ее автором (см. также статью В.В. Шилова «Это древнее слово — “компьютер”» в “Информатике” № 15/2011).

Юбилей термина “информатика”

В начале 1960-х годов американский программист Уолтер Бауэр с несколькими коллегами решили создать собственную компанию по разработке программного обеспечения. Разумеется, немедленно встал вопрос о ее названии. Поскольку программы работают с *данными* (по-английски — *data*), в первую очередь вспомнилось слово *Datamatics* (в том, что название должно заканчиваться греческим суффиксом *-atics*, имеющим значение “наука о...”, никто не сомневался). Однако зарегистрировать его Бауэру не удалось, поскольку оно уже было использовано фирмами *Raytheon* и *Honeywell*, выпустившими компьютер *Datamatic 1000*.

Разочарование было недолгим, и следующий вариант был принят с восторгом. Слово *informatics* понравилось всем, и название без промедления было зарегистрировано. Правда, о том, кто же именно предложил его, компаньоны спорят до сих пор.

В это же время во Франции возникла другая программистская фирма — *Societe pour L'Informatique et Applique (SIA)*, основанная одним из первых французских программистов Филиппом Дрейфусом. Очень быстро слово *L'Informatique* обрело популярность и в значении “наука об обработке информации с помощью электронных вычислительных машин”, получив одобрение Французской академии, было официально включено в словарь современного французского языка. В формах *informatik*, *informatica*, *информатика* и других оно вскоре вошло во все европейские языки.

А вот в США дело обстояло совершенно по-иному. Фирма *Informatics* на протяжении многих лет успешно отстаивала свои права на зарегистрированную торговую марку. Однажды за разрешением использовать слово к Бауэру обратились представители крупнейшего американского профессионального общества *ACM (Association for Computing Machinery)*, которое решило изменить свое название на *Society for Informatics* (“Общество информатики”). Предложение было очень лестным, но, посоветовавшись с юристами, Бауэр отклонил его — название фирмы является неотъемлемой частью ее активов, и его изменение грозило принести акционерам значительные убытки.

Свое старое название *ACM* сохранило до сегодняшнего дня, а наука, изучаемая нами как информатика, в США носит название *Computer Science*.

Между прочим, когда несколько лет спустя Бауэр и Дрейфус встретились в Париже и разговорились, то выяснили, что они оба придумали новое слово одновременно — в марте 1962 года.

Так что можно сказать, что слову “информатика” — 50 лет!

Уважаемые коллеги!

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

Заявку на диплом просьба прислать в апреле–мае 2011 года в адрес редакции (vmi@1september.ru). В школы, не имеющие доступа в Интернет, могут быть направлены дипломы учащимся согласно заявке обычной почтой.

Задача, которую вы решаете, может быть очень скромной, но если она бросает вызов вашей любознательности и если вы решаете ее собственными силами, то вы сможете испытать вездущее к открытию напряжение ума и насладиться радостью победы.

Джордж Пойя

Ответы, решения, разъяснения к заданиям, опубликованным в разделе “В мир информатики” № 169 (“Информатика” № 15/2011)

Задача “Вкусные ломтики”

Решение

Если пронумеровать ломтики, то задача поджаривания четырех ломтиков за минимально возможное время решается за четыре шага (этапа), представленных в табл. 1, а пяти ломтиков — за пять шагов, показанных в табл. 2.

Таблица 1

№ этапа	На сковородке находятся:		Продолжительность поджаривания, сек.	Готов ломтик
	Ломтик	Сторона		
1	№ 1	Первая	30	
	№ 2	Первая		
2	№ 1	Вторая	30	№ 1
	№ 2	Вторая		№ 2
3	№ 3	Первая	30	
	№ 4	Первая		
4	№ 3	Вторая	30	№ 3
	№ 4	Вторая		№ 4
Всего			2 минуты	

Таблица 2

№ этапа	На сковородке находятся:		Продолжительность поджаривания, сек.	Готов ломтик
	Ломтик	Сторона		
1	№ 1	Первая	30	
	№ 2	Первая		
2	№ 1	Вторая	30	№ 1
	№ 3	Первая		
3	№ 2	Вторая	30	№ 2
	№ 3	Вторая		№ 3
4	№ 3	Первая	30	
	№ 4	Первая		
5	№ 3	Вторая	30	№ 4
	№ 4	Вторая		№ 5
Всего			2,5 минуты	

Ответы представили:

— Александрова Алена, Антошко Михаил и Горшков Алексей, Республика Башкортостан, г. Стерлитамак, школа № 24, учитель **Орлова Е.В.**;

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

— Волков Владимир, Демьянова Елена и Хомякова Анна, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Голик Екатерина, Головченко Ксения, Синицын Никита, Телегин Дмитрий и Юматова Светлана, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Горьков Игорь, Грибанов Владлен, Дукач Светлана, Искандарова Лилия и Соболев Иван, г. Лесосибирск Красноярского края, поселок Стрелка, школа № 8 им. Константина Филиппова, учитель **Лопатин М.А.**;

— Корольчук Сергей, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Решетников Виталий, Вадьковская средняя школа, Брянская обл., Погарский р-н, учитель **Цыганкова И.Ю.**;

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

— Семенов Андрей и Турков Андрей, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Шарапова Елизавета, г. Пенза, школа № 512, учитель **Гаврилова М.И.**

В ряде ответов вторая задача решалась за 3 минуты.

Ребусы, посвященные Году космонавтики. Часть 2

Ответы. Ребус № 1: Кубасов. Ребус № 2: Терешкова. Ребус № 3: Николаев. Ребус № 4: Титов Герман. Ребус № 5: Бударин. Ребус № 6: Леонов. Ребус № 7: Быковский. Ребус № 8: Беляев. Ребус № 9: Гагарин.

Ответы прислали:

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

— Бондарев Вадим и Корчагин Александр, основная школа поселка Михинский, Воронежская обл., Таловский р-н, учитель **Удалова А.А.**;

— Васильева Наталья, Закуленкова Виолетта, Захарова Юлия, Иванова Ирина, Козлова Наталья, Олешева Дарья и Топорова Анастасия, Смоленская обл., г. Демидов, школа № 1, учитель **Кордина Н.Е.**;

— Горьков Игорь, Грибанов Владлен, Дукач Светлана, Искандарова Лилия и Соболев Иван, г. Лесосибирск Красноярского края, поселок Стрелка, школа № 8 им. Константина Филиппова, учитель **Лопатин М.А.**;

— Дильмухаметов Рустам и Жукова Елизавета, Республика Башкортостан, г. Стерлитамак, школа № 24, учитель **Орлова Е.В.**;

— Кашбиев Ильфат, Республика Татарстан, Актанышский р-н, село Актаныш, средняя школа № 2, учитель **Гилязова Г.М.**;

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

— Корольчук Сергей, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Ломакина Анастасия, средняя школа рабочего поселка Пинеровка, Саратовская обл., Балашовский р-н, учитель **Пичугин В.В.**;

— Решетников Виталий и Решетникова Наталья, Вадьковская средняя школа, Брянская обл., Погарский р-н, учитель **Цыганкова И.Ю.**;

— Семенов Андрей и Турков Андрей, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

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

— Шарапова Елизавета, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Шейкин Александр, средняя школа села Ириновка, Новобураский р-н Саратовской обл., учитель **Брунов А.С.**

Лучшим признан ответ Виталия Решетникова из Вадьковской средней школы, правильно решившего все ребусы и приведшего фотографии космонавтов и краткую биографическую справку о каждом из них. Виталий будет награжден дипломом.

