

№7-8

Издание основано в 1995 г.

inf.1september.ru

УЧЕБНО-МЕТОДИЧЕСКИЙ ЖУРНАЛ ДЛЯ УЧИТЕЛЕЙ ИНФОРМАТИКИ

ИНФОРМАТИКА



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

издательский
ДОМ
1september.ru

Первое сентября

июль-август
2015

ИНФОРМАТИКА Подписка на сайте www.1september.ru или по каталогу «Почта России»: 79066 — бумажная версия, 12684 — CD-версия



► 30 лет назад, 1 сентября 1985 г., предмет "Информатика" пришел в массовую школу. По поводу того, откуда он вышел на эту дорогу в школу, есть разные точки зрения. И, конечно, нельзя назвать одно такое место — эксперименты по обучению школьников программированию проводились в разных городах страны, в специализированных центрах и школах при различных научных центрах. Но один город и один научный центр всегда будет занимать в истории нашего предмета особую роль — Новосибирск, Академгородок. Этой небольшой скульптурной композиции — мыши, которая вяжет спираль ДНК, в Академгородке тогда еще, конечно, не было. Но за последнее время она стала очень узнаваемым символом.

Фото: фотобанк Лори

- 3** МАРАФОН-2015
► День учителя информатики
- 4** ОГЭ + ЕГЭ
► ЕГЭ: операции с массивами
- 34** ТЕХНОЛОГИИ
► Первые шаги к "дополненной реальности": Google Glass и другие
- 44** ЗАНИМАТЕЛЬНЫЕ МАТЕРИАЛЫ ДЛЯ ПЫТЛИВЫХ УЧЕНИКОВ И ИХ ТАЛАНТЛИВЫХ УЧИТЕЛЕЙ
► "В мир информатики" № 209

Облачные технологии от Издательского дома "Первое сентября"

Все подписчики журнала имеют возможность получать электронную версию, которая не только является полной копией бумажной, но и включает дополнительные электронные материалы для практической работы. Для получения электронной версии:

1) Откройте Личный кабинет на портале "Первое сентября" (www.1september.ru).

2) В разделе "Газеты и журналы / Получение" выберите свой журнал и кликните на кнопку "Я — подписчик бумажной версии".

3) Появится форма, посредством которой вы сможете отправить нам копию подписной квитанции.

После этого в течение одного рабочего дня будет активирована электронная подписка на весь период действия бумажной. Справки: podpiska@1september.ru или через службу поддержки на портале "Первое сентября".



ЭЛЕКТРОННЫЕ МАТЕРИАЛЫ

Презентации к статьям номера

ИНФОРМАТИКА

ПОДПИСНЫЕ ИНДЕКСЫ

по каталогу "Почта России": 79066 — бумажная версия, 12684 — электронная версия

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

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

РЕДАКЦИЯ:
гл. редактор С.Л. Островский
редакторы

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

Дизайн макета И.Е. Лукьянов
верстка Н.И. Пронская
корректор Е.Л. Володина
секретарь Н.П. Медведева
Фото: фотобанк Shutterstock
Журнал распространяется по подписке
Цена свободная
Тираж 18 000 экз.
Тел. редакции: (499) 249-48-96
E-mail: inf@1september.ru
<http://inf.1september.ru>



Школа цифрового века:
[facebook.com/School.of.Digital.Age](https://www.facebook.com/School.of.Digital.Age)

**ИЗДАТЕЛЬСКИЙ ДОМ
"ПЕРВОЕ СЕНТЯБРЯ"**

Главный редактор:
Артем Соловейчик
(генеральный директор)

Коммерческая деятельность:
Константин Шмарковский
(финансовый директор)

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

**Реклама, конференции
и техническое обеспечение
Издательского дома:**
Павел Кузнецов

Производство:
Станислав Савельев

**Административно-
хозяйственное обеспечение:**
Андрей Ушков

Педагогический университет:
Валерия Арсланьян (ректор)

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

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

УЧРЕДИТЕЛЬ:
ООО "ИЗДАТЕЛЬСКИЙ ДОМ
«ПЕРВОЕ СЕНТЯБРЯ»"

**Зарегистрировано
ПИ № ФС77-58447
от 25.06.2014**

в Роскомнадзоре
Подписано в печать:
по графику 15.04.2015,
фактически 15.04.2015
Заказ №
Отпечатано в ОАО "Первая
Образцовая типография"
Филиал "Чеховский Печатный Двор"

ул. Полиграфистов, д. 1,
Московская область,
г. Чехов, 142300
Сайт: www.chpd.ru
E-mail: sales@chpk.ru
Факс: 8 (495) 988-63-76
АДРЕС ИЗДАТЕЛЯ:
ул. Киевская, д. 24,
Москва, 121165
Тел./факс: (499) 249-31-38

Отдел рекламы:
(499) 249-98-70
<http://1september.ru>
ИЗДАТЕЛЬСКАЯ ПОДПИСКА:
Телефон: (499) 249-47-58
E-mail: podpiska@1september.ru

День учителя информатики-2015

► График подготовки и сдачи номеров журнала устроен таким образом, что лишь в этом номере мы можем разместить короткий фоторепортаж об очередном Дне учителя информатики, который прошел в рамках Московского педагогического марафона. День был замечательный, мероприятия — интересные, ведущие — блестящие, гости — довольные. На сайте марафон.1сентября.рф можно полностью посмотреть в записи два мероприятия этого дня.

Ждем весной 2016-го!



Людмила Леонидовна БОСОВА,
д. п. н., автор учебников, за кафедрой МПГУ



Надежда Николаевна САМЫЛКИНА,
к. п. н., профессор, автор учебников



Денис Павлович КИРИЕНКО,
учитель школы № 179 МИОО



Михаил Николаевич БОРОДИН,
директор издательства "Бином"



Евгений Карлович ХЕННЕР, д. ф.-м. н., проректор ПГУ, автор учебников информатики и учебников по методике преподавания информатики. Евгений Карлович вел мероприятие на самом первом Дне учителя информатики 14 лет назад. И вот 14 лет спустя...



Кадр из фильма об Игоре Геннадьевиче СЕМАКИНЕ, который был показан на семинаре, посвященном его памяти



ЕГЭ: операции с массивами

Массив знаний о массивах

► В статье, основанной на конспектах уроков и занятий со школьниками, приведен полный и систематический материал по теме “Массивы”.

Что такое массив

Во многих задачах требуется выполнять одни и те же операции с большим количеством данных одного и того же типа. Например, это может быть поиск требуемой фамилии в электронном телефонном справочнике, сортировка по возрастанию цен в прайс-листе интернет-магазина, вычисление средней годовой оценки в классе, изменение яркости растрового изображения путем изменения интенсивности цветовых составляющих для каждого пикселя и т.д.

Для таких применений предусмотрены составные типы данных, одним из наиболее широко используемых среди которых является тип “массив”.

Одномерный массив представляет собой пронумерованную последовательность переменных одного и того же типа, имеющих общее **имя**. Обращение к конкретной переменной (**элементу массива**) производится по **имени массива** и порядковому номеру (**индексу**) нужного элемента в нем.

Двумерный массив можно рассматривать как одномерный массив, элементами которого являются одномерные массивы. Другое представление двумерного массива — **матрица**: прямоугольная или квадратная таблица, в которой элементы массива соответствуют ячейкам, а номера строк и столбцов образуют два индекса каждого элемента.

Аналогично, трехмерный массив может быть рассмотрен как **куб данных**, состоящий из элементарных кубиков (элементов массива), каждый

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

Д.Ю. Усенков,
Москва

из которых имеет три индекса (т.е. трехмерный массив рассматривается как одномерный массив “слов” — матриц).

По тому же принципу могут быть определены массивы большей размерности.

Характерные элементы двумерного массива (матрицы)

Главная диагональ матрицы — это совокупность ее элементов, расположенных по диагонали от верхнего левого (первого) до нижнего правого (последнего) элемента. Эти элементы имеют равные значения индексов: $[1,1]$, $[2,2]$, ..., $[n,n]$, где n — размер (порядок) матрицы.

[1,1]	[1,2]	[1,3]	...	[1,n]
[2,1]	[2,2]	[2,3]	...	[2,n]
[3,1]	[3,2]	[3,3]	...	[3,n]
...
[n,1]	[n,2]	[n,3]		[n,n]

Побочная диагональ матрицы — это совокупность ее элементов, расположенных по диагонали от верхнего правого до нижнего левого элемента. Таким образом, побочная диагональ является “зеркально симметричной” по отношению к главной. Ее элементы имеют следующие значения индексов: $[1, n]$, $[2, n - 1]$, ..., $[n, 1]$, где n — размер (порядок) матрицы.

[1,1]	[1,2]	[1,3]	...	[1,n]
[2,1]	[2,2]	[2,3]	...	[2,n]
[3,1]	[3,2]	[3,3]	...	[3,n]
...
[n,1]	[n,2]	[n,3]		[n,n]

“Верхняя” матрица — это совокупность элементов, расположенных на главной диагонали и выше нее. Эти элементы имеют значения индексов: $[i, j]$, где i меняется от 1 до n (размер матрицы), а j — меняется во вложенном цикле от i до n . Для **“верхней” матрицы без главной диагонали** i меняется от 1 до n (размер матрицы), а j — меняется во вложенном цикле от $i + 1$ до n .

[1,1]	[1,2]	[1,3]	...	[1,n]
[2,1]	[2,2]	[2,3]	...	[2,n]
[3,1]	[3,2]	[3,3]	...	[3,n]
...
[n,1]	[n,2]	[n,3]		[n,n]

“Нижняя” матрица — это совокупность элементов, расположенных на главной диагонали и ниже нее. Эти элементы имеют значения индексов: $[i, j]$, где i меняется от 1 до n (размер матрицы), а j — меняется во вложенном цикле от 1 до i . Для **“нижней” матрицы без главной диагонали** i меняется от 2 до n (размер матрицы), а j — меняется во вложенном цикле от $i - 1$ до n .

[1,1]	[1,2]	[1,3]	...	[1,n]
[2,1]	[2,2]	[2,3]	...	[2,n]
[3,1]	[3,2]	[3,3]	...	[3,n]
...
[n,1]	[n,2]	[n,3]		[n,n]

Массивы в языке Паскаль

Объявление n -мерного массива:

array [<диапазон индекса 1>, ..., <диапазон индекса N>] **of** <базовый тип>

где <диапазон индекса> — записанные через две точки начальное и конечное значения индексов элементов по соответствующей размерности (целые числа или символы); <базовый тип> — тип элементов массива.

Примеры:

1) объявление одномерного массива из 10 целых чисел (базовый тип — integer), индексы которых меняются от 0 до 9:

```
var Mas: array [0..9] of integer;
```

2) объявление одномерного массива из 15 символов (базовый тип — char), индексы которых меняются от -7 до 7:

```
var Mas: array [-7..7] of char;
```

3) объявление одномерного массива из вещественных чисел (базовый тип — `real`), в качестве индексов которых используются латинские буквы:

```
var Mas: array ['a'..'z'] of real;
```

4) объявление двумерного массива размером 10×15 из целых чисел (базовый тип — `integer`), индексы которых отсчитываются с единицы:

```
var Mas: array [1..10, 1..15] of integer;
```

или

```
var Mas: array [1..10] of array [1..15] of integer;
```

Обращение к элементу:

1) одномерного массива: `Mas[i]`

2) двумерного массива: `Mas[i, j]`

Примечание. При обращении к элементу массива его индекс (индексы) может быть задан константой соответствующего типа (совпадающего с типом индекса в описании массива), переменной или выражением, результат которого имеет соответствующий тип. Это дает возможность реализовать поочередную обработку элементов массива в цикле, в котором требуемым образом меняются индексы элементов. При этом важно следить, чтобы значение индекса при обращении к элементу массива не выходило за пределы массива (за пределы указанного в его описании диапазона изменения индексов).

В программе может быть также описан “типовой” массив как своего рода “шаблон”, по которому затем могут быть определены “экземпляры” таких массивов. Пример:

```
type
  Mas = array [1..10] of integer;
var M1, M2: Mas;
```

Здесь вначале описан “шаблон” одномерного массива из 10 целочисленных элементов, а затем по нему, как по образцу, определены два таких массива с именами `M1` и `M2`. В результате с точки зрения программы оба эти массива полностью идентичны и для них допустима операция присваивания одного массива другому целиком: `M1 := M2` (все элементы массива `M2` копируются в соответствующие им элементы `M1`).

Заполнение массива

Производится в цикле (для многомерного массива — во вложенных циклах) поэлементно путем ввода каждого элемента с клавиатуры (оператором `read`) либо присваивания каждому элементу некоторого значения — константы (например, при обнулении массива) или вычисленного значения выражения.

Пример:

```
var Mas: array [1..10] of integer;
begin
  for i := 1 to 10 do
    read(Mas[i]);
  end.
```

Другой возможный вариант — указание значений элементов массива при его объявлении.

Примеры:

1) обнуление одномерного массива:

```
var mas: array[1..4] of integer := (0,0,0,0);
```

2) присваивание начальных значений элементам двумерного массива:

```
var mas: array[1..2,1..5] of integer := ((1,2,3,4,5), (6,7,8,9,10));
```

Вывод массива на экран

Производится в цикле (для многомерного массива — во вложенных циклах) поэлементно путем вывода каждого элемента с клавиатуры (оператором `write`).

Для одномерного массива обычно используется оператор `write`, в котором, кроме обращения к i -му элементу массива, предусмотрен разделяющий пробел (пример: `write(Mas[i], ' ');`), тогда элементы массива выводятся в строку.

Пример:

```
var Mas: array [1..10] of integer;
. . . . .
for i := 1 to 10 do
  write(Mas[i], ' ');
end.
```

Возможен альтернативный вариант, когда используется оператор `writeln`, в котором указывается значение индекса и значение элемента с этим индексом (пример: `writeln(i, '-й элемент равен ', Mas[i]);`), тогда каждый элемент выводится в отдельной строке.

Пример:

```
var Mas: array [1..10] of integer;
. . . . .
for i := 1 to 10 do
  writeln(i, '-й элемент равен ', Mas[i]);
end.
```

Для двумерного массива вывод строк обычно производится в одну строку (оператор `write`), а по завершении вывода очередной строки отдельно добавляется пустой оператор `writeln` для перехода на следующую строку.

```
var Mas: array [1..10,1..10] of integer;
. . . . .
for i := 1 to 10 do begin
  for j := 1 to 10 do
    write(Mas[i], ' ');
  writeln;
end;
end.
```

Обмен местами элементов массива

Производится аналогично обмену значений двух обычных переменных (обычно при помощи дополнительной “буферной” переменной).

Обработка элементов массива (определение максимума/минимума, вычисление суммы, произведения, среднего и пр.)

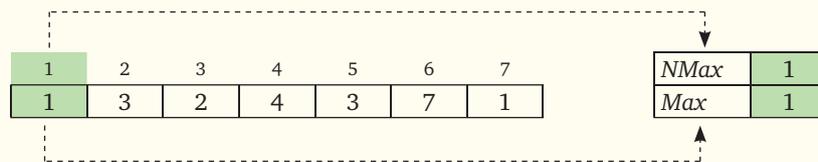
В цикле (для многомерного массива — во вложенных циклах) производится перебор элементов массива (полный или частичный — для фрагмента массива).

• **При поиске максимума/минимума** за предполагаемый максимум/минимум берется первый элемент массива либо константа, заведомо меньшая/большая любого элемента. Далее каждый очередной элемент массива сравнивается с предполагаемым максимумом/минимумом, и если этот элемент больше/меньше предполагаемого максимума/минимума, то значение этого элемента запоминается в качестве нового предполагаемого максимума/минимума. Дополнительно при этом в отдельной переменной (переменных) может перезапоминаться индекс (индексы) очередного предполагаемого максимума/минимума.

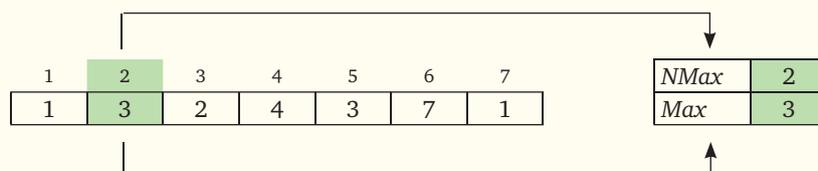
Поиск максимума	Поиск минимума
<pre>Max := Mas[1]; NMax := 1; for i := 2 to N do if Mas[i] > Max then begin Max := Mas[i]; NMax := i; end; end;</pre>	<pre>Min := Mas[1]; NMin := 1; for i := 2 to N do if Mas[i] < Min then begin Min := Mas[i]; NMin := i; end; end;</pre>

Пример: поиск максимума в массиве (1,3,2,4,3,7,1).

1) первоначально:



2) $i = 2$: $Mas[2] = 3$ больше, чем Max :

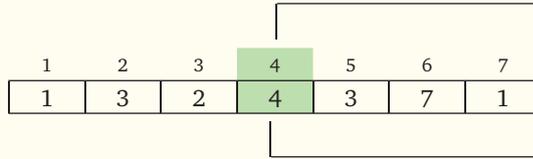


3) $i = 3$: $Mas[3] = 2$ меньше, чем Max :

1	2	3	4	5	6	7
1	3	2	4	3	7	1

$NMax$	2
Max	3

4) $i = 4$: $Mas[4] = 4$ больше, чем Max :



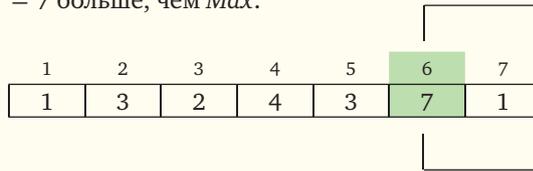
$NMax$	4
Max	4

5) $i = 5$: $Mas[5] = 3$ меньше, чем Max :

1	2	3	4	5	6	7
1	3	2	4	3	7	1

$NMax$	4
Max	4

6) $i = 6$: $Mas[6] = 7$ больше, чем Max :



$NMax$	6
Max	7

7) $i = 7$: $Mas[7] = 1$ меньше, чем Max :

1	2	3	4	5	6	7
1	3	2	4	3	7	1

$NMax$	6
Max	7

Результат:

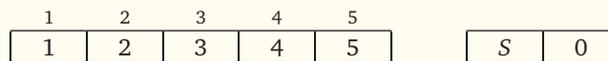
$NMax$	6
Max	7

• При вычислении суммы/произведения вначале переменной, выделенной для накопления суммы/произведения, присваивается инициализированное значение (ноль — для суммы, единица — для произведения). Затем в цикле (либо вложенных циклах) выполняется перебор элементов массива и текущее значение суммы/произведения складывается/умножается на текущий элемент массива.

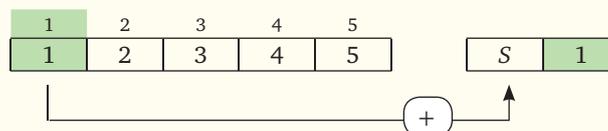
Вычисление суммы	Вычисление произведения
$S := 0;$ for $i := 1$ to N do $S := S + Mas[i];$	$P := 1;$ for $i := 1$ to N do $P := P * Mas[i];$

Пример: вычисление суммы элементов массива (1,2,3,4,5).