Задача “Три лампочки”

Решение

Задача решается за один переход из комнаты в комнату. Надо включить одну клавишу, подождать некоторое время (скажем, 5 минут), выключить ее и включить другую, после чего идти в комнату с лампочками. Одна клавиша в этот момент включена — соответствующая лампочка горит, из двух других та лампочка, что недавно горела, будет заметно горячее той, которая все это время была выключена. Той лампочке, что горячее, соответствует клавиша, которая была включена и потом выключена, оставшейся — клавиша, которую совсем не трогали.

Ответы представили:

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

— Горьков Игорь, Грибанов Владлен, Дукач Светлана, Искандарова Лилия и Соболев Иван, г. Лесосибирск Красноярского края, поселок Стрелка, школа № 8 им. Константина Филиппова, учитель **Лопатин М.А.**;

— Корольчук Сергей, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Решетников Виталий, Вадьковская средняя школа, Брянская обл., Погарский р-н, учитель **Цыганкова И.Ю.**;

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

Задача “Торговка и пирожки”

Решение

Можно, конечно, составить и решить уравнение, но такой вариант — объемный и трудоемкий. Лучше решать задачу, идя, так сказать, “с конца”. Так как после того, как воры украли 7 пирожков и полостатка, торговка принесла на рынок 1 пирожок, то до этого у нее было $1 \times 2 + 7 = 9$ пирожков. Продолжая аналогичные рассуждения и расчеты, можно получить число пирожков, которое было у торговки первоначально, — 79.

Ответы представили:

— Антипов Анатолий, средняя школа поселка Осинка, Алтайский край, учитель **Евдокимова А.И.**;

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

— Корольчук Сергей, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Зубкова Арина, Кадура Светлана, Деревянко Оксана, Ставропольский край, Кочубеевский р-н, станция Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

— Шарапова Елизавета, г. Пенза, школа № 512, учитель **Гаврилова М.И.**

Задача “Расписание уроков”

Ответ

Возможны два варианта расписания: МФБИ и ИМФБ.

Правильные ответы прислали:

— Алексеева Алена, Голик Екатерина, Головченко Ксения, Изотова Александра и Телегин Дмитрий, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Андрющенко Александр, Зубкова Арина, Кадура Светлана, Деревянко Оксана и Свистунов Николай, Ставропольский край, Кочубеевский р-н, станция Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

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

— Волков Владимир, Демьянова Елена и Хомякова Анна, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Горьков Игорь, Грибанов Владлен, Дукач Светлана, Искандарова Лилия и Соболев Иван, г. Лесосибирск Красноярского края, поселок Стрелка, школа № 8 им. Константина Филиппова, учитель **Лопатин М.А.**;

— Довгань Алексей, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Решетников Виталий, Вадьковская средняя школа, Брянская обл., Погарский р-н, учитель **Цыганкова И.Ю.**;

— Согомонян Серине, Воронежская обл., поселок Каменка, средняя школа № 1 им. Героя Советского Союза В.П. Захарченко, учитель **Старикова М.Е.**

Задание “Составить слово”

Напомним, что необходимо было из букв заданных слов составить термин или фамилию ученого, связанные с информатикой и компьютерами.

Ответы

- А. Адрес, адаптер, архитектура.
- Б. Бейсик, блокнот, браузер.
- В. Верстка, видеосигнал, видеотерминал.
- Г. Гипертекст, графика, графопостроитель.
- Д. Дискета, деинсталлятор, диалог.
- Е. Емкость, Ершов, если.
- Ж. Жесткий, жесткость, жидкокристаллический.
- З. Заголовок, заливка, заставка.
- И. Импорт, исполнитель.
- К. Кнопка, клавиатура, колонтитул.
- Л. Ластик, линейка, логика.
- М. Макрос, макрокоманда, модель.
- Н. Навигатор, настройка, носитель.
- О. Обработка, оператор, очистка.
- П. Палитра, пиктограмма, протокол.
- Р. Робот, растеризация, редактор.
- С. Сканер, сортировка, синтаксис.
- Т. Таймер, текстура, терминал.
- У. Упаковка, управление, установка.
- Ф. Формат, формула, Фортран.
- Х. Хакер, характеристика, Холлерит.
- Ц. Целостность, целостность, центральный.
- Ч. Черепашка, частное, частота.
- Ш. Шифратор, штрихкод, штырек.
- Щ. Щелкать, щелкнуть, щелчок.
- Э. Экран, эмулятор, эхоконтроль.
- Ю. Юзер, “Юнивак”, “Юникс”.
- Я. Ядро, яркость, ярлык.

Ответы представили:

— Андриющенко Александр, Деревянко Оксана, Зубкова Арина, Кадура Светлана, Куценко Евгения, Парамонова Анастасия и Свистунов Николай, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

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

— Жданова Нелли, Сухоруков Антанас и Файзулина Лилия, Республика Башкортостан, г. Стерлитамак, школа № 24, учитель **Орлова Е.В.**;

— Кашбиев Ильфат, Республика Татарстан, Актанышский р-н, село Актаныш, средняя школа № 2, учитель **Гилязова Г.М.**;

— Низамов Максим, Республика Башкортостан, г. Уфа, лицей № 60, учитель **Гильзер Н.В.**;

— Поляков Илья, Совхозная средняя школа, Московская обл., Серебряно-Прудский р-н, поселок Успенский, учитель **Жарикова Е.Н.**;

— Тагирова Алина, г. Астрахань, школа № 33 им. Н.А. Мордовиной, учитель **Лепехина С.М.**;

— Берлинов Руслан, Тарасов Дмитрий и Филимонов Олег, г. Орел, лицей № 4 им. Героя Советского Союза Г.Б. Злотина, учитель **Чапкевич И.М.**;

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

— Шейкин Александр, средняя школа села Ириновка, Новобураский р-н Саратовской обл., учитель **Брунов А.С.**

Шестнадцать офицеров

Один из возможных вариантов ответа:

П1	М4	К2	Л3
Л2	К3	М1	П4
М3	П2	Л4	К1
К4	Л1	П3	М2

Правильные ответы прислали:

— Антипов Анатолий, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

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

— Вострикова Анна, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Живило Андрей, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Касатонova Мария и Шарапова Мария, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Тимошин Виталий, средняя школа поселка Троицкое Пензенской обл., учитель **Симоненко О.В.**

На шахматной доске

Ответ

Всего есть 15 маршрутов, ведущих из клетки с5 шахматной доски в клетку а1:

- 1) с5, b5, a5, a4, a3, a2, a1;
- 2) с5, b5, b4, a4, a3, a2, a1;
- 3) с5, b5, b4, b3, a3, a2, a1;
- 4) с5, b5, b4, b3, b2, a2, a1;
- 5) с5, b5, b4, b3, b2, b1, a1;
- 6) с5, c4, b4, a4, a3, a2, a1;
- 7) с5, c4, b4, b3, a3, a2, a1;
- 8) с5, c4, b4, b3, b2, a2, a1;
- 9) с5, c4, b4, b3, b2, b1, a1;
- 10) с5, c4, c3, b3, a3, a2, a1;
- 11) с5, c4, c3, b3, b2, a2, a1;
- 12) с5, c4, c3, b3, b2, b1, a1;
- 13) с5, c4, c3, c2, b2, a2, a1;
- 14) с5, c4, c3, c2, b2, b1, a1;
- 15) с5, c4, c3, c2, c1, b1, a1.

Правильные ответы представили:

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

— Вострикова Анна, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Голик Екатерина, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Согомонян Серине, Воронежская обл., поселок Каменка, средняя школа № 1 им. Героя Советского Союза В.П. Захарченко, учитель **Старикова М.Е.**;

— Тимошин Виталий, средняя школа поселка Троицкое Пензенской обл., учитель **Симоненко О.В.**

Задача “Если закодировать ВГАБ”

Напомним, что необходимо было, используя систему кодирования букв А, Б, В и Г в виде двухрядных двоичных чисел (от 00 до 11), определить восьмеричные числа:

1) соответствующие закодированной последовательности символов ВГАБ;

2) максимальное и минимальное числа, которые можно при этом получить, если конкретный вариант указанной кодировки неизвестен.

Ответы

1) $10110001_2 — 261_8$;

2) $11100100_2 — 344_8$ (максимальное),
 $00011011_2 — 33_8$ (минимальное).

Правильные ответы прислали:

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

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

— Вострикова Анна, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Горьков Игорь, Грибанов Владлен, Дукач Светлана, Искандарова Лилия и Соболев Иван, г. Лесосибирск Красноярского края, поселок Стрелка, школа № 8 им. Константина Филиппова, учитель **Лопатин М.А.**;

— Живило Андрей, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

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

— Тимошин Виталий, средняя школа поселка Троицкое Пензенской обл., учитель **Симоненко О.В.**;

— Юматова Светлана, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**

Последовательности цифр

Ответы

1. Последняя строка заканчивается на следующие 10 цифр: 0123456789.

2. Цифра 4 первый раз появляется в строке (4), в следующей строке она встречается 2 раза и так далее, поэтому в последней строке цифра 5 встретится 32 раза.

3. В последней строке 1023 цифры. Можно считать количество вхождений каждой цифры, как это сделано для цифры 4 (для остальных — аналогично), а потом сложить, но экономнее обосновать, что при заданном рекуррентном соотношении получается $d_i = 2 * d_{i-1} + 1 = 2^{i+1} - 1$, откуда получаем, что для последней строки (9) $d_9 = 2^{10} - 1 = 1023$. Здесь d_i — длина i -й строки.

4. В последней строке последние 10 цифр мы знаем, а перед ними идет комбинация из цифр “10”, поэтому на 1012-м месте стоит цифра 1 (далее — цифра 0, а затем 10 цифр от 0 до 9).

5. Используя результат пункта 3, получим, что суммарное количество цифр во всех строках равно:

$$2^1 - 1 + 2^2 - 1 + 2^3 - 1 + 2^4 - 1 + 2^5 - 1 + 2^6 - 1 + 2^7 - 1 + 2^8 - 1 + 2^9 - 1 + 2^{10} - 1 = (2^{11} - 2) - 10 = 2048 - 12 = 2036.$$

Ответы представили:

— Антипин Никита, Куминская средняя школа, Тюменская область, Ханты-Мансийский автономный округ — Югра, Кондинский р-н, учитель **Шишигина О.В.**;

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

— Волков Владимир, Демьянова Елена и Хомякова Анна, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Касатонова Мария, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

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

— Согомонян Серине, Воронежская обл., поселок Каменка, средняя школа № 1 им. Героя Советского Союза В.П. Захарченко, учитель **Старикова М.Е.**

Задача “А и Б (но не на трубе ☺)”

Напомним, что необходимо было перечислить все цепочки из n букв А и Б, удовлетворяющие требованию: “никакая буква не должна стоять в цепочке подряд три или более раз”, при $n = 5$.

Решение

Удобно применить прием, примененный Юрием Базылевым, Республика Карелия, поселок Надвоицы, школа № 1 (учитель **Богданова Л.М.**), и Виталием Решетниковым, Вадьковская средняя школа, Брянская обл., Погарский р-н (учитель **Цыганкова И.Ю.**), — вместо букв А и Б использовать цифры 0 и 1:

00100

00101

00110

01001

01010

01011

01100

01101

10010

10011

10100

10101

10110

11001

11010

11011

Кроме Виталия и Юрия, ответы представили:

— Алексеева Алена, Голик Екатерина и Синицын Никита, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

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

— Касатонова Мария, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

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

— Михайлов Иван, Гаязова Фатима и Хорькова Анна, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Харитонов Сергей, средняя школа поселка Воскресенский Новгородской обл., учитель **Кузнецов А.А.**

Правильный ответ на задачу “Наименьшее число”, опубликованную ранее, представили:

— Ломакина Анастасия, средняя школа рабочего поселка Пинеровка, Саратовская обл., Балашовский р-н, учитель **Пичугин В.В.**;

— Лошак Антон, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**

Решения задач “Индюки и жеребята” и “Штирлиц и IP-адрес” прислал также Кузьмин В., Тобурдановская средняя школа, Чувашская Республика, Канашский р-н, учитель **Макарова Л.Ф.**, а задачи “Пять карточек” — Авдеев Александр, Гребеньков Егор, Жданов Никита, Исаков Максим, Кочкин Даниил, Машкин Дмитрий, Надворный Александр, Никифоров Алексей, Праслова Анжела, Сизикова Полина и Шибков Максим, Республика Карелия, г. Сегежа, школа № 5, учитель **Меньшиков В.В.**

Списки читателей, приславших ответы “обычной” почтой, будут опубликованы в следующем выпуске.

Еще одна задача о шахматной доске

Имеется шахматная доска с обозначением клеток согласно стандартной шахматной нотации ($a1$ — нижняя левая, ..., $h8$ — верхняя правая):

$a8$	$b8$	$c8$	$d8$	$e8$	$f8$	$g8$	$h8$
$a7$	$b7$	$c7$	$d7$	$e7$	$f7$	$g7$	$h7$
$a6$	$b6$	$c6$	$d6$	$e6$	$f6$	$g6$	$h6$
$a5$	$b5$	$c5$	$d5$	$e5$	$f5$	$g5$	$h5$
$a4$	$b4$	$c4$	$d4$	$e4$	$f4$	$g4$	$h4$
$a3$	$b3$	$c3$	$d3$	$e3$	$f3$	$g3$	$h3$
$a2$	$b2$	$c2$	$d2$	$e2$	$f2$	$g2$	$h2$
$a1$	$b1$	$c1$	$d1$	$e1$	$f1$	$g1$	$h1$

Из некоторой начальной клетки нужно проложить маршрут в клетку $a1$, соблюдая следующее правило: каждый ход делается либо на одну клетку влево, либо на одну клетку вниз, либо “вниз–влево”. Например, из клетки $d3$ допустимы ходы на клетки $c3$, $d2$, $c2$.

Для сокращения записи принята кодировка:

Л — ход влево;

Д — ход по диагонали “вниз–влево”;

Н — ход вниз.

Каждый маршрут записывается в виде набора букв, которые соответствуют обозначениям ходов (например, ЛДННД).

Перечислите все допустимые маршруты, ведущие из начальной клетки $c3$ в клетку $a1$.

Пятеро друзей

На скамейке около дома сидели четверо друзей: Антон, Борис, Виктор, Георгий и Дмитрий. Известно, что:

- 1) Виктор сидит правее Антона;
- 2) Дмитрий сидит левее Бориса;
- 3) Борис сидит рядом с Виктором;
- 4) Георгий сидит рядом с Виктором;
- 5) Антон и Борис не сидят рядом;
- 6) Дмитрий сидит на краю скамейки.

Выясните, в каком порядке сидели ребята.

Уравнение

В некоторой системе счисления записали уравнение $X6 * 2 = Y5$, где $X6$ — двузначное число с неизвестной первой цифрой и второй цифрой 6, $Y5$ — двузначное число с неизвестной первой цифрой и второй цифрой 5. Обе неизвестные цифры (и X , и Y) не могут быть равны нулю. Определите, в какой системе счисления составлено это уравнение, и найдите все его решения (то есть все пары чисел X и Y , являющиеся решениями).