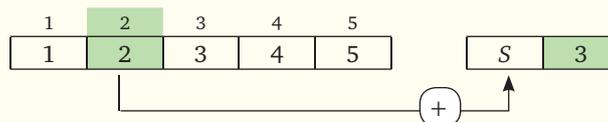
1) первоначально:



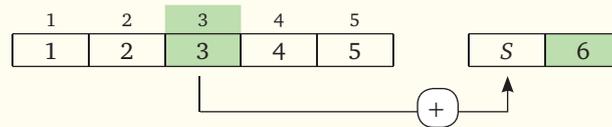
2) $i = 1$:



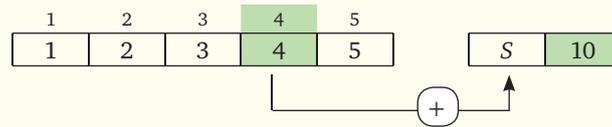
3) $i = 2$:



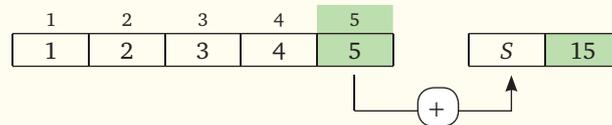
4) $i = 3$:



5) $i = 4$:



6) $i = 5$:



Результат:



• При вычислении среднего значения выполняется суммирование элементов массива, а затем полученная сумма делится на количество элементов в массиве.

```
S := 0;
for i := 1 to N do
  S := S + Mas[i];
Sredn := S/N;
```

• При определении максимума/минимума, суммы, произведения, среднего значения элементов, удовлетворяющих заданному условию (например, только положительных), дополнительно добавляется условный оператор с соответствующим условием, и требуемое действие (проверка и переприсваивание предполагаемого максимума/минимума, сложение, умножение) выполняется только при истинности этого условия. При вычислении среднего значения также предусматривается отдельная переменная-счетчик, которая увеличивается на один каждый раз, когда к сумме добавляется очередной удовлетворяющий условию элемент массива, и после вычисления суммы она делится на значение этого счетчика (количество вошедших в сумму элементов).

Примечание. В задаче на простой поиск минимума/максимума в качестве предполагаемого минимума/максимума проще всего брать первый элемент массива, а просмотр и сравнение их с предполагаемым минимумом/максимумом вести со второго элемента массива.

Если требуется поиск минимума/максимума среди элементов, удовлетворяющих заданным условиям, то формально брать в качестве предполагаемого минимума/максимума первый элемент уже нельзя: он может не удовлетворять заданным условиям.

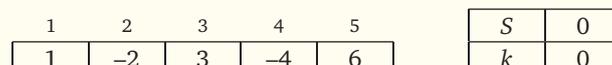
Если из условий задачи известен возможный диапазон значений элементов массива, то в качестве предполагаемого минимума можно брать константу, заведомо *большую*, чем возможное самое большое значение элементов массива, удовлетворяющих заданному условию, а в качестве предполагаемого максимума брать константу, заведомо *меньшую*, чем возможное самое маленькое значение элементов массива, удовлетворяющих заданному условию.

Если из условий задачи невозможно определить диапазон допустимых значений элементов массива, то в качестве предполагаемого минимума/максимума может быть взята произвольная константа (“заглушка”), заведомо *не удовлетворяющая* заданным условиям, а в операторе проверки условий переприсваивания предполагаемого минимума/максимума необходимо учесть необходимость замены этого “значения-заглушки” первым же элементом массива, удовлетворяющим условиям.

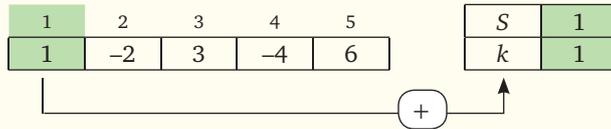
Пример 1: среднее арифметическое положительных элементов массива (1, -2, 3, -4, 6):

```
S := 0;
k := 0;
for i := 1 to N do
  if Mas[i] > 0 then begin
    S := S + Mas[i];
    k := k + 1;
  end;
Sredn := S/k;
```

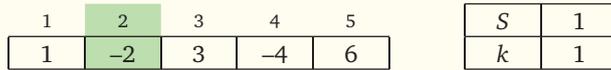
1) первоначально:



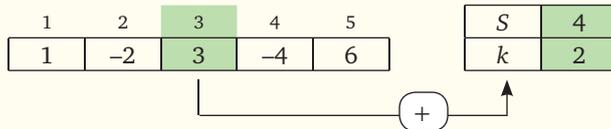
2) $i = 1: Mas[1] = 1$ — положительное число:



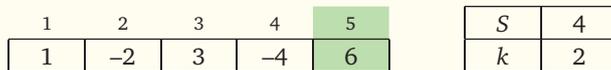
3) $i = 2: Mas[2] = -2$ — отрицательное число:



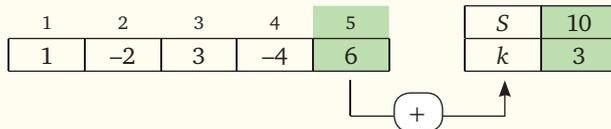
4) $i = 3: Mas[3] = 3$ — положительное число:



5) $i = 4: Mas[4] = -4$ — отрицательное число:



6) $i = 5: Mas[5] = 6$ — положительное число:



Результат:

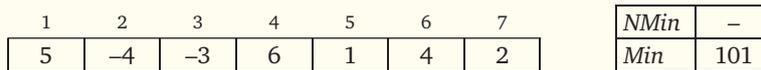
S	10
k	3

$$\Rightarrow Sredn = 10/3 = 3,3333$$

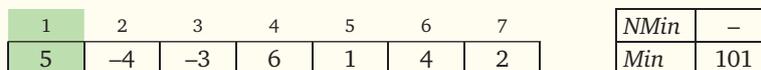
Пример 2: минимум среди четных положительных элементов массива (5, -4, -3, 6, 1, 4, 2) (известно, что значения элементов не превышают 100):

```
Min := 101; // предполагаемый минимум больше максимально возможного значения элемента
for i := 1 to N do
  if (Mas[i] > 0) AND (Mas[i] mod 2 = 0) AND (Mas[i] < Min) then
    begin
      Min := Mas[i];
      NMin := i;
    end;
end;
```

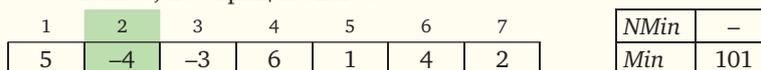
1) первоначально:



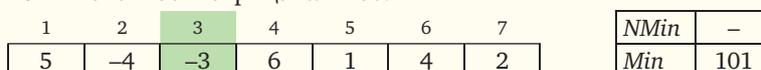
2) $i = 1: Mas[1] = 5$ — нечетное:



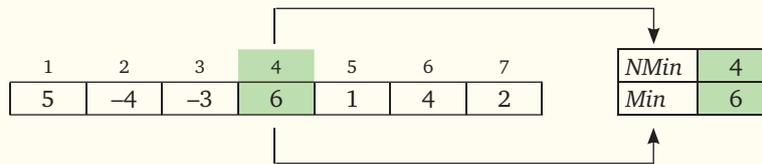
3) $i = 2: Mas[2] = -4$ — четное, но отрицательное:



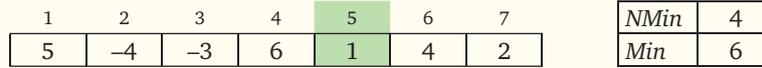
4) $i = 3: Mas[3] = -3$ — нечетное и отрицательное:



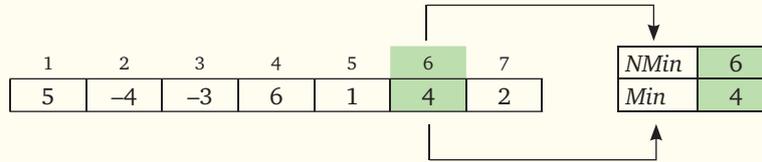
5) $i = 4$: $Mas[4] = 6$ — четное, положительное, меньше, чем Min :



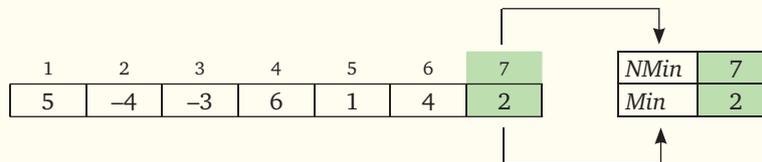
6) $i = 5$: $Mas[5] = 1$ — нечетное:



7) $i = 6$: $Mas[6] = 4$ — четное, положительное, меньше, чем Min :



8) $i = 7$: $Mas[7] = 2$ — четное, положительное, меньше, чем Min :



Результат:

<i>NMin</i>	7
<i>Min</i>	2

Пример 3: максимум среди нечетных положительных элементов массива (5, -4, -3, 7, 1, 3, 9), причем диапазон возможных значений элементов массива неизвестен:

```

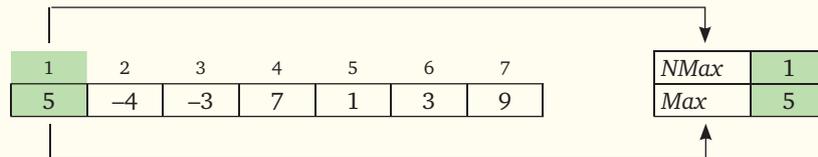
Max := 0; // поскольку требуется поиск максимума среди положительных
// элементов, ноль является значением, заведомо меньшим любого
// значения элемента массива, удовлетворяющего условию положительности
for i := 1 to N do
  if (Mas[i] > 0) AND (Mas[i] mod 2 <> 0) AND (Mas[i] > Max) then
    begin
      Max := Mas[i];
      NMax := i;
    end;
end;

```

1) первоначально:



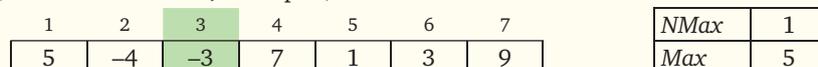
2) $i = 1$: $Mas[1] = 5$ — нечетное, положительное, больше, чем Max :



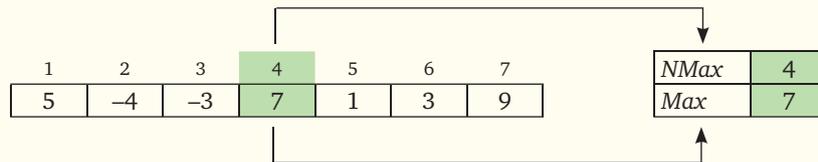
3) $i = 2$: $Mas[2] = -4$ — четное:



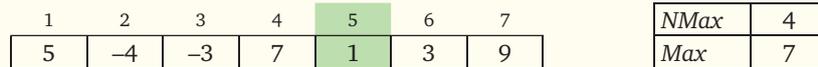
4) $i = 3$: $Mas[3] = -3$ — нечетное, но отрицательное:



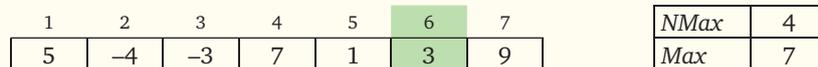
5) $i = 4$: $Mas[4] = 7$ — нечетное, положительное, больше, чем Max :



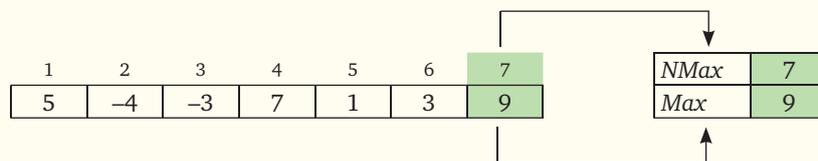
6) $i = 5$: $Mas[5] = 1$ — нечетное, положительное, но меньше, чем Max :



7) $i = 6$: $Mas[6] = 3$ — нечетное, положительное, но меньше, чем Max :



8) $i = 7$: $Mas[7] = 9$ — нечетное, положительное, больше, чем Max :



Результат:

$NMax$	7
Max	9

Пример 4: минимум среди положительных элементов массива (5, -6, -3, 9, 1, 3, 6), кратных трем, причем диапазон возможных значений элементов массива неизвестен:

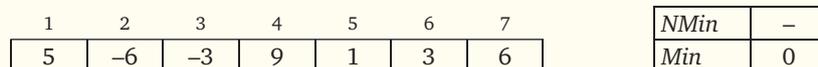
Примечание. Поскольку мы не знаем, какое возможно максимальное значение элементов массива, мы не можем указать, какое заведомо большее этого максимального значения число можно взять вместо предполагаемого минимума.

```

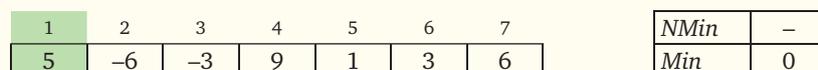
Min := 0; // требуется поиск минимума среди положительных
          // элементов массива; ноль не удовлетворяет этому условию
          // и потому может быть выбран в качестве "значения-заглушки"
          // для исходного присваивания предполагаемому минимуму
for i := 1 to N do
  if (Mas[i] > 0) AND (Mas[i] mod 2 <> 0) AND ((Mas[i] < Min) OR (Min = 0)) then
    // переписывание Min выполняется в двух случаях:
    // 1) если текущий элемент удовлетворяет заданным условиям,
    // а предполагаемый минимум равен "значению-заглушке";
    // цель этой операции — подставить в качестве предполагаемого минимума
    // первый же элемент массива, удовлетворяющий заданным условиям;
    // 2) если текущий элемент удовлетворяет заданным условиям и меньше
    // предполагаемого минимума
  begin
    Min := Mas[i];
    NMin := i;
  end;
end;

```

1) первоначально:



2) $i = 1$: $Mas[1] = 5$ — положительное, но не кратное 3:



3) $i = 2$: $Mas[2] = -6$ — отрицательное:

1	2	3	4	5	6	7
5	-6	-3	9	1	3	6

<i>NMin</i>	-
<i>Min</i>	0

4) $i = 3$: $Mas[3] = -3$ — отрицательное:

1	2	3	4	5	6	7
5	-6	-3	9	1	3	6

<i>NMin</i>	-
<i>Min</i>	0

5) $i = 4$: $Mas[4] = 9$ — положительное, кратное 3; кроме того, *Min* равно “значению-заглушке” — нулю:

1	2	3	4	5	6	7
5	-6	-3	9	1	3	6

<i>NMin</i>	4
<i>Min</i>	9

6) $i = 5$: $Mas[5] = 1$ — положительное, но не кратное 3:

1	2	3	4	5	6	7
5	-6	-3	9	1	3	6

<i>NMin</i>	4
<i>Min</i>	9

7) $i = 6$: $Mas[6] = 3$ — положительное, кратное 3, меньше, чем *Min*:

1	2	3	4	5	6	7
5	-6	-3	9	1	3	6

<i>NMin</i>	6
<i>Min</i>	3

8) $i = 7$: $Mas[7] = 6$ — положительное, кратное 3, но больше, чем *Min*:

1	2	3	4	5	6	7
5	-6	-3	9	1	3	6

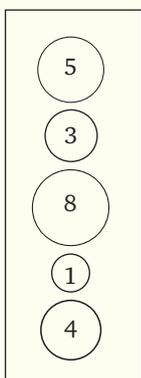
<i>NMin</i>	6
<i>Min</i>	3

Результат:

<i>NMin</i>	6
<i>Min</i>	3

Сортировка массива

Выполняется путем многократного просмотра массива, попарного сравнения его элементов между собой и при необходимости — соответствующей перестановке этих элементов местами (при сортировке по убыванию на первые места переносятся элементы с большими значениями; при сортировке по убыванию — наоборот).



Существует несколько методов сортировки массивов, различающихся сложностью их алгоритма и скоростью работы: сортировка перемешиванием, сортировка выбором, сортировка вставками, быстрая сортировка, пирамидальная, “гномья”, слиянием, деревом, сортировка Шелла, но наиболее понятной и наглядной является **пузырьковая сортировка** (она же — **сортировка простыми обменами**).

Рассмотрим метод пузырьковой сортировки на примере числового массива из пяти элементов: (5, 3, 8, 1, 4), — который нужно отсортировать по возрастанию. А для наглядности расположим их по вертикали, друг над другом, и обведем кружочками разного размера.

С чего начать? Если бы у нас было всего два элемента (например, числа 5 и 3, с которых начинается наш массив), то достаточно было бы сравнить их друг с другом и, если первое из этих чисел больше второго, поменять их местами.

Попробуем делать то же самое и для нашего массива — поочередно выбирать пары элементов (начиная с первого), сравнивать их и менять местами, если первый из взятой пары элементов окажется больше второго (напомним — нам требуется сортировка по возрастанию). А поскольку нам нужно разработать алгоритм для исполнения компьютером, воспользуемся конструкцией “цикл” (конечное значение цикловой переменной i определим в ходе разбора алгоритма).

№ цикла	№ шага	№ выбранных элементов (i и $i + 1$)	До обмена элементов	После обмена элементов
1	1	1 и 2		
1	2	2 и 3		
1	3	3 и 4		
1	4	4 и 5		

Как видим, в цикле значение цикловой переменной i должно меняться от 1 до 4, или, для массива произвольной длины n , — от 1 до $(n - 1)$. Запомним это.

Кроме того, мы видим, что хотя наш массив еще не отсортирован, самое большое число 8 уже “опустилось” в самый низ, на соответствующую ему правильную позицию. Легко показать, что и в любом другом массиве максимальный элемент окажется на последнем месте: “всплывет”, как пузырек. Этот факт и объясняет название алгоритма.

Что нужно теперь сделать? Правильно: повторить наш цикл еще раз! Однако, зная, что последнее число уже стоит на своем месте и его можно уже исключить из сравнений, будем теперь менять значение цикловой переменной от 1 до 3 (т.е. для произвольного массива — от 1 до $(n - 2)$).

№ цикла	№ шага	№ выбранных элементов (i и $i + 1$)	До обмена элементов	После обмена элементов
2	1	1 и 2		
2	2	2 и 3		
2	3	3 и 4		

Теперь и предпоследняя позиция в массиве тоже занята “правильным” элементом.

Очевидно, что для окончания сортировки в данном случае достаточно поменять местами два первых элемента. Но это — нам очевидно. Компьютер же, как мы знаем, является формальным исполнителем, — да и массив может быть каким угодно, тогда как наш алгоритм должен обладать свойством универсальности — быть пригодным для сортировки любого массива. Поэтому нам потребуется выполнить еще два цикла, в первом из которых переменная i должна меняться от 1 до 2, а во втором — быть равной числу 1, даже если один из шагов этого цикла окажется “лишним”.