Три ученика

Классный руководитель пожаловался директору школы, что у него в классе появилась компания из трех учеников, один из которых всегда говорит правду, второй — всегда лжет, а третий — говорит через раз то ложь, то правду. Директор знает, что их зовут Сергей, Максим и Кирилл, но не знает, кто из них правдив, а кто — нет. Однажды все трое опоздали на урок. Директор вызвал всех в свой кабинет и побеседовал с мальчиками. Кирилл сказал: “Максим никогда не врет. А вот от Сергея, наоборот, никогда не услышишь правды”. Максим сказал: “Кирилл сказал правду про меня”. Директор понял, кто из них кто. А сможете ли это сделать вы, уважаемый читатель?

Задачи Анании Ширакаци

Севак Карпетян,
ученик 7-го класса
гимназии № 1530, Москва

Анания Ширакаци (610-е годы – 685 г.) — знаменитый армянский ученый VII века. Точное место его рождения неизвестно. По мнению многих исследователей, он родился в Ширакаване или в деревне Анания области Ширак вблизи городища Ани и принадлежал к роду Камсаракан или Ар-

цруни — правителей этой области. Считается, что первоначальное образование он получил в расположенном неподалеку монастыре Дпреванк, где с ранних лет изучал математическую науку.

Ширакаци путешествовал в Византию, изучал математику и философию и, вернувшись на родину, основал школу, в которой преподавал математику, астрономию, географию. Им был составлен армянский учебник арифметики, выпущен трактат по космографии. Этот трактат свидетельствует о глубоком знании Ширакаци трудов греческого ученого Аристотеля. В своем сочинении он рассматривает и чисто астрономические вопросы: пытается оценить расстояние до Солнца и Луны, составляет календарь, свидетельствующий об основательном знании им движений Солнца и Луны и трудов древних ученых по этому вопросу. Ширакаци был разносторонним ученым, связавшим молодую армянскую науку с античным наследием.

В учебнике арифметики Ширакаци приведен ряд задач. Предлагаю читателям решить некоторые из них.

Задача № 8

“Во время восстания армян против персов, когда Заурак Камсаракан убил Сурена, один из азатов армянских отправил посла к персидскому царю, чтобы доложить ему эту печальную весть; он проезжал в день по пятьдесят миль; когда узнал об этом спустя пятнадцать дней Заурак Камсаракан, он отправил погоню вернуть его; гонцы проезжали в день по восемьдесят миль. Итак, узнай, во сколько дней они могли нагнать посла”.

Задача № 9

“Была у Камсараканов охота в Гене; захвачено было много дичи, и мне прислали, как долю добычи, кабана; так как он был чудовищен на вид, я взвесил его, и оказалось, что внутренности его составляли

четвертую часть всего веса, голова десятую часть, ноги двадцатую, клыки девяностую; туловище его весило двести двенадцать зитров. Итак, узнай, сколько всего зитров весил кабан”.

Задача № 14

“Было вино, сдобренное розой, в одном карасе, и были три каменных кувшина, и приказал я перелить в них вино; один вместил третью часть вина, другой шестую, а третий — четырнадцатую; а остальное вино перелили в другие сосуды, и составило это пятьдесят четыре паса. Итак, узнай, сколько всего было вина”.

Задача № 19

“Один муж зашел в три церкви и просил Бога в первой: «Поддай мне столько, сколько у меня есть, и дам я Тебе двадцать пять дахеканов». И во второй церкви так же просил он и дал двадцать пять дахеканов, так же и в третьей, и ничего у него не осталось. Итак, узнай, сколько у него прежде было”.

После решения прошу подумать о том, что учебник арифметики Ширакаци был написан более чем 1000 лет назад (!), и в те древние времена ученики — читатели книги знали только таблицу умножения и не имели тех знаний о вычислениях с дробями и о других особенностях математики, которые знаем мы. Так, например, ответ на одну из задач приводится не в виде $21\frac{7}{8}$, а как “двадцать один и половина и четверть и восьмая”.

Примечания

1. Зитра — старинная армянская мера веса (1 зитра \approx 326,4 г).
2. Пас — старинная армянская мера жидкостей (1 пас \approx 5,5 литра).
3. Дахекан — старинная армянская денежная единица.

ИСТОРИЯ ИНФОРМАТИКИ

Умножение решеткой и палочки Непера

Д.М. Златопольский, Москва

В средневековой Европе был широко распространен способ умножения многозначных чисел, известный как “умножение решеткой”, или “способ жалюзи”. По-видимому, он был разработан в Индии, но имел применение и в других странах Востока [1]. Этот способ легко уяснить на примере.

Пусть необходимо умножить 456 на 97.

Рисуетя табличка из трех столбцов (число 456 — трехзначное) и двух строк (97 — двузначное число), каждая клетка которой разделена диагональю так, как показано на рис. 1:

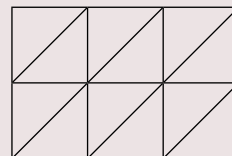


Рис. 1

Цифры чисел 456 и 97 записываются, соответственно, над табличкой и справа от нее (см. рис. 2).

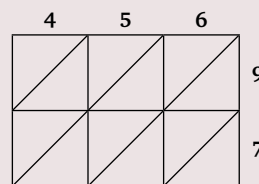


Рис. 2

После этого в каждую клетку записывается произведение цифры, стоящей в соответствующем столбце сверху, на цифру в соответствующей строке справа, причем десятки и единицы произведения разделяются упомянутой выше диагональю — *рис. 3*:

	4	5	6	
3	6	4	5	4
2	8	3	5	4
				9
				7

Рис. 3

Теперь можно определить результат умножения. Для этого необходимо просуммировать цифры по наклонным полоскам справа налево, при необходимости перенося “в уме” в соседнюю слева полоску единицу или двойку (см. *рис. 4*) и записывая эти суммы так, как показано на *рис. 4*.

	4	5	6	
3	6	4	5	4
2	8	3	5	4
				9
				7
	2	3	2	

Рис. 4

Результат следует читать слева от таблички сверху вниз, а затем под табличкой слева направо — он равен 44 232. Красиво, не правда ли?

Чтобы оценить преимущества умножения решеткой, предлагаю читателям сравнить время, требующееся для получения произведения, скажем чисел 53 896 и 274, при использовании этого способа и обычного умножения “в столбик”. Интересно, получатся ли результаты одинаковыми? ©

Способ умножения решеткой был положен в основу счетного прибора, описанного шотландским математиком Джоном Непером (кстати — изобретателем логарифмов) в 1617 году. Этот простой счетный прибор в дальнейшем получил название “палочки Непера”, “бруски Непера”, “пластины Непера” и т.п.

Прибор представлял собой набор прямоугольных пластин (палочек), в который входили:

- палочки с результатами умножения всех чисел от 0 до 9 на числа от 0 до 9; сверху каждой палочки наносилось число от 0 до 9 (на *рис. 5* справа показаны девять таких палочек). Результат умножения на палочках представлен двумя цифрами (в том числе начальным нулем), разделенными наклонной чертой;

- одна палочка с нанесенными на нее цифрами от 1 до 9 (указатель строк); на *рис. 5* она изображена слева.

Для умножения с помощью этого прибора выбирались палочки, соответствующие значениям разряда множимого, и выкладывались в ряд так, чтобы цифры сверху каждой палочки составляли множимое. На *рис. 6* показан пример умножения для числа 4938. Так как в множимом могли быть одинако-



Рис. 5

вые цифры, то необходимо было иметь несколько палочек с каждой цифрой.

Слева прикладывали палочку — указатель строк, по которой выбирали строки, соответствующие разрядам множителя. Для умножения, например, на 3 рассматривались соответствующие строки на палочках с цифрами 4, 9, 3 и 8. Результат умножения определялся следующим образом:

- последняя цифра произведения равна 4 (цифра под чертой в крайней справа палочке);

- остальные цифры определялись суммированием цифр “по наклонной линии”: предпоследняя цифра равна 1 (2 + 9 = 11, единица переходит в старший разряд), следующая справа цифра — 8 (0 + 7, и еще 1 перешло справа), следующая — 4 (2 + 2), первая слева — 1.