№ цикла	№ шага	№ выбранных элементов (i и $i + 1$)	До обмена элементов	После обмена элементов
3	1	1 и 2		

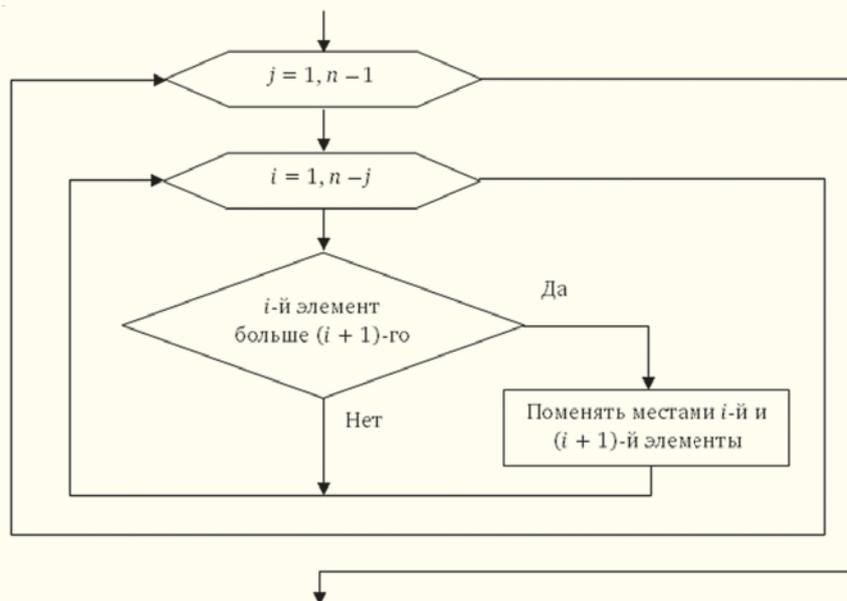
Продолжение табл. на с. 16

№ цикла	№ шага	№ выбранных элементов (i и $i + 1$)	До обмена элементов	После обмена элементов
3	2	2 и 3		
4	1	1 и 2		

А теперь обобщим только что разобранные действия на случай произвольного массива из n элементов:

- 1) потребуется $(n - 1)$ раз выполнять циклы просмотра элементов массива;
- 2) в каждом из этих циклов конечное значение цикловой переменной должно становиться меньше на 1, а в самом первом из этих циклов оно должно быть равно $(n - 1)$;
- 3) на каждом шаге выполняется сравнение пары элементов и (для сортировки по возрастанию) если первый из них больше второго, то они меняются местами.

Очевидно, что реализовать все это можно при помощи конструкции из двух вложенных циклов:



Напишем соответствующую программу (например, на Паскале):

```

program bubble_sort;
const nn = 10; // максимально возможный размер массива
var i, j: integer; // цикловые переменные
    n: integer; // реальный размер массива
    mass: array[1..nn] of integer; // массив
    t: integer; // вспомогательная переменная для обмена
    
```

```

begin
write('Введите размер массива: ');
readln(n);

writeln('Введите (через пробел) элементы массива');
for i := 1 to n do
  read(mass[i]);
// сортировка методом пузырька
for j := 1 to n - 1 do begin
  for i := 1 to n - j do begin
    if mass[i] > mass[i + 1] then begin
      // меняем местами элементы
      t := mass[i];
      mass[i] := mass[i + 1];
      mass[i + 1] := t;
    end;
  end;
end;
writeln('Отсортированный массив');
for i := 1 to n do
  write(mass[i], ' ');
writeln;
end.

```

Разбор задач из демоверсий ЕГЭ

Задача 1¹. В программе используется одномерный целочисленный массив A с индексами от 0 до 9. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre> For i = 0 To 9 A.SetValue(9 - i, i) Next i For i = 0 To 4 K = A.GetValue(i) A.SetValue(A.GetValue(9 - i), i) A.SetValue(k, 9 - i) Next i </pre>	<pre> for i := 0 to 9 do A[i] := 9 - i; for i := 0 to 4 do begin k := A[i]; A[i] := A[9 - i]; A[9 - i] := k; end; </pre>
Си	Алгоритмический язык
<pre> for (i = 0; i <= 9; i++) A[i] = 9 - i; for (i = 0; i <= 4; i++) { K = A[i]; A[i] = A[9 - i]; A[9 - i] = k; } </pre>	<pre> нц для i от 0 до 9 A[i] := 9 - i кц нц для i от 0 до 4 k := A[i] A[i] := A[9 - i] A[9 - i] := k кц </pre>

Чему будут равны элементы этого массива после выполнения фрагмента программы?

- 1) 9 8 7 6 5 4 3 2 1 0
- 2) 0 1 2 3 4 5 6 7 8 9
- 3) 9 8 7 6 5 5 6 7 8 9
- 4) 0 1 2 3 4 4 3 2 1 0

¹ A12/ДЕМО2012.

Решение

1. Модель массива А:

0	1	2	3	4	5	6	7	8	9

2. Массив при заполнении обрабатывается весь (диапазон изменения цикловой переменной совпадает с размером массива), а при изменении обрабатывается только часть массива (для i от 0 до 4).

3. Заполнение массива: каждому элементу присваивается значение, равное разности константы 9 и индекса этого элемента:

0	1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1	0

4. Изменение массива выполняется стоящей в теле второго цикла парой операторов:

```
k := A[i];
A[i] := A[9 - i];
A[9 - i] := k;
```

Эта алгоритмическая конструкция — обмен местами двух элементов (i -го и $(9-i)$ -го) через буферную переменную k .

Выполняем эту операцию последовательно для всех значений i :

$i = 0$:

0	1	2	3	4	5	6	7	8	9
<u>0</u>	8	7	6	5	4	3	2	1	<u>9</u>

$i = 1$:

0	1	2	3	4	5	6	7	8	9
0	<u>1</u>	7	6	5	4	3	2	<u>8</u>	9

$i = 2$:

0	1	2	3	4	5	6	7	8	9
0	1	<u>2</u>	6	5	4	3	<u>7</u>	8	9

$i = 3$:

0	1	2	3	4	5	6	7	8	9
0	1	2	<u>3</u>	5	4	<u>6</u>	7	8	9

$i = 4$:

0	1	2	3	4	5	6	7	8	9
0	1	2	3	<u>4</u>	<u>5</u>	6	7	8	9

То есть элементы массива меняются местами: первый — с последним, второй — с предпоследним и т.д. В результате мы получаем массив 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Ответ: массив 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (вариант ответа № 2).

Примечание. Будьте внимательны при определении обрабатываемой части массива! Если бы данный массив был обработан целиком, то, начиная со значения $i = 5$, происходил бы обратный обмен значений элементов, в результате чего мы вновь получили бы исходный массив.

Задача 2². В программе описан одномерный целочисленный массив А с индексами от 0 до 10. Ниже представлен фрагмент этой программы, записанный на разных языках программирования, в котором значения элементов массива сначала задаются, а затем меняются.

² А17/ДЕМО2011.

Бейсик	Паскаль
<pre>FOR i = 0 TO 10 A(i) = i - 1 NEXT i FOR i = 10 TO 1 STEP -1 A(i - 1) = A(i) NEXT i</pre>	<pre>for i := 0 to 10 do A[i] := i - 1; for i := 10 downto 1 do A[i - 1] := A[i];</pre>
Си	Алгоритмический язык
<pre>for (i = 0; i <= 10; i++) A[i] = i - 1; for (i = 10; i >= 1; i--) A[i - 1] = A[i];</pre>	<pre>нц для i от 0 до 10 A[i] := i - 1 кц нц для i от 10 до 1 шаг -1 A[i - 1] := A[i] кц</pre>

Чему окажутся равны элементы этого массива?

- 1) 999999999999
- 2) 01234567899
- 3) 012345678910
- 4) -1-1012345678

Решение

1. Модель массива A:

0	1	2	3	4	5	6	7	8	9	10

2. Массив обрабатывается весь.

3. Заполнение массива: каждому элементу присваивается значение, равное его индексу, уменьшенному на 1:

0	1	2	3	4	5	6	7	8	9	10
-1	0	1	2	3	4	5	6	7	8	9

4. Изменение массива выполняется циклом:

```
for i := 10 downto 1 do
  A[i - 1] := A[i];
```

То есть массив проходит справа налево (в обратном порядке — от последних элементов к первым), и текущий элемент записывается (копируется) в элемент с индексом на 1 меньше.

Выполняем эту операцию последовательно для всех значений i :

$i = 10$:

0	1	2	3	4	5	6	7	8	9	10
-1	0	1	2	3	4	5	6	7	<u>9</u>	9

$i = 9$:

0	1	2	3	4	5	6	7	8	9	10
-1	0	1	2	3	4	5	6	<u>9</u>	9	9

$i = 8$:

0	1	2	3	4	5	6	7	8	9	10
-1	0	1	2	3	4	5	<u>9</u>	9	9	9

$i = 7$:

0	1	2	3	4	5	6	7	8	9	10
-1	0	1	2	3	4	<u>9</u>	9	9	9	9

$i = 6$:

0	1	2	3	4	5	6	7	8	9	10
-1	0	1	2	3	<u>9</u>	9	9	9	9	9

$i = 5$:

0	1	2	3	4	5	6	7	8	9	10
-1	0	1	2	<u>9</u>	9	9	9	9	9	9

$i = 4$:

0	1	2	3	4	5	6	7	8	9	10
-1	0	1	<u>2</u>	9	9	9	9	9	9	9

$i = 3$:

0	1	2	3	4	5	6	7	8	9	10
-1	0	<u>2</u>	9	9	9	9	9	9	9	9

$i = 2$:

0	1	2	3	4	5	6	7	8	9	10
-1	<u>2</u>	9	9	9	9	9	9	9	9	9

$i = 1$:

0	1	2	3	4	5	6	7	8	9	10
<u>2</u>	9	9	9	9	9	9	9	9	9	9

Таким образом, в результате получаем массив, в котором все элементы равны 9.

Ответ: массив 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9 (вариант ответа № 1).

Задача 3³. В программе используется одномерный целочисленный массив A с индексами от 0 до 10. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre>FOR i = 0 TO 10 A(i) = i NEXT i FOR i = 0 TO 10 A(10 - i) = A(i) A(i) = A(10 - i) NEXT i</pre>	<pre>for i := 0 to 10 do A[i] := i; for i := 0 to 10 do begin A[10 - i] := A[i]; A[i] := A[10 - i]; end;</pre>
Си	Алгоритмический язык
<pre>for (i = 0; i <= 10; i++) A[i] = i; for (i = 0; i <= 10; i++) { A[10 - i] = A[i]; A[i] = A[10 - i]; }</pre>	<pre><u>нц</u> <u>для</u> <u>i</u> <u>от</u> 0 <u>до</u> 10 A[i] := i <u>кц</u> <u>нц</u> <u>для</u> <u>i</u> <u>от</u> 0 <u>до</u> 10 A[10 - i] := A[i] A[i] := A[10 - i] <u>кц</u></pre>

Чему будут равны элементы этого массива после выполнения фрагмента программы?

- 1) 10 9 8 7 6 5 4 3 2 1 0
- 2) 0 1 2 3 4 5 6 7 8 9 10
- 3) 10 9 8 7 6 5 6 7 8 9 10
- 4) 0 1 2 3 4 5 4 3 2 1 0

Решение

1. Модель массива A :

0	1	2	3	4	5	6	7	8	9	10

2. Массив обрабатывается весь (диапазон изменения цикловой переменной совпадает с размером массива).

3. Заполнение массива: каждому элементу присваивается значение, равное его индексу:

0	1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8	9	10

4. Изменение массива выполняется стоящей в теле второго цикла парой операторов:

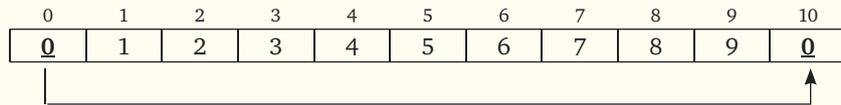
```
A[10 - i] := A[i];
A[i] := A[10 - i];
```

³ А6/ДЕМО2010.

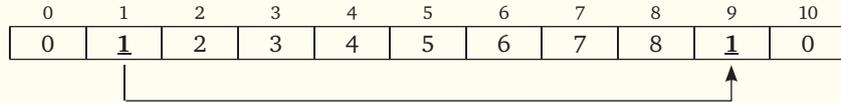
Эта алгоритмическая конструкция *похожа* на обмен местами двух элементов (i -го и $(10 - i)$ -го), но **не является обменом**, поскольку здесь не используется буферная переменная! Здесь просто сначала в $(10 - i)$ -й элемент заносится значение i -го элемента (а прежнее значение теряется), а затем выполняется обратное присваивание, которое, очевидно, уже ничего не меняет.

Выполняем эту операцию последовательно для всех значений i :

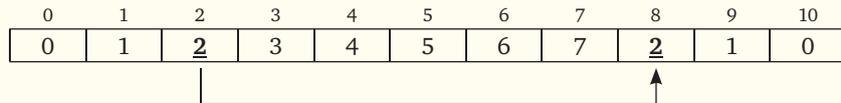
$i = 0$:



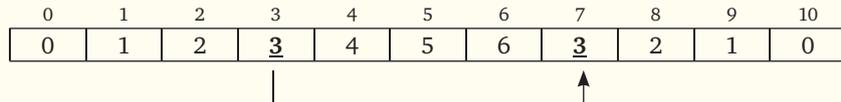
$i = 1$:



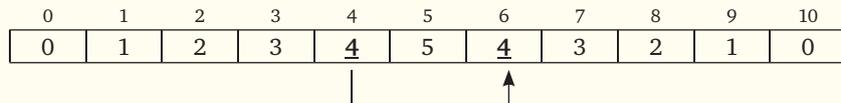
$i = 2$:



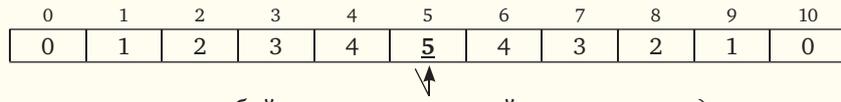
$i = 3$:



$i = 4$:

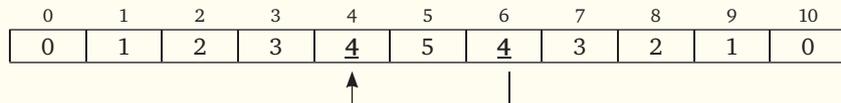


$i = 5$:



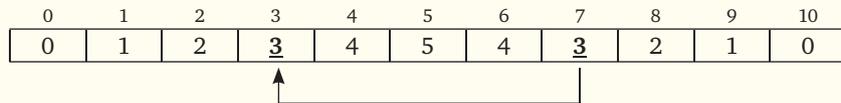
(При обмене элемента самого с собой никаких изменений не происходит.)

$i = 6$:

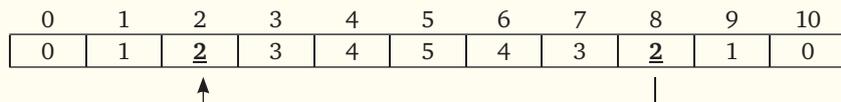


(Очевидно, при дальнейших операциях массив не изменяется.)

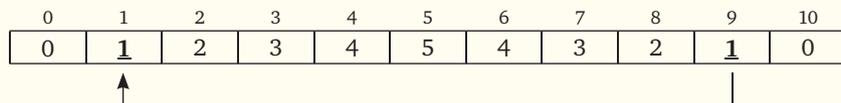
$i = 7$:



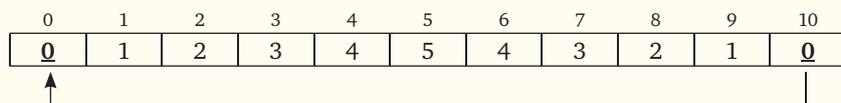
$i = 8$:



$i = 9$:



$i = 10$:



Таким образом, в результате получаем массив из элементов: 0, 1, 2, 3, 4, 5, 4, 3, 2, 1, 0.

Ответ: массив 0, 1, 2, 3, 4, 5, 4, 3, 2, 1, 0 (вариант ответа № 4).

Задача 4⁴. Все элементы двумерного массива A размером 10 × 10 элементов первоначально были равны 0. Затем значения элементов меняются с помощью вложенного оператора цикла в представленном фрагменте программы (ниже представлена одна и та же программа, записанная на разных языках программирования).

Бейсик	Паскаль	Алгоритмический язык
<pre>FOR n = 1 TO 4 FOR k = n TO 4 A(n, k) = A(n, k) + 1 A(k, n) = A(k, n) + 1 NEXT k NEXT n</pre>	<pre>for n := 1 to 4 do for k := n to 4 do begin A[n, k] := A[n, k] + 1; A[k, n] := A[k, n] + 1; end</pre>	<pre>нц для n от 1 до 4 нц для k от n до 4 A[n, k] := A[n, k] + 1 A[k, n] := A[k, n] + 1 кц</pre>

Сколько элементов массива в результате будут равны единице?

- 1) 0
- 2) 16
- 3) 12
- 4) 4

Решение

1. Модель двумерного массива:

k \ n	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

2. Выделяем обрабатываемую область:

k \ n	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

3. Заполняем массив поэлементно, отслеживая изменение индексных переменных (ниже приводится только нужный фрагмент модели массива; при его выделении нужно быть внимательным — в зависимости от записи индексов элемента возможен выход значений индексов за пределы изменения значений индексных переменных в цикле).

1) $n = 1, k = n = 1$

k \ n	1	2	3	4
1	2	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

$A[n, k] = A[k, n]$

⁴ А8/ДЕМО2005.

Вначале на единицу увеличено значение $A[n, k]$, т.е. $A[1, 1]$. Затем на единицу должно быть увеличено значение $A[k, n]$, т.е. $A[1, 1]$. Поскольку это тот же самый элемент, его значение увеличивается на единицу еще раз и становится равным 2. Это, очевидно, будет происходить всегда, когда $n = k$.

2) $n = 1, k = 2$

	k	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	
$A[n, k]$	n	<u>1</u>	2	1	0	0
		<u>2</u>	1	0	0	0
$A[k, n]$		<u>3</u>	0	0	0	0
		<u>4</u>	0	0	0	0

Вначале на единицу увеличено значение $A[n, k]$, т.е. $A[1, 2]$. Затем на единицу увеличено значение $A[k, n]$, т.е. $A[2, 1]$.

3) $n = 1, k = 3$

	k	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	
$A[n, k]$	n	<u>1</u>	2	1	1	0
		<u>2</u>	1	0	0	0
$A[k, n]$		<u>3</u>	1	0	0	0
		<u>4</u>	0	0	0	0

Вначале на единицу увеличено значение $A[n, k]$, т.е. $A[1, 3]$. Затем на единицу увеличено значение $A[k, n]$, т.е. $A[3, 1]$.

4) $n = 1, k = 4$

	k	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	
$A[n, k]$	n	<u>1</u>	2	1	1	1
		<u>2</u>	1	0	0	0
$A[k, n]$		<u>3</u>	1	0	0	0
		<u>4</u>	1	0	0	0

Вначале на единицу увеличено значение $A[n, k]$, т.е. $A[1, 4]$. Затем на единицу увеличено значение $A[k, n]$, т.е. $A[4, 1]$.

5) Внутренний цикл завершен, выполняется новый проход внешнего цикла и внутренний цикл запускается заново:

$n = 2, k = n = 2$

	k	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	
$A[n, k] = A[k, n]$	n	<u>1</u>	2	1	1	1
		<u>2</u>	1	2	0	0
		<u>3</u>	1	0	0	0
		<u>4</u>	1	0	0	0

Вначале на единицу увеличено значение $A[n, k]$, т.е. $A[2, 2]$. Затем на единицу должно быть увеличено значение $A[k, n]$, т.е. снова $A[2, 2]$.

6) $n = 2, k = 3$

	k	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	
$A[n, k]$	n	<u>1</u>	2	1	1	1
		<u>2</u>	1	2	1	0
$A[k, n]$		<u>3</u>	1	1	0	0
		<u>4</u>	1	0	0	0

Вначале на единицу увеличено значение $A[n, k]$, т.е. $A[2, 3]$. Затем на единицу увеличено значение $A[k, n]$, т.е. $A[3, 2]$.

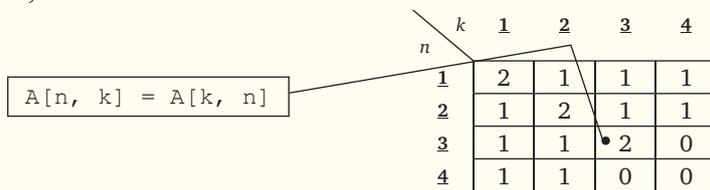
7) $n = 2, k = 4$

	k	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	
$A[n, k]$	n	<u>1</u>	2	1	1	1
		<u>2</u>	1	2	1	1
$A[k, n]$		<u>3</u>	1	1	0	0
		<u>4</u>	1	1	0	0

Вначале на единицу увеличено значение $A[n, k]$, т.е. $A[2, 4]$. Затем на единицу увеличено значение $A[k, n]$, т.е. $A[4, 2]$.

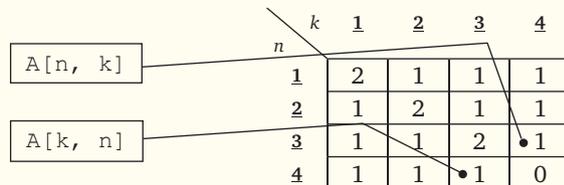
8) Внутренний цикл завершен, выполняется новый проход внешнего цикла и внутренний цикл запускается заново:

$$n = 3, k = n = 3$$



Вначале на единицу увеличено значение $A[n, k]$, т.е. $A[3, 3]$. Затем на единицу должно быть увеличено значение $A[k, n]$, т.е. снова $A[3, 3]$.

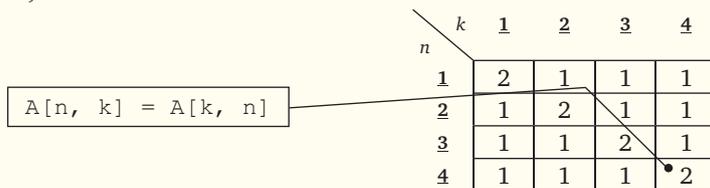
$$9) n = 3, k = 4$$



Вначале на единицу увеличено значение $A[n, k]$, т.е. $A[3, 4]$. Затем на единицу увеличено значение $A[k, n]$, т.е. $A[4, 3]$.

10) Внутренний цикл завершен, выполняется новый проход внешнего цикла и внутренний цикл запускается заново:

$$n = 4, k = n = 4$$



Вначале на единицу увеличено значение $A[n, k]$, т.е. $A[4, 4]$. Затем на единицу должно быть увеличено значение $A[k, n]$, т.е. снова $A[4, 4]$.

11) Внутренний цикл завершен, внешний цикл также завершен.

4. В полученном массиве подсчитываем количество единиц:

- всего ненулевых ячеек массива: $4 \times 4 = 16$ (остальные ячейки остались равны нулю и нас не интересуют);

- двойками заняты четыре ячейки;
- тогда единицы содержат $16 - 4 = 12$ ячеек.

Ответ: 12 элементов (вариант ответа № 3).

Задача 5⁵. 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 позволяет найти и вывести минимальный балл среди учащихся, получивших за тест положительную оценку. Известно, что в классе хотя бы один учащийся получил за тест положительную оценку.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	Бейсик
<pre>const N = 30; var a: array [1..N] of integer; i, j, min: integer; begin for i := 1 to N do readln(a[i]); ... end.</pre>	<pre>N = 30 DIM A(N) AS INTEGER DIM I, J, MIN AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>

⁵ С2/ДЕМО2010.

Си	Естественный язык
<pre>#include <stdio.h> #define N 30 void main(void) {int a[N]; int i, j, min; for (i = 0; i < N; i++) scanf("%d", &a[i]); } ...</pre>	<p>Объявляем целочисленные переменные I, J, MIN. В цикле от 1 до 30 вводим элементы массива А с 1-го по 30-й. ...</p>

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 те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

Решение

Данная задача подобна поиску минимума в массиве, но с дополнительным условием: искать минимум надо только среди элементов, равных или превышающих 20.

Соответственно этому, в типовой алгоритм поиска минимума следует внести следующие изменения:

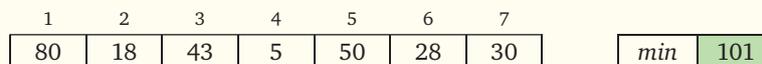
- поскольку мы не можем в качестве предполагаемого минимума первоначально использовать первый элемент (он может оказаться не удовлетворяющим условию “ ≥ 20 ”), нужно в качестве первого значения предполагаемого минимума брать число, заведомо превышающее любое значение в массиве (в данном случае это может быть число 101);
- в оператор if, проверяющий, не является ли текущий элемент просматриваемого массива меньшим, чем предполагаемый минимум, надо добавить условие “текущий элемент больше или равен 20”.

Полный текст программы:

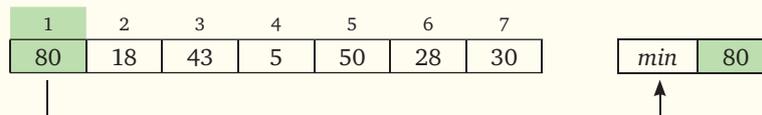
```
const N = 30;
var a: array [1..N] of integer;
    i, min: integer; // переменная j не используется
begin
  for i := 1 to N do readln(a[i]); // считали массив
  min := 101;
  // в качестве исходного предполагаемого минимума берется значение,
  // превышающее максимально возможное значение элемента в массиве
  for i := 1 to N do // полный просмотр массива
    if (a[i] >= 20) and (a[i] < min) then min := a[i];
  // если текущий элемент удовлетворяет условию (больше или равен 20)
  // И этот элемент меньше предполагаемого минимума, то этот элемент запоминаем
  // как новый предполагаемый минимум
  writeln(min); // вывод найденного минимума на экран
end.
```

Пример для массива (80,18,43,5,50,28,30).

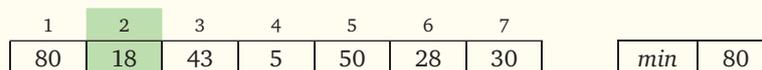
1) первоначально:



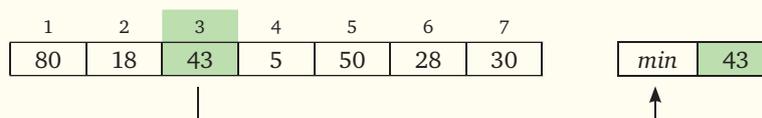
2) $i = 1$: $Mas[1] = 80$ — больше 20 и меньше, чем min:



3) $i = 2$: $Mas[2] = 18$ — меньше 20:



4) $i = 3$: $Mas[3] = 43$ — больше 20 и меньше, чем min:



5) $i = 4$: $Mas[4] = 5$ — меньше 20:

1	2	3	4	5	6	7	min	43
80	18	43	5	50	28	30		

6) $i = 5$: $Mas[5] = 50$ — больше 20, но больше, чем min :

1	2	3	4	5	6	7	min	43
80	18	43	5	50	28	30		

7) $i = 6$: $Mas[6] = 28$ — больше 20 и меньше, чем min :

1	2	3	4	5	6	7	min	28
80	18	43	5	50	28	30		

↑

8) $i = 7$: $Mas[7] = 30$ — больше 20, но больше, чем min :

1	2	3	4	5	6	7	min	28
80	18	43	5	50	28	30		

Результат:

min	28
-------	----

Задача 6⁶. 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.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	Бейсик
<pre>const N = 30; var a: array [1..N] of integer; i, x, y: integer; s: real; begin for i := 1 to N do readln(a[i]); ... end.</pre>	<pre>N = 30 DIM A(N) AS INTEGER DIM I, X, Y AS INTEGER DIM S AS SINGLE FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
Си	Естественный язык
<pre>#include <stdio.h> #define N 30 void main(void) {int a[N]; int i, x, y; float s; for (i = 0; i < N; i++) scanf("%d", &a[i]); ... }</pre>	<p>Объявляем массив А из 30 элементов. Объявляем целочисленные переменные I, X, Y. Объявляем вещественную переменную S. В цикле от 1 до 30 вводим элементы массива А с 1-го по 30-й.</p> <p>...</p>

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.

Решение

Речь идет о вычислении среднего арифметического элементов массива, удовлетворяющих заданному условию (здесь — нечетность). Для этого нужно в цикле просмотра массива подсчитывать сумму только элементов, удовлетворяющих этому условию, и одновременно подсчитывать количество таких элементов.

⁶ С2/ДЕМО2011.

Полный текст программы:

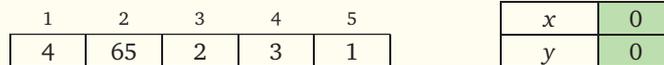
```

const N = 30;
var a : array [1..N] of integer;
    i, x, y: integer;
    s: real;
begin
  for i := 1 to N do readln(a[i]); // считали массив
  x := 0;
  y := 0;
  // x – переменная для подсчета суммы элементов, удовлетворяющих заданному условию;
  // y – переменная для подсчета количества этих элементов;
  // изначально обе переменные обнуляются
  for i := 1 to N do // полный просмотр массива
    if (a[i] mod 2 = 1) then begin
      // если текущий элемент удовлетворяет условию (нечетный), то
      x := x + a[i];
      // прибавляем этот элемент к накапливаемой сумме
      y := y + 1;
      // и увеличиваем количество просуммированных элементов на 1
    end;
  s := x/y;
  // по окончании цикла вычисляем среднее арифметическое,
  // деля полученную сумму нечетных чисел на их количество
  writeln(s); // вывод среднего арифметического значения нечетных элементов
end.

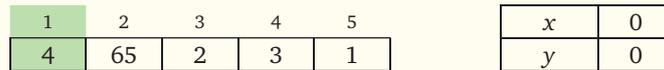
```

Пример для массива (4,65,2,3,1).

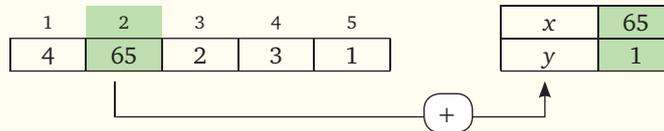
1) первоначально:



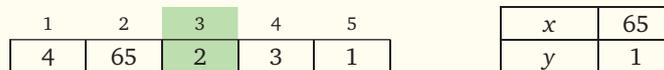
2) $i = 1$: Mas[1] = 4 — четное:



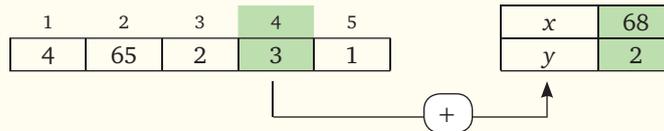
3) $i = 2$: Mas[2] = 65 — нечетное:



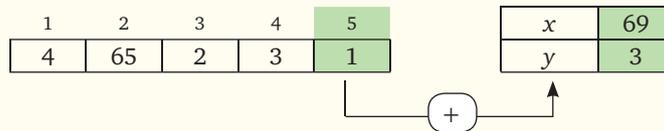
4) $i = 3$: Mas[3] = 2 — четное:



5) $i = 4$: Mas[4] = 3 — нечетное:



6) $i = 5$: Mas[5] = 1 — нечетное:



Результат:



Задача 7⁷. Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 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.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но использовать все описанные переменные не обязательно.

Паскаль	Алгоритмический язык
<pre>const N = 20; var a: array [1..N] of integer; i, j, min: integer; begin for i := 1 to N do readln(a[i]); ... end.</pre>	<pre>алг нач цел N = 20 целтаб a[1:N] цел i, j, MIN нц для i от 1 до N ввод a[i] кц ... кон</pre>
Бейсик	Си
<pre>N = 20 DIM A(N) AS INTEGER DIM I, J, MIN AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>#include <stdio.h> #define N 20 void main(void){ int a[N]; int i, j, min; for (i = 0; i < N; i++) scanf("%d", &a[i]); ... }</pre>
Естественный язык	
<p>Объявляем массив A из 20 элементов. Объявляем целочисленные переменные I, J, MIN. В цикле от 1 до 20 вводим элементы массива A с 1-го по 20-й. ...</p>	

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 те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

Решение

Данная задача по смыслу решения полностью аналогична задаче 1: здесь также нужно искать минимум среди элементов массива, удовлетворяющих заданному условию (в данном случае — четность и неделимость нацело на 3).

Напомним изменения, которые нужно внести в типовой алгоритм поиска минимума в массиве:

- поскольку мы не можем в качестве предполагаемого минимума первоначально использовать первый элемент (он может оказаться не удовлетворяющим заданному условию), нужно в качестве первого значения предполагаемого минимума брать число, заведомо превышающее любое значение в массиве (в данном случае это может быть число 1001);

- в оператор `if`, проверяющий, не является ли текущий элемент просматриваемого массива меньшим, чем предполагаемый минимум, надо добавить условия “текущий элемент четный” и “текущий элемент не делится нацело на 3”.

Полный текст программы:

```
const N = 20;
var
a: array [1..N] of integer;
i, min: integer; // переменная j не используется
begin
for i := 1 to N do readln(a[i]); // считали массив
```

⁷ C2/ДЕМО2012.

```

min := 1001;
// в качестве исходного предполагаемого минимума берется значение,
// превышающее максимально возможное значение элемента в массиве
for i := 1 to N do // полный просмотр массива
if (a[i] mod 2 = 0) and (a[i] mod 3 <> 0) and (a[i] < min) then min := a[i];
// если текущий элемент удовлетворяет условиям: он четный И он не делится
// без остатка на 3 И этот элемент меньше предполагаемого минимума,
// то этот элемент запоминаем как новый предполагаемый минимум
writeln(min); // вывод найденного минимума на экран
end.

```

Пример для массива (333,80,43,60,44,3,68).

1) первоначально:

1	2	3	4	5	6	7	<i>min</i>	1001
333	80	43	60	44	3	68		

2) $i = 1$: $Mas[1] = 333$ — нечетное:

1	2	3	4	5	6	7	<i>min</i>	1001
333	80	43	60	44	3	68		

3) $i = 2$: $Mas[2] = 80$ — четное, не делится на 3, меньше, чем min :

1	2	3	4	5	6	7	<i>min</i>	80
333	80	43	60	44	3	68		

4) $i = 3$: $Mas[3] = 43$ — нечетное:

1	2	3	4	5	6	7	<i>min</i>	80
333	80	43	60	44	3	68		

5) $i = 4$: $Mas[4] = 60$ — четное, но делится на 3:

1	2	3	4	5	6	7	<i>min</i>	80
333	80	43	60	44	3	68		

6) $i = 5$: $Mas[5] = 44$ — четное, не делится на 3, меньше, чем min :

1	2	3	4	5	6	7	<i>min</i>	44
333	80	43	60	44	3	68		

7) $i = 6$: $Mas[6] = 3$ — нечетное:

1	2	3	4	5	6	7	<i>min</i>	44
333	80	43	60	44	3	68		

8) $i = 7$: $Mas[7] = 68$ — четное, не делится на 3, но больше, чем min :

1	2	3	4	5	6	7	<i>min</i>	44
333	80	43	60	44	3	68		

Результат:

<i>min</i>	44
------------	----

Задача 8. Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 до 1000. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести порядковый номер минимального значения среди нечетных положительных элементов массива.

Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого нечетно и положительно. Если в массиве имеется несколько равных друг другу элементов, значение которых является минимальным среди нечетных положительных чисел, то нужно вывести номер последнего такого элемента (т.е. максимальный из найденных номеров элементов — минимумов).

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но использовать все описанные переменные не обязательно.

Паскаль	Алгоритмический язык
<pre>const N = 20; var a: array [1..N] of integer; i, j, min, nmin: integer; begin for i := 1 to N do readln(a[i]); ... end.</pre>	<pre>алг нач цел N = 20 целтаб a[1:N] цел i, j, MIN, NMIN нц для i от 1 до N ввод a[i] кц ... кон</pre>
Бейсик	Си
<pre>N = 20 DIM A(N) AS INTEGER DIM I, J, MIN, NMIN AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>#include <stdio.h> #define N 20 void main(void){ int a[N]; int i, j, min, nmin; for (i = 0; i < N; i++) scanf("% d", &a[i]); ... }</pre>
Естественный язык	
<p>Объявляем массив А из 20 элементов. Объявляем целочисленные переменные I, J, MIN, NMIN. В цикле от 1 до 20 вводим элементы массива А с 1-го по 20-й. ...</p>	

В качестве ответа необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия.

Решение

Особенность этой задачи — в том, что при наличии в массиве *нескольких* минимальных элементов надо запомнить и вывести именно конкретный номер элемента — наибольший среди номеров таких элементов. Поэтому нужно принять меры, чтобы проигнорировать значения минимумов, номера которых меньше последнего такого элемента. Как это сделать?

Способ 1

В условии, проверяющем, является ли текущий элемент просматриваемого массива меньшим, чем предполагаемый минимум, надо использовать *знак нестрогого неравенства*. Тогда для встреченных далее значений, равных текущему минимуму, номер элемента *nmin* будет перезапоминаться. В итоге последним запомненным будет максимальный из номеров таких элементов.

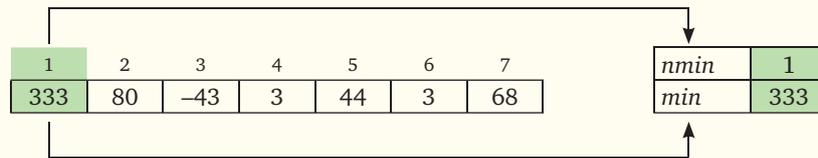
```
const N = 20;
var a: array [1..N] of integer;
    i, j, min, nmin: integer;
begin
for i := 1 to N do
readln(a[i]);
min := 1001;
// значение, заведомо большее максимального значения элементов
for i := 1 to N do
if (a[i] > 0) AND (a[i] mod 2 <> 0) AND (a[i] <= min) then
min := a[i];
nmin := i;
end;
end;
writeln('Номер последнего минимального элемента: ', nmin);
end.
```

Пример для массива (333, 80, -43, 3, 44, 3, 68).