Итак, результат равен 14 814.

Если множитель являлся многозначным, то результаты, полученные для каждой строки (для каждой цифры множителя), складывались между собой с учетом порядка разрядов.

На *рис. 6* показан пример умножения числа 4938 на число 385.



Рис. 6

Задание для самостоятельной работы

Докажите (“словесно” или с помощью формул) справедливость использованного метода вычислений.

Литература

1. Анокин И.А., Майстров Л.Е. Развитие вычислительных машин. М.: Наука, 1974.
2. http://all-hitech.msk.ru/inf/history/p_0_12.html.

Нет, уважаемый читатель, в этой новой рубрике мы не будем рассказывать об известном американском киноактере, сыгравшем главную роль в одноименном фильме, Брюсе Уиллисе. В ней мы будем проводить анализ задач и головоломок, решение которых вызвало трудности.



Задача “Правильные часы”

Напомним условие. “Часовому мастеру принесли трое часов и попросили выверить их ход. Мастер включил секундомер и посмотрел на часы № 1 и 2. За 11 минут хода часов № 1 часы № 2 отсчитали 10 минут. Потом он сравнил часы № 2 и 3: за 12,5 минуты хода часов № 2 часы № 3 прошли 12 минут. Посмотрев затем в течение 8,25 минуты на часы № 1, мастер остановил секундомер и впервые взглянул на него — он отсчитал ровно 30 минут. Определите, какие часы идут точно. Укажите также номера часов в порядке увеличения «скорости» их работы”.

Решение

Из условия следует, что какие-то часы идут точно. Рассмотрим возможные варианты.

Пусть точно идут часы № 1. В этом случае при первом замере за “настоящих” 11 минут часы № 2 отсчитали 10 минут. Значит, каждая минута часов № 2 соответствует $\frac{10}{11}$ “настоящей” минуты. Следовательно, во время второго замера за 12,5 минуты этих же часов прошло $12,5 \times \frac{11}{10}$ минут “реального времени”. При третьем замере при нашем допущении было получено “настоящее” значение. Значит, должно соблюдаться равенство значений $11 + 12,5 \times \frac{11}{10} + 8,25$ и 30, а оно не соблюдается. То есть допущение неверное.

Пусть точно идут часы № 2. Тогда первые два замера продолжались $10 + 12,5 = 22,5$ минуты, а третий — $8,25 \times \frac{10}{11} = 7,5$ минуты, где $\frac{10}{11}$ — коэффициент для пересчета времени, показанного часами № 1, в “реальное”. Сумма времен трех замеров равна 30, что совпадает с показаниями секундомера.

Ответ на первое задание: точно идут часы № 2.

Можно также решить задачу, составив систему уравнений.

Ответ на второе задание: часы № 3, часы № 2, часы № 1.

Числовой ребус “БАЙТ на БАЙТ”

Напомним, что необходимо было решить числовой ребус:

$$\begin{array}{r}
 \times \quad \text{Б А Й Т} \\
 \text{Б А Й Т} \\
 \hline
 \text{Б * * *} \\
 * * \text{Б * *} \\
 * * * \text{Б *} \\
 * * * * \text{Б} \\
 \hline
 * * * * \text{Б * * *}
 \end{array}$$

Благодаря Игоря Горькова, Владлена Грибанова, Светлану Дукач, Лилию Искандарову и Ивана Соболева, г. Лесосибирск Красноярского края, поселок Стрелка, школа № 8 им. Константина Филиппова (учитель **Лопатин М.А.**), и Екатерину Голик, Владимирская обл., г. Струнино, школа № 11 (учитель **Волков Ю.П.**), представивших правильный ответ, предлагаем читателям еще раз подумать над его решением. При этом приведем начало решения ребуса.

Так как $\text{БАЙТ} \times \text{Т} = \text{Б***}$, то $\text{Т} = 1$. С учетом этого можем оформить пример в виде:

$$\begin{array}{r}
 \times \quad \text{Б А Й 1} \\
 \text{Б А Й 1} \\
 \hline
 * * \text{Б * Й} \\
 * * * \text{Б А} \\
 * * * * \text{Б} \\
 \hline
 * * * * \text{Б * * 1}
 \end{array}$$

Далее необходимо исследовать разряд, в котором определяется сумма $\text{Б} + \text{Б} + \text{Б} + \text{Б}$.

Звездочки, нули и единицы

Напомним, что было необходимо заменить звездочки единицами и нулями так, чтобы после преобразования полученных двоичных чисел в десятичные по горизонтали сумма равнялась 135, а по вертикали — 103:

$$\begin{array}{r}
 * \\
 * * * \\
 * \\
 * * * 0 * * * \\
 * \\
 * * * \\
 *
 \end{array}$$

Решение

Так как числа по горизонтали и по вертикали не могут начинаться на ноль, то можно записать в таблицу единицы:

			1			
		1	?	1		
			1			
1	?	?	0	?	?	1
			1			
		1	?	1		
			1			

После этого, учитывая весомости каждой единицы, надо определить сумму значений в каждой строке (оформлены синим) и в каждом столбце (оформлены зеленым):

			1				1	
		1	?	1			5	
			1				1	
1	?	?	0	?	?	1	65	
			1				1	
		1	?	1			5	
			1				1	
1		17	85	17		1		121
							79	

Кроме того, нужно рассчитать две суммы — оформленные цветом и полужирным начертанием (для расчетов целесообразно использовать электронную таблицу Microsoft Excel или др.).

Осталось только исследовать цифры на месте символов “?” и учесть их весомость так, чтобы вместо значений 121 и 79 получились значения, указанные в условии (соответственно, 103 и 135).

Решение головоломки “Звездочки, нули и единицы” и числового ребуса “БАЙТ на БАЙТ”, пожалуйста, присылайте в редакцию.

ЭТО ПОЛЕЗНО ЗНАТЬ

День числа пи

Праздники бывают разные, в каждой стране — свои. Есть “общенародные”, такие, как Новый год, День Победы и др. Есть так называемые “профессиональные” — День строителя, День железнодорожника и т.п. Но есть праздники, хотя и неофициальные, но которые отмечают люди многих профессий в разных странах.

Один из них — День числа пи. Этот праздник отмечается любителями математики 14 марта в 1:59:26. Его придумал в 1987 году физик из Сан-Франциско Ларри Шоу (Larry Shaw), подметивший, что в американской системе записи дат, в которой сначала указывается номер месяца, а затем — число, цифры даты 14 марта (3/14) и времени 1:59:26 совпадают с первыми разрядами числа $\pi = 3,1415926\dots$

Обычно праздник отмечают в 1:59:26 дня (в 12-часовом формате исчисления времени), но придерживающиеся 24-часового формата считают, что в этот момент время 13:59, и предпочитают праздновать ночью.

В это время читают хвалебные речи в честь числа пи, его роли в жизни человечества, пекут и едят пирог (кстати, слово “пирог” по-английски — *pie*) с изображением греческой буквы π или с первыми цифрами самого числа (их называют “пи-пирог”, “*pi pie*”), играют в игры, начинающиеся на пи, решают математические головоломки и загадки, водят хороводы вокруг предметов, связанных с этим числом.



Заметим, что существуют и пи-стихотворения — стихотворения, состоящие из последовательности слов, в которых количество букв представляет собой последовательность цифр числа пи (3,14...). На русском языке известны несколько анонимных пи-моностихов: “Что я знаю о кругах”, “Что я тебе и Дарье рассказал”, “Это я знаю и продолжаю знать” и даже двустишие: “Где волк и выдра властелин — ту страну лютый рок обошел”.