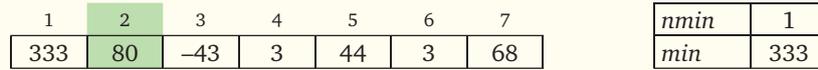
1) первоначально:

1	2	3	4	5	6	7	<i>nmin</i>	-
333	80	-43	3	44	3	68	<i>min</i>	1001

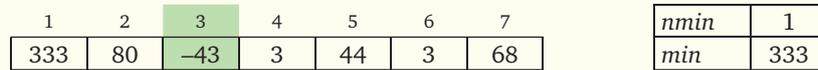
2) $i = 1$: $Mas[1] = 333$ — положительное, нечетное, меньше, чем min :



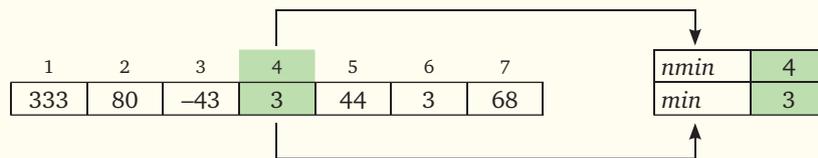
3) $i = 2$: $Mas[2] = 80$ — положительное, но четное:



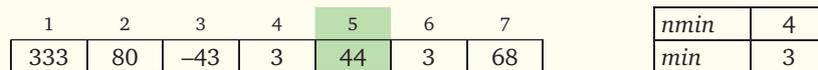
4) $i = 3$: $Mas[3] = -43$ — отрицательное:



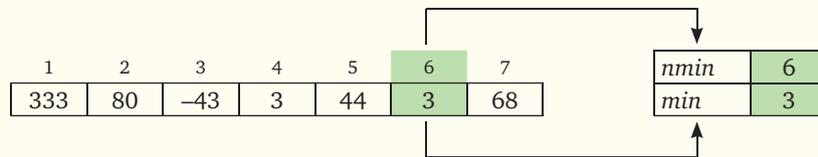
5) $i = 4$: $Mas[4] = 3$ — положительное, нечетное, меньше, чем min :



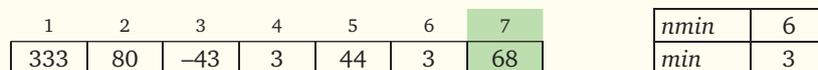
6) $i = 5$: $Mas[5] = 44$ — положительное, но четное:



7) $i = 6$: $Mas[6] = 3$ — положительное, нечетное, **равное** min :



8) $i = 7$: $Mas[7] = 68$ — положительное, но четное:



Результат:

$nmin$	6
min	3

Способ 2

Можно использовать знак строгого неравенства, но просмотр массива вести с конца. Тогда первый из встреченных равных друг другу минимумов будет запомнен, а все расположенные в массиве левее него такие минимумы будут проигнорированы (как равные, но не меньшие).

```
const N = 20;
var a: array [1..N] of integer;
    i, j, min, nmin: integer;
begin
  for i := 1 to N do
    readln(a[i]);
  min := 1001; // значение, заведомо большее максимального значения элементов
  for i := N downto 1 do
    if (a[i] > 0) AND (a[i] mod 2 <> 0) AND (a[i] < min) then
      min := a[i];
      nmin := i;
    end;
  end;
  writeln('Номер последнего минимального элемента: ', nmin);
end.
```

Пример для массива (333,80,-43,3,44,3,68).

1) первоначально:

1	2	3	4	5	6	7	<i>nmin</i>	-
333	80	-43	3	44	3	68	<i>min</i>	1001

2) $i = 7$: $Mas[7] = 68$ — положительное, но четное:

1	2	3	4	5	6	7	<i>nmin</i>	-
333	80	-43	3	44	3	68	<i>min</i>	1001

3) $i = 6$: $Mas[2] = 3$ — положительное, нечетное, меньше, чем *min*:

1	2	3	4	5	6	7	<i>nmin</i>	6
333	80	-43	3	44	3	68	<i>min</i>	3

4) $i = 5$: $Mas[5] = 44$ — положительное, но четное:

1	2	3	4	5	6	7	<i>nmin</i>	6
333	80	-43	3	44	3	68	<i>min</i>	3

5) $i = 4$: $Mas[4] = 3$ — положительное, нечетное, равно *min* (но не меньше!):

1	2	3	4	5	6	7	<i>nmin</i>	6
333	80	-43	3	44	3	68	<i>min</i>	3

6) $i = 3$: $Mas[3] = -43$ — отрицательное:

1	2	3	4	5	6	7	<i>nmin</i>	6
333	80	-43	3	44	3	68	<i>min</i>	3

7) $i = 2$: $Mas[2] = 80$ — положительное, но четное:

1	2	3	4	5	6	7	<i>nmin</i>	6
333	80	-43	3	44	3	68	<i>min</i>	3

8) $i = 1$: $Mas[1] = 333$ — положительное, нечетное, но большее, чем *min*:

1	2	3	4	5	6	7	<i>nmin</i>	6
333	80	-43	3	44	3	68	<i>min</i>	3

Результат:

<i>nmin</i>	6
<i>min</i>	3

Продолжается прием заявок на 2015/16 учебный год

Фестиваль педагогических идей «Открытый урок» festival.1september.ru

Свидетельство о регистрации СМИ Эл. № ФС77-53231

В течение 12 лет – самый массовый, представительный и посещаемый педагогический форум Рунета. Самая большая коллекция авторских разработок учителей.

Разместить публикацию может каждый педагог. Всем авторам предоставляются документы о публикации. По итогам каждого учебного года выпускаются электронные и бумажные сборники.

В рамках фестиваля для желающих проводится конкурс презентаций. Всем участникам конкурса высылаются специальные дипломы.

Удобный Личный кабинет участника фестиваля, возможность автоматического создания личного профессионального портфолио. В помощь участникам – квалифицированные сотрудники оргкомитета. Единственный в России образовательный сайт, имеющий службу поддержки в режиме on-line 7 дней в неделю.



Участвуйте в фестивале, размещайте свои работы,
получайте документы о публикации!

Фестиваль творческих и исследовательских работ учащихся «Портфолио ученика» project.1september.ru

Свидетельство о регистрации СМИ Эл. № ФС77-53211

Площадка для публикации работ учащихся, выполненных под руководством педагогов.

Всем ученикам и педагогам предоставляются документы о публикации. По итогам каждого учебного года выпускаются электронные и бумажные сборники.

В рамках фестиваля для желающих проводится конкурс проектных работ.

Все участники конкурса награждаются специальными дипломами.



Участвуйте вместе с учениками!



Первые шаги к “дополненной реальности”: Google Glass и другие

Что такое “дополненная реальность”

Д.Ю. Усенков,
Москва

▶ Чтобы понять, что такое “дополненная реальность”, нам потребуется вспомнить еще об одной компьютерной технологии — “виртуальной реальности”. Это сочетание аппаратных и программных средств, позволяющее воспроизвести графическое изображение окружающего пользователя созданного компьютером пространства, так что пользователь будет воспринимать его не менее реально, чем обычный, “настоящий” окружающий мир. Для этого используется целый ряд технических решений:

- **стереоскопическое изображение и стереозвук** — позволяют пользователю видеть и слышать

созданный компьютером “виртуальный мир” в объеме и с гораздо более полным охватом поля зрения, чем на обычном дисплее или даже на 3D-мониторе. Для этого используются специальные стереочки или шлем виртуальной реальности: два дисплея при помощи линзовой оптической системы воспроизводят соответствующие изображения для левого и правого глаз, образующие стереопару;

- **моделирование физики** — еще один элемент повышения реалистичности создаваемого компьютером изображения. В виртуальном мире все взаимодействия с его объектами должны происходить с учетом физических параметров этих объектов, причем все сказанное относится также и к самому пользователю — точ-

нее, к его “аватару” (персонажу, в образе которого он представлен в виртуальном мире). Правда, физические константы могут быть и изменены (например, сила тяжести в виртуальном мире может быть меньше либо, наоборот, больше обычной), но общие принципы взаимодействия с виртуальными объектами остаются в рамках физических законов;

- **интерактивность** — позволяет естественным образом менять положение “точки наблюдения” в виртуальном мире. В трехмерной компьютерной игре от первого лица (такой, например, как Doom или Quake) перемещение персонажа осуществляется мышью или нажатиями клавиш на клавиатуре и при этом компьютер “пересчитывает” и перерисовывает на экране дисплея, как должно выглядеть игровое пространство уже из новой точки наблюдения. В системе же виртуальной реальности в шлем непосредственно встроена гироскопическая система, датчики которой постоянно отслеживают повороты и наклоны головы пользователя и тут же перерисовывают видимую им картинку. В результате возникает естественное ощущение “осматривания” виртуального пространства;

- **сенсорная чувствительность** — реализуется при помощи целого набора специальных периферийных устройств — от перчаток, позволяющих компьютеру отслеживать только положение пальцев пользователя, или “жилеток”, отслеживающих положение его рук, до полного костюма (комбинезона) виртуальной реальности, передающего в компьютер полную информацию о позе пользователя в каждый конкретный момент, и до специальных устройств, отслеживающих положение тела пользователя в окружающем его пространстве. Такие сенсорные “элементы одежды” выполняют и обратную задачу — имитируют для пользователя тактильные (осязательные) ощущения прикосновения, надавливания, тепла или холода и т.д. Поэтому пользователь в виртуальном мире может, например, не просто поднять виртуальный камень, нагнувшись и схватывая изображение этого камня изображениями своих пальцев (которые двигаются синхронно с реальными), но и почувствовать его вес (как камень давит на ладонь), твердость (насколько “материал” камня сопротивляется надавливанию на него, — соответствующие сигналы и видимое изменение очертаний виртуального объекта формируются путем вычислений с учетом законов физики и сопромата) и пр.;

- **мультимедийность** — задействование по возможности всех органов чувств пользователя, их “переключение” с восприятия реального окружающего мира на восприятие сигналов с компьютера. Чем более полно пользователь окажется “подключен” к компьютеру, тем больше будет и ощущение реальности созданного компьютером виртуального мира. Отчасти при этом “включается в игру” и сам человеческий мозг, он многое сам, на основе всего имеющегося житейского опыта, “домыслива-

ет” по получаемым им “грубым” сигналам-ощущениям, предоставленным компьютером.

Таким образом, виртуальная реальность предполагает полное “переключение” пользователя с восприятия реального мира на восприятие компьютерной графики. А что если совместить и то и другое? Например, накладывать создаваемые компьютером изображения объектов на изображение, реально видимое пользователем? Вот такое совмещение восприятия реального мира с элементами компьютерного изображения и называют “дополненной реальностью”.

Что может дать такая технология? Вот наглядный пример. Предположим, вы приехали на научную конференцию в незнакомый вам город и не знаете дорогу с вокзала в гостиницу. Если вы используете оборудование “дополненной реальности”, то все будет выглядеть примерно так. Вы даете своему компьютеру (скажем, смартфону) команду включить режим навигации и сообщаете адрес конечной точки пути. Компьютер определяет ваше местоположение (в системе GPS или ГЛОНАСС), прокладывает маршрут, запрашивает текущую информацию о нем (например, о наличии автомобильных пробок), — все как в обычном автомобильном навигаторе. Отличие только в том, как проложенный маршрут визуализируется для вас. На обычном навигаторе это — линия на карте города с голосовыми сообщениями типа “через 100 метров повернуть направо”. А при пользовании системой “дополненной реальности”, если вы оглянетесь вокруг, то увидите начинающуюся у ваших ног “путеводную нить” в виде светящейся линии или в виде цепочки из флажков либо стрелочек. Видеть ее конечно же будете только вы (а кто-то из окружающих, пользующихся такой же системой “дополненной реальности”, будет в это время видеть свою собственную “путеводную нить” к другой, нужной ему цели). Вам останется только лишь идти вдоль нее, выполняя указания компьютера о том, на какой общественный транспорт следует садиться и на какой остановке выходить. Причем компьютер может сам (при помощи видеокамеры) “видеть” и распознавать номера автобусов, троллейбусов или трамваев и определить среди них нужный. В результате вы доберетесь до нужной вам гостиницы наиболее оптимальным в текущей ситуации маршрутом. Если, пока вы в пути, вам кто-нибудь позвонит, то вы увидите “призрачное” изображение вашего собеседника, сопровождающее вас в этой прогулке (и тоже видимое только вам). Если в толпе обнаружится кто-нибудь из ваших знакомых (образ которого занесен в вашу базу данных), ваш компьютер сможет узнать его и отметит его видимой только вам стрелкой-указателем с соответствующей подписью. А если вам будут демонстрировать на выставке какое-либо сложное устройство, система “дополненной реальности” запросит через компьютерную сеть и “наложит” по-

верх того, что вы видите, схему этого устройства с необходимыми пояснениями, которые помогут вам понять принципы ее работы.

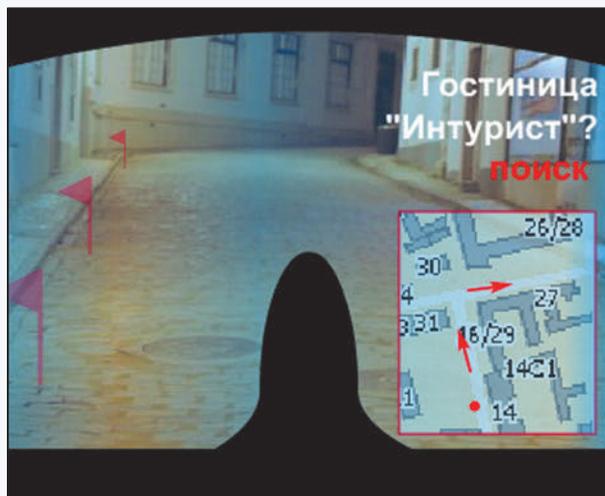


Рис. 1. Возможный вид интерфейса “дополненной реальности”

Пока это, конечно, немного фантастика ☺. Но некоторые элементы такой “дополненной реальности” уже реализованы при помощи очков Google Glass (о которых так много говорили и писали последние пару лет) или их аналогов. Например, пользователь очков может, где бы он ни находился — на улице, в транспорте, дома или на работе, — просматривать новости и электронную почту, делать фотографии и отправлять их в социальные сети прямо на ходу, причем руки останутся свободными.

“Дополненная реальность”: плюсы и минусы

При всех возможностях, предоставляемых технологией “дополненной реальности”, она несет в себе не только плюсы. Как и любая техническая новинка, она имеет и целый ряд недостатков, причем запредельно высокая цена за в общем-то пока довольно несовершенные устройства — это далеко не самый важный минус.

Главным недостатком таких систем является их потенциальная опасность для пользователя. Уже сегодня постоянно растет количество несчастных случаев и ДТП, происходящих из-за того, что пользователи чересчур увлекаются своими смартфонами или даже обычными плеерами и из-за этого не замечают автомобиль или поезд. А когда такие пользователи получают в свое распоряжение, например, возможность прямо в пути смотреть фильмы или общаться в социальной сети, риск “выпасть из реальности” и потерять за собой контроль в реальном мире окажется очень велик.

Другой существенный минус — возможность незаметно для окружающих вести фото- и видеосъемку. Конечно, для владельца очков это может быть полезной и удобной функцией, но далеко не все окружающие будут согласны с таким вторжением в

их частную жизнь — ведь фотографии и видео позже могут быть опубликованы владельцем очков где угодно. Правда, этот недостаток можно устранить. Так, в Японии уже разработан концепт очков, оборудованных инфракрасными светодиодами: они засвечивают лицо их владельца и защищают его от узнавания на фотографиях и видео, сделанных с помощью Google Glass.

По этой же причине подобные устройства могут быть отнесены к “шпионским”. Из-за этого очки Google Glass и их аналоги могут оказаться в некоторых странах вне закона; например, опасение возникновения подобных ситуаций имелось в России, Белоруссии и Украине, хотя ФСБ и СБУ уже официально заявили, что не видят оснований считать покупку очков Google Glass незаконной. Впрочем, и при разрешении ношения и использования таких устройств в масштабах страны владельцы частных заведений (ресторанов и в особенности кинотеатров, по понятной причине — из-за опасения видеопиратства) нередко сами устанавливают запрет на ношение подобных устройств на их территории. Так что, купив такие очки, их владелец еще не получает гарантию возможности их использования везде, где он захочет.

Что такое Google Glass и как оно работает

Как совместить создаваемое компьютером изображение с тем, что реально видит вокруг себя пользователь?

Один из способов, более простой в техническом исполнении, но менее удобный, — видеочки “виртуальной реальности” с встроенными видеокameraми. Изображение с левой и правой видеокamera транслируется, соответственно, на левый и правый дисплеи, что позволяет пользователю видеть окружающую реальность. А компьютер может накладывать на это видеоизображение свои графические изображения или текстовые надписи. Именно так была реализована одна из самых первых попыток “дополненной реальности” — студенческий проект, позволяющий играть в Doom на улицах обычного города ☺. Неудобство такого способа очевидно: та самая изоляция от окружающего реального мира, которая была преимуществом виртуальной реальности, затрудняет ориентацию в реальном мире.

По этой причине создатели Google Glass использовали другой способ проекции компьютерного изображения — прямую проекцию на сетчатку глаза пользователя, что позволяет ему максимально беспрепятственно видеть все окружающее.

Хотя Google Glass называют очками, на самом деле это не совсем очки. Конструктивно это — наголовная консоль, которая может использоваться вместе с очками или без них. Google Glass иногда называют “гарнитурой” (поскольку это “периферийное устройство” для смартфонов на базе Android). Можно считать

Google Glass и “нателным компьютером”: в них имеются память и встроенный процессор, работающий на базе ОС Android. Мы же будем использовать для Google Glass название “наголовный дисплей”.



Рис. 2. Google Glass в сочетании с обычными очками (фото с официального сайта Google Glass)

Конструктивно Google Glass представляет собой этакий “нимб” с двумя дужками. Левая дужка — несущая, ее назначение — фиксировать Google Glass на голове пользователя. Правая же дужка — основная. В ней встроены аккумулятор, микрофон, видеокамера, динамик (передающий звук не напрямую у ухо, как наушники, а вибрацией через кости черепа) и вся электронная начинка, а главное — компактный проектор в виде полупрозрачной призмы.

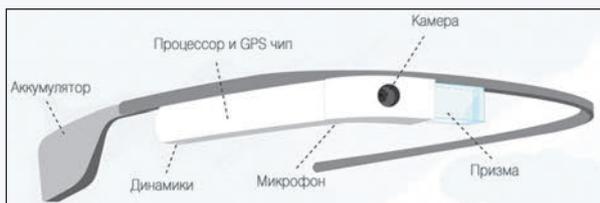


Рис. 3. Конструкция Google Glass

Компактный проектор располагается у края правого глаза и при помощи небольшой призмы проецирует изображение с разрешением 640×360 в виде маленького прямоугольника на сетчатку глаза в районе ямки — участка наиболее четкого зрения.

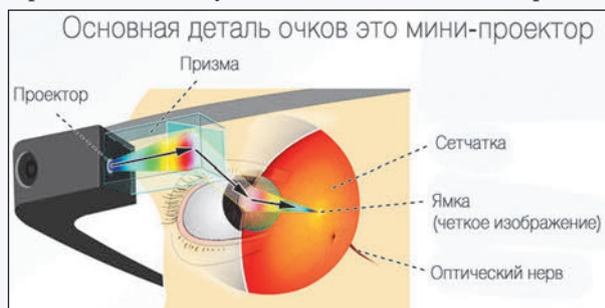


Рис. 4. Конструкция проектора Google Glass

Изображение, генерируемое проектором, накладывается в виде полупрозрачного графического слоя на видимое пользователем реальное изображение.

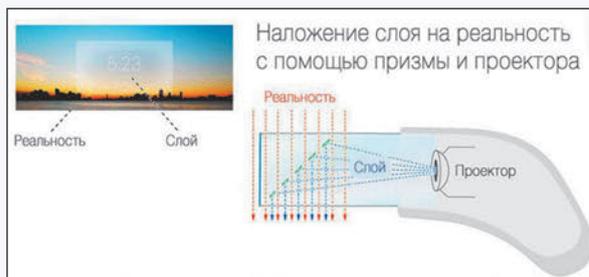


Рис. 5. Наложение проецируемого изображения на реальное

При этом, наклоня призму (приближая ее к глазу или удаляя от него), можно фокусировать создаваемое компьютером изображение на сетчатку.

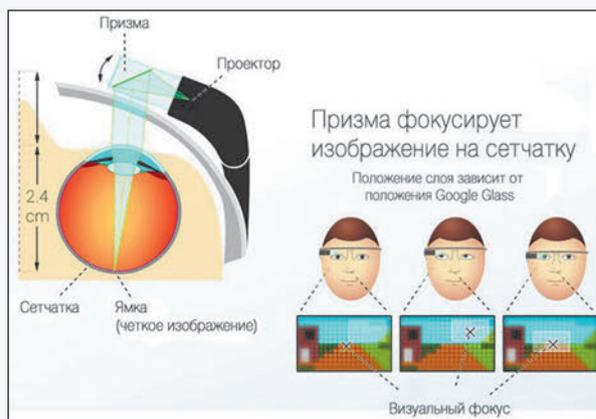


Рис. 6. Фокусировка проецируемого изображения

Если же пользователь носит обычные (“диоптрийные”) очки, то проецируемое изображение можно намеренно исказить — так, чтобы учесть преломляющий эффект линзы очков.

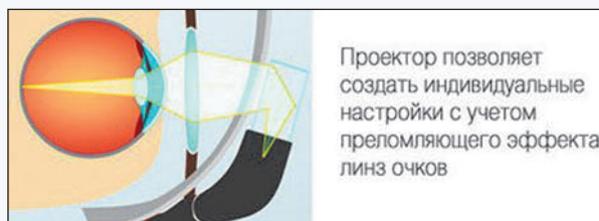


Рис. 7. Проекция изображения через “диоптрийные” очки

В результате пользователь видит перед собой на фоне окружающей реальности примерно следующую картину:



Рис. 8. Визуальный эффект при работе Google Glass

Видеокамера, закрепленная на Google Glass спеледи, позволяет снимать видео и фотографии с раз-

решением 5 мегапикселей, а также реализовать технологию распознавания изображений (штрих-кодов, лиц и т.д.).

Микрофон позволяет осуществлять взаимодействие пользователя с Google Glass посредством голосовых команд. При этом каждая такая команда должна предваряться “сигнальной” фразой “Ok, Glass” (немного знакомо, правда? ☺), после которой следует просьба выполнить те или иные действия. Например, фраза “ok, glass, take a picture” — это команда сделать фотографию (хотя для этого имеется и специальная кнопка, единственная на всем устройстве); фраза “ok, glass, give directions to [место назначения]” включает навигатор с указанием требуемого места, к которому надо проложить маршрут; фраза “ok, glass, say [текст] in [язык]” активизирует переводчик; фраза “ok, glass, google [поисковый запрос]” запускает поисковый сервис и т.д. Кроме того, через микрофон можно надиктовывать текст для его компьютерного набора.

Наконец, хотя основным “каналом общения” с Google Glass является голосовой, в дужке очков расположен тачпад, позволяющий распознавать жесты.

Все это техническое оснащение предназначено для реализации трех основных функций: “дополненной реальности” (такой, как описано в начале этой статьи), мобильной связи и доступа в сеть Интернет и “видеодневника” (возможности вести фото- и видеозапись событий своей личной жизни). Однако первые версии Google Glass полноценно реализуют только “видеодневник” (подарок любителям “Инстаграмма”) и лишь частично — “дополненную реальность” и коммуникации. Их более полная реализация предполагается в последующих версиях. Например, компания Google запатентовала концепцию жестового интерфейса для Google Glass: с помощью определенных жестов руками, которые Google Glass будет распознавать благодаря своей видеокамере, можно будет запускать те или иные действия — например, сложив ладони сердечком, поставить “лайк” в соцсети.

Существующие модификации Google Glass можно считать скорее “экспериментальными”, выпущенными в продажу в расчете на отработку имеющегося функционала, изучение спроса и окупаемость затрат на разработку последующих более “продвинутых” моделей. Впервые информация о Google Glass появилась в газете “New York Times” в феврале 2012 года, а их тестирование началось в апреле того же 2012 года. Годом позже появилась вторая версия очков Google Glass 2.0 (с встроенным мононаушником), а в мае 2014 года очки Google Glass наконец поступили в открытую продажу по цене 1500 долларов. Правда, в Россию официальные поставки не предполагались: очки Google Glass попадали в нашу страну неофици-

ально и стоили (по курсу доллара на начало осени 2014 г.) от 100 до 120 тысяч рублей. (Любопытно, что по сообщению “Вести.Ру” — <http://hitech.vesti.ru> со ссылкой на аналитическую компанию IHS, оценивающую себестоимость популярных гаджетов, и на экспертов, определивших цену каждого компонента Google Glass, себестоимость аппаратной части очков без учета их сборки и затрат на разработку составляет всего 152 доллара 47 центов.)

Учитывая такой уровень цен и ограничения на “открытую” продажу, трудно было предполагать широкую популярность Google Glass (можно вспомнить про похожее устройство — очки **Dyson Halo**, которое разрабатывалось в компании Dyson с 2001 года и тоже имело голосовое управление и речевой модуль для чтения входящих сообщений, но так и не дошло до массового выпуска). Итог всей истории подвела недавно появившаяся информация о том, что компания Google объявила о прекращении с 19 января 2015 года продаж нынешней экспериментальной версии Google Glass. Теперь все усилия разработчиков будут нацелены на доработку изделия в рамках отдельного исследовательского проекта, — хотя сроки и тем более цены появления в продаже уже доработанной коммерческой версии пока не объявлены.

Все это открывает “зеленый свет” другим разработчикам проектов “дополненной реальности”, изделия которых могут оказаться более дешевыми, более удобными и скорее всего предоставляющими больше возможностей (ведь по проторенному пути идти все-таки намного легче).

Аналоги Google Glass

Кроме уже упомянутого концепта Dyson Halo, так и оставшегося на стадии экспериментальной разработки, уже существует целый ряд конструкций, похожих на Google Glass. Правда, далеко не все они являются их аналогом, например, **Neurocam** для iPhone (наголовная фотовидеокамера, управляемая, по задумке ее разработчиков, электромагнитными сигналами мозга и позволяющая делать снимки при помощи “мысленных” команд) позволяет только выполнять съемку. А интерактивный шлем **Ride:HUD** реализует элементы “дополненной реальности” (вывод информации о погоде, сообщений в твиттере или карт GPS), но, судя по имеющейся информации о нем, остальной функционал Google Glass в этом шлеме не реализован. Более или менее полноценными аналогами Google Glass можно считать, пожалуй, только китайский **Baidu Eye**, тайваньский **SiMEye Smart Glass**, **Smart Glasses M100** компании Vuzix, проект **Microsoft Glass Project** и особенно — очень интересную разработку **Microsoft HoloLens**, которая заслуживает особого рассмотрения.

Новый проект по созданию пользовательской системы “дополненной реальности” под названием **Windows Holographic** и очки “дополненной реальности” **HoloLens** были представлены публике во время пресс-конференции, посвященной Windows 10. (Руководителем этого проекта, кстати, является Алекс Кимпан — тот самый, который ранее разработал контроллер Kinect.) Специалисты Microsoft называют этот проект “новым поколением ПК”: Windows Holographic позволит пользователю самому создавать для себя мир “дополненной реальности” и работать с виртуальными объектами просто голыми руками, без каких-либо дополнительных перчаток или приспособлений. Нужны будут только специальные очки Microsoft HoloLens.



Рис. 9. Внешний вид очков Microsoft HoloLens (кадр из презентации Microsoft)

Система Windows Holographic позволит создать вокруг себя целый виртуальный мир, интегрированный в реальность, в которой находится пользователь, а затем — видеть, слышать и взаимодействовать с этим миром. Например, пользователь сможет завести себе виртуального питомца, который будет жить не на экране ПК, а рядом, в реальной квартире. Можно будет перемещаться внутри виртуального мира игры Minecraft или разговаривать с голограммами собеседников, как в “Звездных войнах”. Можно будет вращать виртуальный глобус на реальном столе или конструировать что-то при помощи виртуального конструктора, существующего только в памяти ПК. (Увидеть, как будет работать “дополненная реальность” от Microsoft, можно на двух роликах, размещенных на YouTube:

<http://www.youtube.com/watch?v=aThCrOPsyuA> и <http://www.youtube.com/watch?v=vsKMi1377cU>.)



Рис. 10. “Виртуальное в реальном” глазами пользователя очков HoloLens



Рис. 11. “Виртуальные информационные панели” станут повседневной реальностью



Рис. 12. Конструируем на экране ПК в программе САПР — тут же видим действительно объемный результат: похоже, в ближайшем будущем работа инженера будет выглядеть именно так

Последний пример как раз и был продемонстрирован на конференции. Представитель Microsoft, надев очки Microsoft HoloLens, сконструировала дрон, используя детали из висящего в воздухе виртуального конструктора. При этом детали она брала голыми руками, без каких-либо перчаток виртуальной реальности. Чтобы взять деталь, использовался специальный жест — указательный палец сначала надо было направить вертикально, а затем резко согнуть, как будто нажимая на невидимый выключатель. Такое же действие позволяет отпустить взятый объект. Собираемый виртуальный дрон при этом также висел в воздухе (съемка сюжета велась тоже через очки HoloLens). А затем Алекс Кимпан вынес на сцену уже реальное устройство, собранное по этому эскизу.



Рис. 13а. Конструирование дрона из деталей виртуального конструктора



Рис. 13б. Реальный дрон, созданный по этому эскизу (слева)

Очки Microsoft HoloLens — это главный элемент платформы Windows Holographic. Это — полностью самостоятельное беспроводное устройство, не требующее подключения к ПК. Очки HoloLens содержат экран, процессор, графический ускоритель и отдельный модуль “Holographic Processing Unit” (HPU), динамики окружающего звучания и все необходимые датчики. Модуль HPU при этом как раз и отвечает за распознавание жестов, голоса и окружающего пространства, в том числе за обработку информации, поступающей с датчиков, следящих за руками пользователя.

Для создания “голограмм” Microsoft разработала и специальное программное решение — **HoloStudio**. Созданные в нем виртуальные объекты затем можно “оживить” с помощью соответствующего программного обеспечения или даже напечатать на 3D-принтере. Предполагается, что очки HoloLens будут поддерживать универсальные приложения с единым кодом, написанные сразу для всех разновидностей компьютерных устройств (та же самая концепция, что и в Windows 10). Кроме того, Microsoft планирует добавить поддержку сторонних устройств, в том числе Oculus Rift и Magic Leap.

В Microsoft предполагают, что очки Microsoft HoloLens станут доступны пользователям одновременно с Windows 10, т.е. до конца 2015 года. Однако стоимость устройства пока не озвучивается, хотя глава компании Microsoft Сатья Наделла заявил, что хотел бы, чтобы возможность купить HoloLens была не только у коммерческих организаций, но и у обычных пользователей.

Google Glass — своими руками?

Можно ждать появления в продаже Google Glass или его аналогов и платить за них большие деньги, а можно и попытаться самому создать не-

что подобное. Именно так рассуждал специалист по разработке приложений в области систем “дополненной реальности” из Оксфорда Уилл Пауэлл. В его конструкции использованы видеоочки Vuzix, две веб-камеры, а также обычная гарнитура с микрофоном и программа распознавания речи Dragon NaturallySpeaking. Программное обеспечение создано на базе Adobe AIR, интерфейс близок к используемому в Google Glass. Очки Пауэлла также способны делать фотографии по команде голосом и публиковать их в Интернете на Google+. Правда, в отличие от Google Glass очки Пауэлла непрозрачны — дисплеи полностью закрывают поле зрения глаз пользователя и на них подается стереоизображение окружающей реальности с двух видеокамер, на которое накладывается компьютерная графика.

Правда, обо всех секретах своей разработки Пауэлл не рассказывает. Так что желающим повторить его опыт придется делать все самостоятельно, “с нуля”.



Рис. 14. Уилл Пауэлл, его самодельные очки “дополненной реальности” (справа) и результат их работы (слева)

Есть, кстати, и еще один проект, в рамках которого ведется разработка очков — аналогов Google Glass по цене... всего лишь около 20 долларов. Эта разработка носит название Faux.Glass. Внешне такие очки почти не отличаются от Google Glass — добавлена только линеечка светодиодов для большей эффектности.



Рис. 15. Очки Faux.Glass за 20 долларов

Вот только ничего другого, кроме мигания светодиодами, очки Faux.Glass не умеют и предназначены для любителей похвастаться. Ведь со стороны владелец таких очков будет смотреться очень “круп-

то”. Особенно если будет делать вид, что реально использует “дополненную реальность” ☺.

Немного юмора напоследок

Конечно, описанные выше “очки” Faux.Glass можно, скорее, отнести к “приколам”, посвященным Google Glass и их аналогам. Если поискать в Интернете, то можно откопать и немало других шуток на эту тему. В качестве примера можно привести лишь несколько юмористических картинок, высмеивающих как саму идею “дополненной реальности”, так и возможные последствия использования очков с проекционным дисплеем, заставляющим пользователя “косить глазом” на формируемое Google Glass компьютерное изображение.



Использованные источники информации

1. “Википедия” (<https://ru.wikipedia.org/?oldid=67104079>).
2. Официальный сайт Google Glass (<http://www.google.com/glass/start>).
3. https://hi-tech.mail.ru/review/google_glass_2_0-rev.html.
4. <http://iblogis.ru/semi-analogov-google-glass.html>.
5. <http://feme.ua/your-life/interesnoe/google-glass-mozhno-budet-upravlyat-s-pomoshchyu-zhestov.html>.
6. <http://android-tornado.ru/news/317-google-glassesbudut-ispolzovat-zhesty-ruk.html>.
7. <http://hitech.vesti.ru>.
8. <http://www.3dnews.ru/821716>.
9. <http://www.google-glass.com.ru/content/analog-google-glass-ot-microsoft>.
10. <http://subscribe.ru/digest/inet/newness/n1792919424.html>.
11. <http://subscribe.ru/digest/inet/newness/n1792909714.html>.
12. <http://www.computerra.ru/72994/atheer-labs-glasses>.
13. <http://hitech.vesti.ru/news/view/id/1866>.
14. <http://digisteam.ru/gadgets/google-glass>.
15. <http://habrahabr.ru/post/141855>.
16. <http://www.gadgetblog.ru/faux-glass-analog-google-glass-za-20-dollarov>.



Педагогический университет
«**Первое сентября**»

ДИСТАНЦИОННЫЕ КУРСЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

(с учетом требований ФГОС)

Ведется прием заявок на первый поток 2015/16 учебного года

образовательные программы:

- НОРМАТИВНЫЙ СРОК ОСВОЕНИЯ – 108 УЧЕБНЫХ ЧАСОВ
Стоимость – 4990 руб.

- НОРМАТИВНЫЙ СРОК ОСВОЕНИЯ – 72 УЧЕБНЫХ ЧАСА
Стоимость – от 3990 руб.

По окончании выдается удостоверение о повышении квалификации
установленного образца

Перечень курсов и подробности – на сайте edu.1september.ru

Пожалуйста, обратите внимание:

заявки на обучение подаются только из Личного кабинета,
который можно открыть на любом сайте портала www.1september.ru



Общероссийский проект
Школа цифрового века

**6 тысяч рублей от школы
за весь 2015/16 учебный год
независимо от количества учителей
в образовательной организации**

Каждому учителю:

- 24 предметных ежемесячных журнала
- десятки курсов повышения квалификации

**Не забудьте принять
или продлить участие!**

Подробности и форма заявки на сайте:

digital.1september.ru



ШКОЛА ПРОГРАММИРОВАНИЯ

Фрактал из окружностей

В данной статье мы опишем методику получения изображения, показанного на рис. 1.

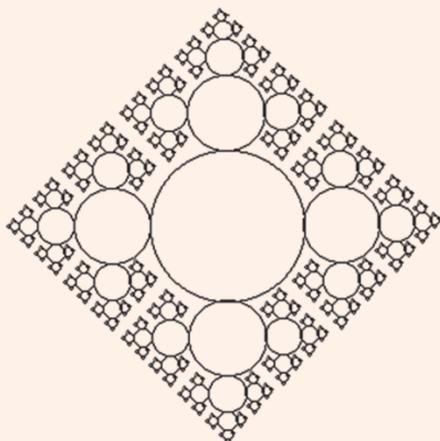


Рис. 1

Но начнем мы с более простой картинки, представленной на рис. 2. Почему более простой — станет ясно ниже, а в чем отличие двух изображений, предлагаем читателям определить самостоятельно.

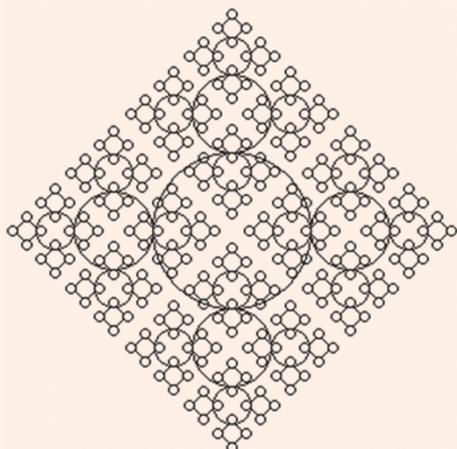


Рис. 2

Анализ рис. 2 показывает, что на нем изображена большая окружность, к которой с четырех сторон примыкают четыре меньшие окружности, каждой из которых также касаются еще четыре окружности меньшего размера и т.д. Если какую-то окружность условно назвать “планетой”, а примыкающие к ней — “спутниками”, то очевидно, что каждый спутник, в свою очередь, может рассматриваться как планета с соответствующими спутниками. Это наводит на мысль о возможности использования такого понятия, как рекурсия [1–2]. Поэтому если составить процедуру, с помощью которой можно изобразить на экране некоторую окружность-планету с рядом окружностей-спутников, а для рисования спутников использовать эту же процедуру, но с другими параметрами — координатами, радиусами и т.п. (то есть применить рекурсию), то можно получить требуемое изображение.

В приведенной ниже процедуре *Окружности* использованы следующие основные величины:

1) ее параметры:

— x и y — координаты окружности-планеты;

— *радиус* — ее радиус;

— n — величина, указывающая на “уровень” расположения спутников и одновременно — определяющая необходимость дальнейшего рекурсивного использования процедуры *Окружности* (напомним, что необходимо так оформлять процедуры (функции), чтобы рекурсивные вызовы осуществлялись по условию, которое когда-то станет ложным). В нашей процедуре величина n будет уменьшаться на 1, а в тело процедуры включено условие, что ее команды должны выполняться только при $n > 0$.

— $k < 1$ — коэффициент уменьшения размера окружностей-спутников по отношению к размеру окружности-планеты;

2) ее локальные переменные параметры:

— x_1 и y_1 — координаты окружности-спутника;

— *рад* — ее радиус.

Итак, процедура *Окружности* на школьном алгоритмическом языке:

```

алг Окружности(арг цел x, y, радиус,
                n, вещ k)
нач цел i, x1, y1, рад
если n > 0
то
    |Рисуем окружность-"планету"
    поз(x, y)
    окружность(радиус)
    |Рисуем окружности-"спутники"
    |и для каждой из них
    |рекурсивно вызываем
    |процедуру Окружности
    |Радиус спутников
    рад := радиус * k
    |Правая окружность-"спутник"
    |Рассчитываем координаты ее центра
    x1 := x + радиус + int(рад)
    y1 := y
    |Рекурсивный вызов
    Окружности(x1, y1, рад, n - 1, k)
    |Нижняя
    x1 := x
    y1 := y + радиус + int(рад)
    Окружности(x1, y1, рад, n - 1, k)
    |Левая
    x1 := x - радиус - int(рад)
    y1 := y
    Окружности(x1, y1, рад, n - 1, k)
    |Верхняя
    x1 := x
    y1 := y - радиус - int(рад)
    Окружности(x1, y1, рад, n - 1, k)

```

все

кон

Напоминаем, что на экране координатная ось *У* направлена вниз.