С.Фединым написан прозаический текст, где количество букв в словах соответствует числу π , — 3,141 592 653 589 793 238 462 643...: “Тип. Я ехал в метро. Тщедушный, но наглый пижон (шея как будто акушером вытянутая) брюзжал — затолкали его. Он был скользок... День спустя на Арбате вижу его...”.

Примечательно, что 14 марта родился Альберт Эйнштейн — создатель теории относительности...

“ЛОМАЕМ” ГОЛОВУ

Три задачи с песочными часами

Имеются двое песочных часов. Используя их, необходимо отмерить некоторый непрерывный промежуток времени. Исходная информация представлена в таблице:

Номер задачи	Первые часы	Вторые часы	Время, которое требуется замерить
1	4 мин.	10 мин.	2 мин.
2	3 мин.	4 мин.	2 мин.
3	3 мин.	10 мин.	5 мин.

Ответы, пожалуйста, оформите в виде схемы, аналогичной приведенной на рисунке (приведена схема решения задачи отмеривания 5 минут с помощью часов на 9 и 11 минут):

Первые часы (9 мин.)	1	3	5
Вторые часы (11 мин.)	2	4	н к

Примечание. На схеме “н” — начало отмеривания, “к” — конец.

Получить из 5 — 49

У исполнителя Удвоитель в систему команд входят две команды:

1. Прибавь 1.
2. Умножь на 2.

Выполняя первую из них, Удвоитель прибавляет к числу на экране 1, а выполняя вторую — умножает его на 2. Запишите порядок команд в программе получения из числа 5 числа 49.

В ответе укажите только номера команд.

Числовой ребус с “ПТИЧКАМИ”

Решите, пожалуйста, числовой ребус:

$$\begin{array}{r}
 \text{П Т И Ч К А} \\
 + \text{ П Т И Ч К А} \\
 \hline
 \text{С Т А Й К А}
 \end{array}$$

В нем, как обычно в таких головоломках, одинаковыми буквами зашифрованы одинаковые цифры, разными буквами — разные цифры.

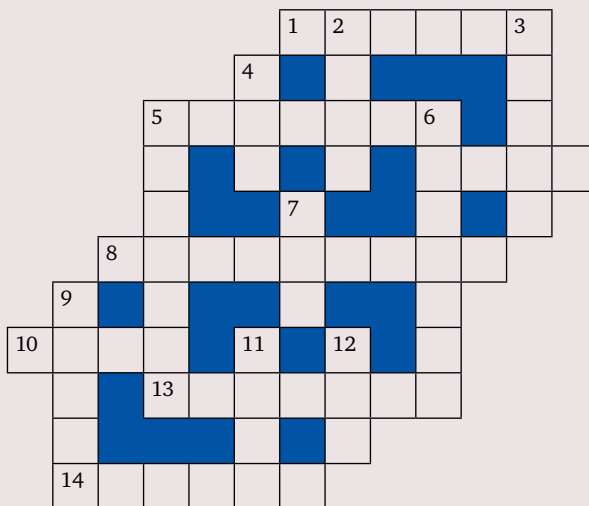
Расшифровка текста

Представьте себе, что в ваши руки попала следующая криптограмма (зашифрованный текст), написанная по принципу замены букв числами (одинаковые буквы заменялись одинаковыми числами):

- 1, 2, 3 – 2, 3 – 4, 5, 6, 7, 4, 8 – 2, 3, 7 – 9, 10, 2, 8
 11, 4, 12, 13, 14 – 1, 15, 16, 17 – 6 – 4, 9, 2 – 13, 9, 17,
 14, 18, 2, 19, 20
 21, 9, 13 – 18, 16, 4, 9, 11 – 22, 6, 23, 24 – 9, 13, 2, 9,
 25, 11, 14, 18, 2, 19, 20
 15, 16, 25, 13, 16, 3, 7, 4, 8 – 26, 22, 6, 25 – 1, 3, 2, 8

Слова в ней отделены друг от друга дефисом (“-”), буквы — запятыми. Известно, что в самом зашифрованном тексте тире и дефисов нет и что буквы е и ё закодированы одним и тем же числом. Расшифруйте эту криптограмму. Не бойтесь попробовать — это можно сделать путем рассуждений.

Кроссворд



По горизонтали

1. Совокупность элементов, каждый из которых содержит указатель на следующий элемент, а также перечень чего-нибудь или кого-нибудь.
5. Нестандартное устройство для вывода информации в компьютере.
7. “if” по-русски.
8. Совокупность правил записи символов в языке программирования.
10. Система знаков для секретного письма.
13. Устройство, осуществляющее преобразование представления и скорости передачи информации между ЭВМ и внешним устройством.
14. Условный знак или физический процесс, передающие некоторую информацию.

По вертикали

2. Цифра десятичной системы счисления.
3. Размер (высота) шрифта.
4. Единица измерения информации.
5. Набор цветов, которые могут использоваться в изображении.
6. Место хранения информации в процессоре.
7. Величина изменения значения переменной цикла.
9. Программа, обладающая способностью к самовоспроизведению.
11. ... данных.
12. Название буквы древнерусского алфавита, напоминающей твердый знак.

Ответы (можно не все) присылайте в редакцию.

ПОИСК ИНФОРМАЦИИ

Пять вопросов

1. Какой паук танцу имя подарил?
2. Коко Шанель любила повторять: “Мода проходит”. А что остается?
3. Кто из теннисистов выиграл Уимблдонский турнир в 17 лет?

4. Мальки какой аквариумной рыбки в случае опасности прячутся во рту у “папаши”?
5. Каково было содержание самой дорогой из проданных с аукциона бутылки вина?

Ответы присылайте в редакцию (можно отвечать не на все вопросы).

Итоги конкурса № 88

Напомним, что необходимо было решить ряд задач, связанных со спичками и их перекладыванием.

Ответы представили:

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

— Байрамалов Сергей, Свердловская обл., Красноуфимский р-н, Тавринская средняя школа, учитель **Ярцев В.А.**;

— Белов Никита, Воронежская обл., поселок Каменка, средняя школа № 1 им. Героя Советского Союза В.П. Захарченко, учитель **Старикова М.Е.**;

— Бородина Ирина и Корольчук Сергей, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Владимиров Виталий, Владимирова Снежана, Емельюкова Виктория, Семенова Екатерина и Яковлева Анастасия, основная школа села Именеве, Республика Чувашия, Красноармейский р-н, учитель **Тимофеева И.А.**;

— Воробьева Валерия, Республика Башкортостан, г. Стерлитамак, школа № 24, учитель **Орлова Е.В.**;

— Гималова Алина, средняя школа села Новое Барятино, Республика Башкортостан, Стерлитамакский р-н, учитель **Евдокимова Н.Л.**;

— Жуков Матвей, г. Самара, школа № 42, учитель **Фоменко Н.В.**;

— Зубкова Арина, Кадура Светлана, Куценко Евгения и Парамонова Анастасия, Ставропольский край, Кочубеевский р-н, станица Барсуковская, школа № 6, учитель **Рябченко Н.Р.**;

— Кашбиев Ильфат, Республика Татарстан, Актанышский р-н, село Актаныш, средняя школа № 2, учитель **Гилязова Г.М.**;

— Кушнир Максим, ученик 3-го класса школы № 9 г. Печора, Республика Коми, учитель **Нагаева И.В.**;

— Васильева Екатерина, Морозова Анна и Порчайкина Юлия, г. Иркутск, гимназия № 3, учитель **Лобх Т.В.**;

— Нагаева Алина, ученица 2-го класса школы № 9 г. Печора, Республика Коми, учитель **Чекрыгина Т.В.**;

— Хомушку Айгуля и Хомушку Чочагай, средняя школа села Барлык, Республика Тыва, Барун-Хемчикский р-н, учитель **Апын-оол Д.И.**

Ответы

Задача 1

Ответ. Учительница смотрела на примеры с другой стороны стола.

Задача 2

Решение показано на рис. 1.



Рис. 1

Задача 3

Решение представлено на рис. 2.

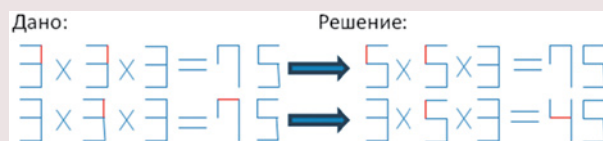


Рис. 2

Задача 4

Решение показано на рис. 3.



Рис. 3

Задача 5

Можно прочитать (как это принято при разгадывании ребусов с картинками) слово СУП (С у П).

Задача 6

Решение представлено на рис. 4.



Рис. 4

Задача 7

Можно прочитать слово "ГОВОР" (см. рис. 5).



Рис. 5

Матвей Жуков из школы № 42 г. Самары нашел также вариант, при котором получается слово "ОПОРА".

Задача 8

Точное время ограблений: 02-41, 03-41, 05-41. Так как это время — ночное, то погода не могла быть солнечной.

Задача 9

Ответ. Можно составить имя ЮРА (краткий вариант имени первого в мире космонавта Юрия Гагарина).

Задача 10

Решение показано на рис. 6.



Рис. 6

Задача 11

Решение представлено на рис. 7.

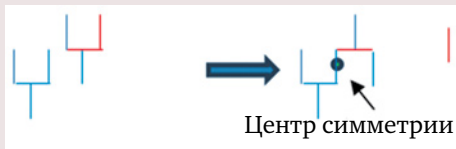


Рис. 7

Задача 12

Это задание вызвало наибольшие трудности. Напомним условие.

“Из спичек выложили два одинаковых “бокала”:

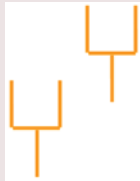


Рис. 8

Переставьте две спички так, чтобы получилась фигура, имеющая ось симметрии. Найдите два различных способа”.

Решение

1) первый способ:

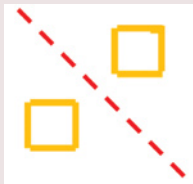


Рис. 9

2) второй способ:



Рис. 10

Задача 13

а) решение представлено на рис. 11.



Рис. 11

б) решение показано на рис. 12.



Рис. 12

Возможны также решения, в которых рыбка “плывет” вниз или вверх.

Задача 14

В задании 1 можно получить слово “СПИНКА”, в задании 2 — “СПЯЧКА”.

Учитывая сложность задания и возраст читателей, для которых были предназначены задания, редакция решила наградить дипломами всех участников конкурса — учащихся не старше 7-го класса. Поздравляем!

Итоги конкурса № 89 будут опубликованы в следующем выпуске.

Среди них — Акмалов Тагир и Валеев Артур, ученики 3-го класса средней школы села Урман, Республика Башкортостан, Иглинский р-н, также являвшиеся участниками конкурса (учитель **Товмасын М.Г.**).

ЗАДАЧНИК

Седловая точка

“Седловой точкой” таблицы называют такой ее элемент, который является наименьшим элементом в своей строке и наибольшим в своем столбце (или же наоборот — наибольшим в строке и наименьшим в столбце). Например, в таблице:

14	48	20	6
77	22	94	48
70	10	47	87
91	39	35	33
39	84	6	83
6	9	2	6
100	76	60	84

такой элемент выделен красным цветом (при втором определении понятия “седловая точка”). Ясно, что в таблице может быть несколько седловых точек, а может их и не быть вовсе.

Решим задачу нахождения седловых точек средствами электронной таблицы Microsoft Excel и средствами программирования.

1. Решение задачи средствами Microsoft Excel

Прежде чем приводить методику решения задачи, заметим, что в Excel имеются функции МИН и МАКС, возвращающие, соответственно, минимальное и максимальное значения из диапазонов ячеек, указанных в качестве их аргументов, а вот функции, возвращающей нужный нам ответ, — нет².

² Подумайте, можно ли в принципе создать такую функцию.

Зафиксировать факт наличия седловых точек, а также одновременно определить их количество можно следующим образом.

1. Заполним диапазон ячеек A1:F10 целыми случайными числами из интервала от 0 до 20 включительно. Для этого введем формулу =ЦЕЛОЕ(СЛЧИС()*10), которую затем распространим (скопируем) на остальные ячейки указанного диапазона.

	A	B	...	F
1	19	17		6
2	3	12		0
...				
10	7	0		10

При каждом нажатии функциональной клавиши **F9** будет происходить перерасчет случайных значений.

2. В строке 11 получим максимальные значения в каждом столбце, а в столбце G — минимальные значения в каждой строке. Соответствующие формулы введите самостоятельно.

3. В каждую из ячеек диапазона A13:F22 запишем 1, если “соответствующая ей” ячейка диапазона A1:F10 является седловой точкой, и 0 — в противном случае. Формула в ячейке A13, позволяющая с наименьшей трудоемкостью заполнить диапазон A13:F22, имеет вид:

=ЕСЛИ(И(A1=A\$11;A1=\$G1);1;0)

Именно эта формула копируется на остальные ячейки (почему это возможно — проанализируйте самостоятельно).

4. Просуммировав значения в диапазоне A13:F22, можно зафиксировать то, что нужно, — если сумма равна 0, то седловых точек нет, в противном случае — есть, а полученное значение при этом укажет на количество седловых точек.

2. Решение задачи средствами программирования

Здесь речь идет о седловых точках двумерного массива. Для их нахождения можно, как и при решении задачи в среде Microsoft Excel, использовать также два одномерных массива:

1) с именем *мин*, в котором будем хранить минимальные значения в каждой строке двумерного массива;

2) с именем *макс*, в который запишем максимальные значения в каждом столбце двумерного массива.

После этого сравним значение каждого элемента последнего с соответствующими значениями массивов *мин* и *макс*:

```
алг Седловые_точки
нач цел табл табл[1:m, 1:n], цел таб
мин[1:m], макс[1:n], цел ми, ма, i, j
. |Заполняем и выводим исследуемый
  |массив табл
. нц для i от 1 до m
. . нц для j от 1 до n
. . . табл[i, j] := int(rnd(10))
. . . вывод табл[i, j], " "
```

```
. . кц
. . вывод нс
. кц
. |Заполняем массив мин
. нц для i от 1 до m |Для каждой строки
. . |Принимаем в качестве значения ми
  |ее 1-й элемент
. . |(ми — текущее значение минимума
  |в строке)
. . ми := табл[i, 1]
. . |Рассматриваем остальные элементы
  |строки
. . нц для j от 2 до n
. . . если табл[i, j] < ми
. . . . то
. . . . . ми := табл[i, j]
. . . все
. . кц
. . мин[i] := ми
. кц
. |Заполняем массив макс
. нц для j от 1 до n |Для каждого столбца
. . |Принимаем в качестве значения
  |ма его 1-й элемент
. . |(ма — текущее значение максимума
  |в столбце)
. . ма := табл[1, j]
. . |Рассматриваем остальные элементы
  |столбца
. . нц для i от 2 до m
. . . если табл[i, j] > ма
. . . . то
. . . . . ма := табл[i, j]
. . . все
. . кц
. . макс[j] := ма
. кц
. |Проверяем каждый элемент двумерного
  |массива
. нц для i от 1 до m
. . нц для j от 1 до n
. . . если табл[i, j] = мин[i] и
  табл[i, j] = макс[j]
. . . . то . |Встретилась седловая точка
. . . . . вывод нс, , i, " ", j
. . . все
. . кц
. кц
кон
```