Если вспомнить значения синуса и косинуса углов 90, 180, 270 и 360 градусов, то вместо четырех фрагментов, изображающих окружности-"спутники" (с последующим рекурсивным вызовом процедуры *Окружности*), можно применить оператор цикла с параметром, в котором для расчета координат использованы значения синуса и косинуса

```

нц для i от 1 до 4
    x1 := x + int((радиус + рад) *
                 cos(6.28/4 * i))
    y1 := y + int((радиус + рад) *
                 sin(6.28/4 * i))
    Окружности(x1, y1, рад, n - 1, k)

```

кц

Обратим внимание на то, что при $i = 1$ рисуется нижний "спутник", при $i = 2$ — левый, при $i = 3$ — верхний, при $i = 4$ — правый (убедитесь в этом!). Это обстоятельство пригодится нам в будущем.

Основная часть программы в любом случае может быть оформлена так:

```

алг Фрактал_из_окружностей
нач цел радиус, n, вещ k

```

```

    |Устанавливаем графический режим
    |работы экрана
    ...
    |Задаем начальное значение
    |величин радиус
    радиус := 50
    |значение величин k
    k := 0.5
    |и n
    n := 5
    |Вызываем процедуру Окружности
    |для рисования самой большой планеты
    |и ее спутников
    Окружности(int(максX/2), int(максY/2),
                радиус, n, k)

```

кон

Примечания

1. *максX* и *максY* — соответственно, максимальное значение координат x и y в выбранном режиме работы экрана.

2. Функция *int* возвращает целую часть ее вещественного аргумента.

А как же получить изображение, показанное на рис. 1? Можно увидеть, что оно отличается от рис. 2 тем, что "спутники", расположенные внутри других окружностей-"планет", не нарисованы. Чтобы добиться этого, сначала изменим нашу программу следующим образом.

Центральную (самую большую) окружность и ее четыре "спутника" будем получать отдельно, а только после этого для каждой из этих четырех окружностей будем рисовать ее "спутники":

```

алг Фрактал_из_окружностей
нач цел радиус, n, вещ k
    |Устанавливаем графический режим
    |работы экрана
    видео(17)
    |Задаем начальное значение величин
    |радиус
    радиус := 50
    |значение величин k
    k := 0.5
    |и n
    n := 41
    |Рисуем центральную окружность
    поз(int(максX/2), int(максY/2))
    окружность(радиус)
    |и ее спутники
    рад := радиус * k
нц для i от 1 до 4
    x1 := int(максX/2) +
           int((радиус + рад) *
              cos(6.28/4 * i))
    y1 := int(максX/2) +
           int((радиус + рад) *
              sin(6.28/4 * i))
    поз(x1, y1)
    окружность(рад)
    |Для изображения новых "спутников"
    |вызываем процедуру Спутники

```

```
Спутники(x1, y1, рад, n, k)
```

```
кц
```

```
кон
```

Видно, что появилась новая процедура — с именем *Спутники*. Она изображает четыре “спутника”, и для каждого из них происходит рекурсивное использование этой же процедуры.

```
алг Спутники(арг цел x, y, радиус, n, вещь k)
```

```
нач цел i, x1, y1, рад
```

```
если n > 0
```

```
то
```

```
|Определяем радиус "спутников"
```

```
рад := радиус * k
```

```
|Для каждого "спутника"
```

```
нц для i от 1 до 4
```

```
|Определяем его координаты
```

```
x1 := x + int((радиус + рад) *  
cos(6.28/4 * i))
```

```
y1 := y + int((радиус + рад) *  
sin(6.28/4 * i))
```

```
|Рисуем "спутник"
```

```
поз(x1, y1)
```

```
окружность(рад)
```

```
|Для изображения новых "спутников"
```

```
|рекурсивно вызываем эту же
```

```
|процедуру
```

```
Спутники(x1, y1, рад, n - 1)
```

```
кц
```

```
все
```

```
кон
```

Теперь мы уже можем получить изображение, показанное на рис. 1. Как уже отмечалось, на нем “спутники”, расположенные внутри других окружностей-“планет”, не нарисованы. Поскольку какие именно окружности-“спутники” не нужно рисовать, зависит от того, какой является окружность-“планета” (правой, левой, верхней или нижней), то введем в процедуру *Спутники* еще один параметр, определяющий эту особенность “планеты”. Дадим ему имя — *номер* (при *номер* = 1 “планета” — нижняя, при *номер* = 2 — левая, при *номер* = 3 — верхняя, при *номер* = 4 — правая; см. выше). В процедуре *Спутники* в зависимости от значения параметра *номер* некоторые окружности-“спутники” изображаться не должны:

```
алг Спутники(арг цел x, y, радиус,  
номер, n)
```

```
нач цел i, x1, y1, рад
```

```
если n > 0
```

```
то
```

```
рад := радиус * k
```

```
выбор |В зависимости от  
|значения номер
```

```
при номер = 1:
```

¹ Значение параметра *n* задается на единицу меньше, чем ранее, так как центральная окружность и ее “спутники” уже нарисованы.

```
|Не рисуется верхняя  
|окружность-"спутник"
```

```
нц для i от 1 до 2
```

```
x1 := x + int((радиус + рад) *  
cos(6.28/4 * i))
```

```
y1 := y + int((радиус + рад) *  
sin(6.28/4 * i))
```

```
поз(x1, y1)
```

```
окружность(рад)
```

```
Спутники(x1, y1, рад, i, n - 1)
```

```
кц
```

```
i := 4
```

```
x1 := x + int((радиус + рад) *  
cos(6.28/4 * i))
```

```
y1 := y + int((радиус + рад) *  
sin(6.28/4 * i))
```

```
поз(x1, y1)
```

```
окружность(рад)
```

```
Спутники(x1, y1, рад, i, n - 1)
```

```
при номер = 2:
```

```
|Не рисуется правая  
|окружность-"спутник"
```

```
нц для i от 1 до 3
```

```
x1 := x + int((радиус + рад) *  
cos(6.28/4 * i))
```

```
y1 := y + int((радиус + рад) *  
sin(6.28/4 * i))
```

```
поз(x1, y1)
```

```
окружность(рад)
```

```
Спутники(x1, y1, рад, i, n - 1)
```

```
кц
```

```
при номер = 3:
```

```
|Не рисуется нижняя
```

```
|окружность-"спутник"
```

```
нц для i от 2 до 4
```

```
x1 := x + int((радиус + рад) *  
cos(6.28/4 * i))
```

```
y1 := y + int((радиус + рад) *  
sin(6.28/4 * i))
```

```
поз(x1, y1)
```

```
окружность(рад)
```

```
Спутники(x1, y1, рад, i, n - 1)
```

```
кц
```

```
при номер = 4:
```

```
|Не рисуется левая
```

```
|окружность-"спутник"
```

```
нц для i от 3 до 4
```

```
x1 := x + int((радиус + рад) *  
cos(6.28/4 * i))
```

```
y1 := y + int((радиус + рад) *  
sin(6.28/4 * i))
```

```
поз(x1, y1)
```

```
окружность(рад)
```

```
Спутники(x1, y1, рад, i, n - 1)
```

```
кц
```

```
i := 1
```

```
x1 := x + int((радиус + рад) *  
cos(6.28/4 * i))
```

```
y1 := y + int((радиус + рад) *  
sin(6.28/4 * i))
```

```
поз(x1, y1)
```

```
окружность(рад)
```

```
Спутники(x1, y1, рад, i, n - 1)
```

```
все
```

```
все
```

```
кон
```

В основной части программы вызов процедуры *Спутники* должен быть оформлен с учетом изменения ее (процедуры) описания:

Спутники($x_1, y_1, \text{рад}, i, n$)

Задание для самостоятельной работы

На языке программирования, который вы изучаете, разработайте программу, с помощью которой можно получить изображение, пока-

занное на рис. 1. Все приславшие правильную программу будут награждены дипломами.

Литература

1. Рекурсия — эффективно, но не всегда эффективно. / “В мир информатики” № 188–189 (“Информатика” № 7–8/2014).

2. *Златопольский Д.М.* Программирование: типовые задачи, алгоритмы, методы. М.: Бином. Лаборатория знаний, 2007.

ЗАДАЧНИК

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

Задача “Размеры футбольного поля”

Напомним условие: “На планете τ -Кита футбольное поле имеет размеры, как и на Земле, — 105×68 м. Но τ -китяне записывают эти размеры как 126×75 м. Почему?”

Ответ: потому что они используют не десятичную систему счисления, а девятеричную.

Обоснование

Обозначим основание используемой τ -китянами системы счисления — p . Используем так называемую “развернутую запись” числа 75:

$$75_p = 7 \times p + 5 = 68,$$

откуда $p = 9$.

Проверка для девятеричного числа 126 подтверждает правильность полученного результата.

Правильные ответы представили:

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

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

— Жакова Светлана, средняя школа поселка Озёры Красноярского края, учитель **Филипченко И.С.**;

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

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

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

— Прокопенко Павел, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Рыбалко Иван, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Таран Алина, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н, учитель **Краснёнкова Л.А.**;

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

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

— Шумкина Евгения, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Япрынцева Ольга, Республика Башкортостан, г. Стерлитамак, школа № 24, учитель **Орлова Е.В.**

Задача “Четыре талантливые девочки”

Напомним, что по ряду фактов следовало установить, на каком музыкальном инструменте играет и какой иностранный язык знает каждая из четырех девочек.

Ответ: Марина — на рояле и английский язык, Иржина — на скрипке и французский язык, Карина — на арфе и итальянский язык, Ирина — на виолончели и немецкий язык.

Правильные ответы представили:

— Абдувахидова Алина, Абдувахидова Софья, Милушкин Дмитрий, Хозин Марат и Чуб Алексей, Владимирская обл., г. Струнино, школа № 11, учителя **Волков Ю.П.** и **Волкова Т.П.**;

— Алексеева Анастасия и Цаплина Анна, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Бабкина Полина, Белоносова Екатерина, Горбунова Ксения, Нилова Екатерина, Маслий Елизавета и Петрова Полина, гимназия г. Шелехова, Иркутская обл., учитель **Водальчук С.А.**;

— Деревянченко Дарья, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Евграфова Ксения, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Иванова Виолетта и Левченко Ирина, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Иванова Ксения и Мухина Светлана, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Кляузер Адам, г. Орел, лицей № 4 им. Героя Советского Союза Г.Б. Злотина, учитель **Чапкевич И.М.**;

— Медведева Анастасия, Владимирская обл., г. Струнино, школа № 11, учитель **Волкова Т.П.**;

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

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

— Приходько Мария, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Хвойновский Вадим и Чернова Ксения, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Япрынцева Ольга, Республика Башкортостан, г. Стерлитамак, школа № 24, учитель **Орлова Е.В.**;

— Яснова Дарья, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**

Задача “Обманутый хозяин”

Напомним, что в задаче речь шла о шкафе в форме квадрата с девятью отделениями, в которых хозяин расставил 60 бутылок масла так, как показано на схеме:

6	9	6
9		9
6	9	6

Слуга подметил, что хозяин с целью экономии времени проверяет число бутылок, только считая бутылки по сторонам квадрата и следя за тем, чтобы на каждой стороне квадрата было по 21 бутылке. Учитывая это, слуга смог унести сначала четыре бутылки, а остальные расставил так, что вновь получилось по 21 на каждой стороне. Так он повторял до тех пор, пока было возможно. Требовалось определить, сколько раз слуга брал бутылки и сколько всего бутылок он унес.

Решение

Слуга брал себе по бутылке из каждого отделения и из тех же отделений, чтобы обмануть хозяина, после каждого воровства прибавлял по бутылке в угловые отделения. Так он брал четыре раза по четыре бутылки, а всего унес 16 бутылок. Все это видно из рисунка.

7	7	7
7		7
7	7	7

Первая кража

8	5	8
5		5
8	5	8

Вторая кража

9	3	9
3		3
9	3	9

Третья кража

10	1	10
1		1
10	1	10

Четвертая кража

Слуга мог расставить бутылки и другими способами. Но всегда в первом и третьем столбцах квадрата он должен был бы оставлять по 21 бутылке и

потому не мог бы унести более $60 - 2 \times 21 = 18$ бутылок, то есть совершить более четырех краж.

Правильные ответы представили:

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

— Дегтярь Анатолий и Новиченко Владимир, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Лукашин Станислав, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Медведева Анастасия, Владимирская обл., г. Струнино, школа № 11, учитель **Волкова Т.П.**;

— Мехмедов Арсен, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Милушкин Дмитрий и Хозин Марат, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

— Омарова Гульнара и Хромченкова Елизавета, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

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

— Сергеева Анна, г. Ростов-на-Дону, лицей № 56, учитель **Назаренко С.Н.**;

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

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

— Хвойновский Вадим и Чернова Ксения, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Шумкина Евгения, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Япрынцева Ольга, Республика Башкортостан, г. Стерлитамак, школа № 24, учитель **Орлова Е.В.**

Задача “Молочник и трое ребят”

Напомним, что следовало определить, как, имея две полных фляги объемом 50 и 40 литров, получить в бидонах объемом 3, 4 и 5 литра по 2 литра молока.

Задача решается за 10 операций переливания (в ряде ответов приведено большее число операций).

Ответы представили:

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

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

— Ким Леонид и Литвинов Андрей, г. Смоленск, школа № 29, учитель **Родикова Р.Д.**;

— Лукашин Станислав и Царев Иван, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Новиченко Владимир, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

— Хвойновский Вадим и Чернова Ксения, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

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

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

Задача “Какая буква спрятана? ☺”

Напомним, что требовалось определить, какая буква заменена вопросительным знаком в двойном буквенном неравенстве:

$$Д + Б + В + Ж + К < Р < Д + Б + В + Ж + К + ?$$

Ответ: буква М.

Обоснование

В одной известной сказке Дедка, Бабка, Внучка, Жучка и Кошка не смогли вытянуть Репку, и только с помощью Мышки это удалось сделать.

Ответы прислали:

— Абдувахидова Алина, Абдувахидова Софья, Милушкин Дмитрий и Чуб Алексей, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Алимова Алина, Степанов Сергей и Трофимова Мария, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Байбуза Дарья, Горелкина Лариса, Кузнецов Семен, Михайлова Алена и Репина Софья, средняя школа села Ириновка, Новобурасский р-н Саратовской обл., учитель **Брунов А.С.**;

— Волкова Валерия и Цаплина Анна, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Евграфова Ксения и Чуйкова Наталья, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Иванова Ксения и Мухина Светлана, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Ким Леонид и Литвинов Андрей, г. Смоленск, школа № 29, учитель **Родикова Р.Д.**;

— Коробов Сергей, Марков Алексей и Яснов Федор, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Медведева Анастасия, Владимирская обл., г. Струнино, школа № 11, учитель **Волкова Т.П.**;

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

— Приходько Мария, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Хвойновский Вадим и Чернова Ксения, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**

Заметим, что приведенный рядом читателей в принципе правильный ответ “Р” не учитывает наличие смайлика в названии задачи.

Задача “Получить 24”

Напомним, что требовалось переставить цифры 1, 3, 4 и 6 в любом порядке и расставить между какими угодно из них знаки арифметических действий и скобки так, чтобы получилось число 24.

Решения:

1) со “склежкой” цифр в многозначное число:

$$(14 - 6) \times 3;$$

2) без “склежки” цифр:

$$6 : (1 - 3 : 4).$$

Имеются и другие варианты.

Ответы прислали:

— Антипов Анатолий, Баков Василий и Кубко Виктория, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Байбуза Дарья, Горелкина Лариса и Кузнецов Семен, средняя школа села Ириновка, Новобурасский р-н Саратовской обл., учитель **Брунов А.С.**;

— Белоносова Екатерина, гимназия г. Шелехова, Иркутская обл., учитель **Водальчук С.А.**;

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

— Литвинов Андрей, г. Смоленск, школа № 29, учитель **Родикова Р.Д.**;

— Лукашин Станислав и Царев Иван, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Надеяев Денис, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н, учитель **Краснёнкова Л.А.**;

— Николаева Анна, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

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

— Тулеев Арсен, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

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

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

— Япрынцева Ольга, Республика Башкортостан, г. Стерлитамак, школа № 24, учитель **Орлова Е.В.**

Заметим, что в ряде ответов использовались знаки факториала и операции возведения в степень и извлечения корня, что не соответствует условию задачи.

Задача “Три карточки”

Напомним условие: “Перед Алешей и Митей на столе лежат три перевернутые карточки, под одной написано число 1, под другой — 2, под третьей — 3. Алеша их перемешал и вытащил одну из них, но какую — Мите не сказал. Митя может задать ему только один вопрос, на который тот, подумав, должен честно ответить “Да”, “Нет” или “Не знаю”, после чего Митя должен

наверняка отгадать число, которое вытащил Алеша. Как?”.

Ответ

Пусть Митя покажет на любую из оставшихся карточек (не переворачивая ее) и спросит: “Число на твоей карточке больше, чем на этой?”.

Если Алеша вытащил “1”, то он ответит “Нет” (ведь $1 < 3$ и $1 < 2$), если 2 — то “Не знаю” (то ли $2 > 1$, то ли $2 < 3$), а если 3 — то “Да” ($3 > 1$ и $3 > 2$).

Заметим, что предложенный рядом читателей вопрос Мити “Это число меньше двух?” не позволит ему определить искомое число, поскольку Алеша ответит “Нет” в двух случаях — когда он вытащил карточку с числом 2 или с числом 3.

Ответы прислали:

— Алексеева Анастасия, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Деревянченко Дарья, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

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

— Коробов Сергей и Яснов Федор, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Николаева Анна, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Нилова Екатерина и Петрова Полина, гимназия г. Шелехова, Иркутская обл., учитель **Водальчук С.А.**;

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

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

— Сергеева Анна, г. Ростов-на-Дону, лицей № 56, учитель **Назаренко С.Н.**;

— Тулеев Арсен, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

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

— Хвойновский Вадим, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**

Задача “Яд и кролики”

Напомним условие: “Есть три бутылки с водой. В одной из них может быть растворен яд. Для его нахождения можно использовать кроликов (к сожалению, для научных исследований, для проведения экспериментов и т.п. иногда используются животные — мыши, кролики и другие²). Яд действует через два дня и смертелен в любой дозе. Как, имея двух кроликов, за минимально возможное время определить, есть ли яд в бутылках и, если есть, то в какой?”.

² Отсюда выражение — “подопытный кролик”.

Решение

Если пронумеровать бутылки и кроликов, то проверка может быть проведена, например, следующим образом.

В первый день первый кролик пьет из бутылки № 1, второй — из бутылки № 2. Во второй день проводится проверка бутылки № 3, для чего используется один из кроликов. Если на третий день умрет один из кроликов, то это даст возможность определить, в какой из бутылок растворен яд (увы, это будет сделано ценой жизни животного). Смерть кролика на четвертый день позволит сделать вывод о том, что вода в бутылке № 3 — отравленная. Если все подопытные животные останутся живы (ура!), то яда в бутылках нет.

Возможны и другие варианты проверки.