Но можно обойтись без использования массивов *мин* и *макс* — при рассмотрении каждого элемента двумерного массива находить минимальное значение в строке и максимальное в столбце и потом сравнивать данный элемент с этими значениями:

```
алг Седловые_точки_2 вариант
нач цел таб табл[1:m, 1:n],
  цел мин, макс, i, j
. нц для i от 1 до m
. . нц для j от 1 до n
. . . табл[i, j] := int(rnd(10))
. . . вывод табл[i, j], " "
. . кц
. . вывод нс
. кц
. нц для i от 1 до m
```

```

. . нц для j от 1 до n
. . . |Находим минимальный элемент
      |в строке
. . . мин := табл[i, 1]
. . . нц для j от 2 до n
. . . . если табл[i, j] < мин
. . . . . то
. . . . . мин := табл[i, j]
. . . . все
. . . кц
. . . |Находим максимальный элемент
      |в столбце
. . . макс := табл[1, j]
. . . нц для i от 2 до m
. . . . если табл[i, j] > макс
. . . . . то
. . . . . макс := табл[i, j]
. . . . все
. . . кц
. . . |Сравниваем
. . . если табл[i, j] = мин и
      табл[i, j] = макс
. . . . то
. . . . . вывод нс, i, " ", j
. . . все
. . кц
кц
кон

```

Видно, что во всех случаях решения учитывалось первое определение понятия “седловая точка”.

Задания для самостоятельной работы

1. Оформите лист электронной таблицы Microsoft Excel для решения обсуждаемой задачи так, чтобы в строке 11 и в столбце G максимальные значения в каждом столбце и минимальные значения в каждой строке не рассчитывались.

2. Проведя ряд экспериментов, ответьте, пожалуйста, на вопросы: “Зависит ли, по вашему мнению, частота появления точек минимакса при случайном заполнении таблицы от ее размеров? А от интервала значений чисел в таблице? Если зависит, то как?”

3. Попробуйте получить три точки минимакса. Что можно о них сказать? Попробуйте доказать этот факт. Таблицу с соответствующими данными и другие результаты выполнения заданий для самостоятельной работы пришлите в редакцию. Фамилии всех приславших будут опубликованы.

Цепочки из латинских букв

Строки (цепочки символов прописных латинских букв) создаются по следующему правилу.

Первая строка состоит из одного символа — латинской буквы “А”. Каждая из последующих цепочек создается такими действиями: в очередную строку сначала дважды подряд выписывается предыдущая строка, затем справа приписывается буква, чей порядковый номер в алфавите соответствует номеру строки (на i -м шаге пишется i -я буква алфавита). Вот первые 4 строки, созданные по этому правилу:

- (1) А
- (2) ААВ
- (3) ААВААВС
- (4) ААВААВСААВААВСD.

Запишите шесть символов подряд, стоящих в восьмой строке с 72-го по 77-е место (считая слева направо). Ответ обоснуйте.

Латинский алфавит (для справки):

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z.

Запросы к поисковому серверу

Когда вы ищете что-то в Интернете с помощью поисковой системы, то после ввода слов в строку поиска и нажатия клавиши **Enter** (или щелчка на соответствующей кнопке) поисковому серверу отправляется запрос. Допустим, что для обозначения логической операции “ИЛИ” в запросе используется символ “|”, а для логической операции “И” — символ “&”.

В таблице приведены четыре запроса к поисковому серверу:

1	графика литература
2	живопись литература графика
3	живопись & литература & графика
4	живопись & графика

Расположите номера запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу. Кстати, а как называется символ “&”?

Состав экспедиции

В состав экспедиции входят Антон, Сергей и Виктор. На обсуждении распределения обязанностей с руководством были высказаны предположения, что старшим будет назначен Антон, Сергей не будет механиком, а Виктор будет утверждён радистом, но старшим в экспедиции не будет.

Потом выяснилось, что только одно из этих четырех утверждений оказалось верным. Определите, кто какие занял должности в экспедиции.

На одной улице

На одной улице стоят в ряд 4 дома, в которых живут 4 человека: Богданов, Донников, Алексеев и Харитонов. Известно, что каждый из них владеет ровно одной из следующих профессий: врач, учитель, слесарь и парикмахер, но неизвестно, кто в каком доме живет. Однако имеется достоверная информация, что:

- 1) врач живет через дом от слесаря;
- 2) учитель живет левее врача;
- 3) парикмахер живет правее врача;
- 4) учитель живет не рядом со слесарем;
- 5) Харитонов не врач;
- 6) Богданов живет рядом с парикмахером;
- 7) Донников живет справа от врача;
- 8) Алексеев живет рядом с учителем.

Выясните, кто какой профессией владеет и кто где живет. Ответ оформите в порядке расположения домов слева направо.

ПОДПИСКА на 2-е полугодие 2012 г.

ж у р н а л

Информатика – Первое сентября

ТАРИФНЫЕ ПЛАНЫ

Максимальный – 1254 руб.

бумажная версия (по почте) + CD (по почте) + доступ к электронной версии на сайте

Оформление подписки – на сайте www.1september.ru или на почте по каталогам:

«Роспечать» – индекс 32291 (для индивидуальных подписчиков и организаций)

«Почта России» – индекс 79066 (для индивидуальных подписчиков и организаций)

Оптимальный – 594 руб.

электронная версия на CD (по почте) + доступ к электронной версии на сайте

Оформление подписки – на сайте www.1september.ru или на почте по каталогам:

«Роспечать» – индекс 19179 (для индивидуальных подписчиков и организаций)

«Почта России» – индекс 12684 (для индивидуальных подписчиков и организаций)

Экономичный – 200 руб.

доступ к электронной версии на сайте

Подписаться по данному тарифному плану можно только на сайте www.1september.ru

Бесплатный – 0 руб. ШКОЛА ЦИФРОВОГО ВЕКА

для педагогических работников образовательных учреждений, участвующих в Общероссийском проекте «Школа цифрового века».

Подробности – на digital.1september.ru



Бумажная версия
(доставка по почте)

CD с электронной версией
журнала
и дополнительными
материалами
для практической работы
(доставка по почте)

Электронная версия в Личном
кабинете подписчика
на сайте www.1september.ru.
Дополнительные материалы
включены

Пользователям электронной
версии высылаются по почте
подтверждающие документы

ЭКОНОМИЧНЫЙ тарифный план

ОПТИМАЛЬНЫЙ тарифный план

МАКСИМАЛЬНЫЙ тарифный план



Общероссийский проект **Школа цифрового века**

Интернет-сопровождение проекта – Издательский дом «ПЕРВОЕ СЕНТЯБРЯ»

2012/13
учебный год

**Предметно-методические
материалы**

**Дистанционные
модульные курсы**

**Бесплатно, адресно
каждому учителю!**

Участие образовательного учреждения в проекте «Школа цифрового века» в 2012/13 учебном году позволит каждому педагогическому работнику получать с 1 августа 2012 года по 30 июня 2013 года в свой Личный кабинет на сайте www.1september.ru предметно-методические журналы Издательского дома «Первое сентября» и пройти дистанционные модульные курсы по Программе развития профессионально-личностных компетенций педагога.

Для образовательных учреждений, подавших заявку до 20 июня 2012 года, оргвзнос – 4 тысячи рублей за весь учебный год.

Педагогическим работникам образовательного учреждения предоставляются документы, подтверждающие участие в проекте.

Прием заявок от школ начался

Подробности и регистрация на сайте

digital.1september.ru



Общероссийский проект «Школа цифрового века» по комплексному обеспечению образовательных учреждений методической интернет-поддержкой разработан в соответствии с программой модернизации системы общего образования России и направлен на повышение профессионального уровня педагогических работников