Ответы прислали:

— Алексеева Анастасия, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Баранова Алена, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Белоносова Екатерина и Горбунова Ксения, гимназия г. Шелехова, Иркутская обл., учитель **Водальчук С.А.**;

— Деревянченко Дарья, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Жакова Светлана, средняя школа поселка Озеры Красноярского края, учитель **Филипченко И.С.**;

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

— Коробов Сергей, Марков Алексей и Яснов Федор, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

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

— Медведева Анастасия, Владимирская обл., г. Струнино, школа № 11, учитель **Волкова Т.П.**;

— Мехмедов Арсен, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Миноцкий Ян и Хозин Марат, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

— Пинигина Екатерина, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н, учитель **Краснёнкова Л.А.**;

— Сергеева Анна, г. Ростов-на-Дону, лицей № 56, учитель **Назаренко С.Н.**;

— Хвойновский Вадим и Чернова Ксения, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

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

— Япрынцева Ольга, Республика Башкортостан, г. Стерлитамак, школа № 24, учитель **Орлова Е.В.**

Головоломка “Переложить одну спичку”

Напомним, что требовалось переложить одну спичку, чтобы слева и справа от знака равенства было указано одно и то же количество:



Решение

Надо воспользоваться тем, что в римской системе счисления X — это 10, а IX — это 9. Итак, головоломка имеет два решения:

1)



2)



Правильные ответы представили:

— Абдувахидова Алина, Абдувахидова Софья, Милушкин Дмитрий, Хозин Марат и Чуб Алексей, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Алимова Алина, Степанов Сергей и Трофимова Мария, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Белоносова Екатерина, Нилова Екатерина, Маслий Елизавета и Петрова Полина, гимназия г. Шелехова, Иркутская обл., учитель **Водальчук С.А.**;

— Бульбова Лидия, средняя школа г. Пионерский Калининградской обл., учитель **Багрова О.А.**;

— Булыкина Алевтина, Костромина Елена и Цаплина Анна, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Васильева Екатерина, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н, учитель **Краснёнкова Л.А.**;

— Ганиева Лилия, Республика Татарстан, г. Казань, МБОУ “Татарская гимназия № 1”, учитель **Багаутдинова Р.Р.**;

— Данилова Мария и Разумова Алина, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Деревянченко Дарья, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

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

— Иванова Ксения и Мухина Светлана, г. Пенза, школа № 512, учитель **Гаврилова М.И.**;

— Ким Леонид и Литвинов Андрей, г. Смоленск, школа № 29, учитель **Родикова Р.Д.**;

— Медведева Анастасия, Владимирская обл., г. Струнино, школа № 11, учитель **Волкова Т.П.**;

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

— Обухова Елизавета, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Огнев Ярослав, г. Орел, лицей № 4 им. Героя Советского Союза Г.Б. Злотина, учитель **Чапкевич И.М.**;

— Рагузин Кирилл, г. Нижний Новгород, школа № 59, учитель **Лебедева М.Р.**;

— Сергеева Анна, г. Ростов-на-Дону, лицей № 56, учитель **Назаренко С.Н.**;

— Тарасов Андрей и Тарасова Нина, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Хвойновский Вадим и Чернова Ксения, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

— Япрынцева Ольга, Республика Башкортостан, г. Стерлитамак, школа № 24, учитель **Орлова Е.В.**

Задание “Пять вопросов” (рубрика “Поиск информации”)

Ответы

1. “Королем риса” называют сорт басмати.
2. Индийский царь Наль проиграл не только свои владения, но и жену в кости.

3. Пророк Иеремия в общем уподобляет еврейский народ агнцу (агнец — ягненок, козленок, приносившийся в жертву).

4. После своей смерти целую неделю пролежал в музее забальзамированным великий испанский художник Сальвадор Дали.

5. Самый известный герой книг Даниеля Дефо был родом из города Йорк. А имя его совпадает с названием книги, изображенной на фотографии, которую прислал Илья Иванов, ученик лицея № 4 им. Героя Советского Союза Г.Б. Злотина, г. Орел (учитель **Чапкевич И.М.**). Эта книга — любимая книга Ильи (на фотографии именно он держит книгу).



Кроме Ильи Иванова, ответы представили:

— Барановский Андрей, Глушко Иван, Казакова Анастасия, Мисюра Алексей, Нетесов Николай, Пак Александра, Пантелюк Руслан, Приходько Геннадий, Сысоев Александр и Фесько Роман, средняя

школа г. Пионерский Калининградской обл., учитель **Багрова О.А.**;

— Белоносова Екатерина и Петрова Полина, гимназия г. Шелехова, Иркутская обл., учитель **Водальчук С.А.**;

— Булыкина Алевтина и Костромина Елена, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Дегтярева Полина и Красненков Александр, средняя школа поселка Ерофей Павлович, Амурская обл., Сковородинский р-н, учитель **Краснёнова Л.А.**;

— Дервянченко Дарья, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Дикарева Марина и Трофимова Мария, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Евграфова Ксения, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Иванова Виолетта и Левченко Ирина, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Измайлова Екатерина, Кляузер Адам, Лобанов Артем, Мазаева Софья, Музалевская Екатерина, Огнев Ярослав, Худокормова Мария и Чекалин Егор, г. Орел, лицей № 4 им. Героя Советского Союза Г.Б. Злотина, учитель **Чапкевич И.М.**;

— Медведева Анастасия, Владимирская обл., г. Струнино, школа № 11, учитель **Волкова Т.П.**;

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

— Тарасов Антон, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Трус Анастасия, средняя школа села Дохновицы Брянской обл., учитель **Клопова Е.В.**;

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

— Хвойновский Вадим и Чернова Ксения, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

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

— Яснова Дарья, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**

В задании были приведены также факты, о которых, найдя соответствующую информацию в Интернете, следовало сказать, правда ли то, что в них утверждается.

Ответы

Факт 1. Пищевая ценность семечек кабачка и тыквы повышается с возрастом. Они относятся к продуктам, которые по мере разложения становятся более питательными. В семенах кабачка и тыквы, хранившихся более пяти месяцев, содержание белка сильно повышено.

Ответ — правда.

Факт 2. Когда Николай Васильевич Гоголь был маленьким, его бабушка рассказывала ему о божественной лестнице, которую спускают с неба ангелы, подавая руку душе умершего. Последними словами классика были: “Лестницу! Поскорее давай лестницу!”.

Ответ — правда.

Отметим ответы читателей из орловского лицея № 4 (учитель **Чапкевич И.М.**), снабдивших их красочными иллюстрациями (Екатерина Музалевская в ответе на вопрос 2 привела также отрывки из повести “Наль и Дамаанти” в переводе В.А. Жуковского).

Задача “Три ученика”

Напомним условие: “Три ученика различных школ из одного города приехали на отдых в летний лагерь. На вопрос вожатого, в каких школах они учатся, каждый дал ответ:

1) Петя: “Я учусь в школе № 24, а Ленья — в школе № 8”;

2) Ленья: “Я учусь в школе № 24, а Петя — в школе № 30”;

3) Коля: “Я учусь в школе № 24, а Петя — в школе № 8”.

Вожатый, удивленный противоречиями в ответах ребят, попросил их объяснить, почему так. Тогда ребята признались, что в ответах каждого из них одно утверждение верно, а другое — ложно. Смог ли вожатый определить, в какой школе учится каждый из мальчиков?”.

Решение

Допустим, что Петя учится в школе № 24. Тогда в высказывании Лени оба утверждения ложны, что противоречит условию. Значит, в высказывании Пети истинным является второе утверждение, то есть Ленья учится в школе № 8. Тогда в высказывании Лени истинным является второе утверждение (то есть Петя — ученик школы № 30), а в высказывании Коли истинное утверждение первое (то есть Коля учится в школе № 30).

Ответ. Вожатый на основе сделанных рассуждений смог определить, что Ленья учится в школе № 8, Коля — в школе № 24, Петя — в школе № 30.

Правильные ответы прислали:

— Абдувахидова Алина, Абдувахидова Софья, Милушкин Дмитрий и Хозин Марат, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Батурин Илья, Бульбова Лидия, Глушко Иван, Мисюра Алексей, Пак Александра, Панасенко Дарья, Приходько Геннадий и Сысоев Александр, средняя школа г. Пионерский Калининградской обл., учитель **Багрова О.А.**;

— Белоносова Екатерина, Грекова Ирина, Нилова Екатерина и Петрова Полина, гимназия г. Шелехова, Иркутская обл., учитель **Водальчук С.А.**;

— Дервянченко Дарья, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Евграфова Ксения, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Коробов Сергей и Яснов Федор, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

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

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

— Прокопенко Павел, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Сергеева Анна, г. Ростов-на-Дону, лицей № 56, учитель **Назаренко С.Н.**;

— Тарасов Антон, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Хвойновский Вадим и Чернова Ксения, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**;

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

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

— Япрынцева Ольга, Республика Башкортостан, г. Стерлитамак, школа № 24, учитель **Орлова Е.В.**

Кроссворд

Ответы

По горизонтали: 6. Домен. 7. Бод. 8. Растр. 12. Альфа. 13. Список. 15. Массив. 20. Вариант. 21. Аксиома. 24. Отрицание. 25. Ноль. 26. Абак. 27. Ресторан. 28. Столбец. 29. Альт.

По вертикали: 1. Адрес. 2. Имя. 3. “Кольцо”. 4. Шаг. 5. Ершов. 9. Пакет. 10. Рамка. 11. Остаток. 14. Пароль. 16. Система. 17. Иголка. 18. Свобода. 19. Паскаль. 22. Тире. 23. Байт.

Ответы прислали:

— Анохин Андрей, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Байбуза Дарья, Горелкина Лариса и Кузнецов Семен, средняя школа села Ириновка, Новобураский р-н Саратовской обл., учитель **Брунов А.С.**;

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

— Деревянченко Дарья, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Ежиков Тимофей и Корбут Илья, средняя школа поселка Ерофей Павлович, Амурская обл., Сквородинский р-н, учитель **Краснёнкова Л.А.**;

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

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

— Прокопенко Павел, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Сергеева Анна, г. Ростов-на-Дону, лицей № 56, учитель **Назаренко С.Н.**;

— Тарасов Антон, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Федотова Виктория, г. Смоленск, школа № 29, учитель **Родикова Р.Д.**;

— Ханин Егор, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

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

— Шумилов Александр, г. Пенза, школа № 512, учитель **Гаврилова М.И.**

Числовые ребусы в троичной системе. Часть 6

Ответы:

1) $2_3 + 22_3 = 101_3$;

2) $20_3 + 20_3 = 110_3$;

3) $21_3 + 21_3 = 112_3$;

4) $10_3 + 10_3 = 20_3$.

Правильные ответы представили:

— Жакова Светлана, средняя школа поселка Озёры Красноярского края, учитель **Филипченко И.С.**;

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

— Лукашин Станислав, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

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

— Рыбалко Иван, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

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

— Ханин Егор, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

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

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

— Шарипова Анастасия, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Чукмасова Надежда, средняя школа поселка Ерофей Павлович, Амурская обл., Сквородинский р-н, учитель **Краснёнкова Л.А.**

Статья “Двуносый чайник и три чашки”

Напомним, что требовалось, используя чайник с двумя носиками, заполнить три чашки:

1) вместимостью 10, 12 и 14 условных единиц;

2) вместимостью 4, 4 и 16 условных единиц;

3) вместимостью 5, 5 и 25 условных единиц.

Во всех случаях принять, что в чайнике имеется достаточно большое количество воды.

Ответы прислали:

— Адаменко Илья, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Анохин Андрей, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Бульбова Лидия, средняя школа г. Пионерский Калининградской обл., учитель **Багрова О.А.**;

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

— Лукашин Станислав, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Муранов Даниил, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Токмурзин Данил, Свердловская обл., Красноуфимский р-н, Тавринская средняя школа, учитель **Ярцев В.А.**;

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

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

— Шальнов Сергей, средняя школа поселка Озеры Красноярского края, учитель **Филипченко И.С.**;

— Шарипова Анастасия, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**

Решения

Оптимальные решения для трех задач (соответственно, за 4, 5 и 7 действий) показаны в таблицах ниже.

№	Действие	В 1-й чашке (10)	Во 2-й чашке (12)	В 3-й чашке (14)
Исходное состояние		0	0	0
1	Налить в 1-ю и 2-ю чашки	10	10	0
2	Налить во 2-ю и 3-ю чашки	10	12	2
3	Вылить из 2-й чашки в чайник	10	0	2
4	Налить во 2-ю и 3-ю чашки	10	12	14

№	Действие	В 1-й чашке (4)	Во 2-й чашке (4)	В 3-й чашке (16)
Исходное состояние		0	0	0
1	Налить в 1-ю и 3-ю чашки	4	0	4
2	Налить во 2-ю и 3-ю чашки	4	4	8
3	Перелить из 1-й чашки в 3-ю	0	4	12
4	Перелить из 2-й чашки в 3-ю	0	0	16
5	Налить в 1-ю и 3-ю чашки	4	4	16

№	Действие	В 1-й чашке (5)	Во 2-й чашке (5)	В 3-й чашке (25)
Исходное состояние		0	0	0
1	Налить в 1-ю и 3-ю чашки	5	0	5
2	Налить во 2-ю и 3-ю чашки	5	5	10
3	Перелить из 1-й чашки в 3-ю	0	5	15
4	Перелить из 2-й чашки в 3-ю	0	0	20
5	Налить в 1-ю и 2-ю емкости	5	5	20
6	Вылить из 1-й чашки в чайник	0	5	20
7	Налить в 1-ю и 3-ю чашки	5	5	25

К сожалению, никто из представивших ответы не привел эти оптимальные варианты (приведенные в ряде ответов действия “Перелить из 1-й и 2-й чашек в 3-ю” и т.п. редакция рассматривает как два разных действия).

Задача “Восстановите набор чисел”

Напомним, что следовало определить, какое число “скрывается” под знаком вопроса, если все три набора чисел построены по одному правилу:

4: 6, 11, 7;
3: 8, 4, 12;
?: 10, 55, 15.

Ответ

Вместо знака вопроса должно быть указано число 8.

Обоснование

Все числа перед двоеточием получаются путем суммирования трех чисел набора, записанных после двоеточия, и деления полученной суммы на первое число набора. Для третьего набора будет: $(10 + 55 + 15)/10 = 8$.

Правильный ответ прислали:

— Леоненко Степан, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Макаров Тимофей и Цаплина Анна, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**;

— Рыбалко Иван, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

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

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

— Чудакова Ирина, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**

Задания, предложенные для самостоятельной работы в статье “Пять задач на системы счисления. Часть 3”, выполнили:

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

— Адаменко Илья, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

— Андросенко Дмитрий, средняя школа села Дохновичи Брянской обл., учитель **Клопова Е.В.**;

— Гируцкий Павел, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Носов Анатолий, г. Пермь, школа № 109, учитель **Онохова М.А.**

Учитывая сложность и важность предложенных заданий, редакция решила наградить всех перечисленных читателей дипломами. Дипломами также будут награждены читатели, правильно выполнившие все задания, предложенные для самостоятель-

ной работы в статье “Фальшивые монеты, отравленное вино и... системы счисления” (январский выпуск “В мир информатики”):

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

— Миноцкий Ян и Хозин Марат, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Прокопчик Дмитрий, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**

Поздравляем!

Японские головоломки sudoku, опубликованные в февральском выпуске “В мир информатики”, решили:

— Анохин Андрей, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Бульбова Лидия, средняя школа г. Пионерский Калининградской обл., учитель **Багрова О.А.**;

— Воротников Григорий, Горбунова Ксения и Попова Елизавета, гимназия г. Шелехова, Иркутская обл., учитель **Водальчук С.А.**;

— Деревянченко Дарья, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Макаров Тимофей, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Милушкин Дмитрий и Хозин Марат, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Морозова Елена, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Сергеева Анна, г. Ростов-на-Дону, лицей № 56, учитель **Назаренко С.Н.**;

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

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

— Чудакова Ирина, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**

Задания, предложенные для самостоятельной работы в статье “Трилистник и другие”, выполнили:

— Гируцкий Павел, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

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

— Пак Александра, средняя школа г. Пионерский Калининградской обл., учитель **Багрова О.А.**;

— Чудакова Ирина, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**

Спасибо всем приславшим ответы!

После подготовки основного списка читателей, приславших ответы, были получены правильные ответы Александра Волкова, в прошлом учебном году — ученика 2-го (!) класса школы “Логос” г. Мытищи Московской обл. (учитель **Елистратова А.А.**):

— с ответом к задаче “Четыре талантливые девочки”;

— с решением головоломки “Переложить одну спичку”;

— с ответом к задаче “Три ученика”;

— с решением задачи “Обманутый хозяин”.

Редакция решила наградить Александра дипломом. Молодец!

Некоторые проблемы и риски информатизации

В.П. Константинов,
Москва

Современная эпоха характеризуется бурным развитием информационных технологий (ИТ), охватывающих самые разные сферы человеческой деятельности. Если ИТ ранее в основном использовались в научных исследованиях и военном деле, то сейчас они внедряются в управление технологическими процессами и производственными предприятиями, в экономику и финансы, в государственное управление, в образование, медицину, а также захватывают и сферу досуга. Электронные средства массовой информации (СМИ) и Интернет становятся основным источником информации для огромной массы людей, вытесняя печатную продукцию. Бурное развитие ИТ оказывает все большее влияние на жизнь отдельного человека и всего общества, увеличивает зависимость человека от компьютера и других электронных устройств, и несет людям не только блага, но и опасности.

О благах информатизации публикуется огромное количество материала, как в специальной литературе, так и в обычных СМИ. О рисках и опасностях пишут и говорят меньше, хотя эти риски и опасности следует знать заранее, чтобы начать бороться с ними как можно раньше.

Возможные опасности, связанные с внедрением любой новой технологии, можно классифицировать следующим образом:

- 1) несовершенство и недостаточная надежность самих новых технологий и основанных на их применении устройств;
- 2) использование новых технологий людьми в деструктивных, антигуманных целях (войны, криминальные действия, терроризм, вандализм, хулиганство и т.д.);
- 3) сложность адаптации людей к новым технологиям и ошибки проектирования систем;
- 4) ошибки людей, использующих эти технологии в своей деятельности;
- 5) влияние новых технологий на здоровье людей;
- 6) общественные последствия: влияние на массовое сознание, воспитание молодежи и в конечном счете на весь уклад жизни.

Можно выделить следующие источники ненадежности современных ИТ:

- ненадежность используемой аппаратуры связи и средств вычислительной техники, применяемых для обработки данных;
- ошибки, возникающие при передаче данных по каналам связи;
- ошибки проектирования.

По мере развития электроники надежность электронной аппаратуры постепенно повышается.

В особо важных случаях используются дублирование аппаратуры, дублирование информационных ресурсов и специальные методы конструирования больших систем, при которых отказ отдельных элементов системы не выводит из строя систему в целом. По такому принципу, в частности, построена сеть Интернет, где при отказе одного из узлов сети связь между компьютерами будет осуществляться по обходным путям.

Ошибки в каналах связи зависят не только от надежности аппаратуры связи, но и от таких случайных факторов, как различные электромагнитные помехи и тепловые шумы. Однако существуют специальные методы помехоустойчивого кодирования, которые позволяют повысить надежность передачи информации за счет некоторого снижения скорости передачи.

Намного сложнее обстоит дело с ошибками проектирования, поскольку они зависят не от качества технологии производства, которое постоянно повышается, а от человеческого фактора. Но, как известно, человеческая природа меняется очень медленно. Ошибки могут быть допущены и в проектировании технических средств, и в разработке алгоритмов и/или программ. Ошибки эти обычно выявляются с помощью тестов. Тестирование сравнительно легко проводить тогда, когда хотя бы для одного конкретного примера задачу можно решить вручную, не пользуясь компьютером. Ситуация становится очень сложной, когда вообще задачу нельзя решить без компьютера и ни для какого частного случая правильный ответ неизвестен. Проверить правильность расчетов на практике возможно не всегда. Не станете же вы проверять правильность расчетов ядерного реактора по принципу “взорвется — не взорвется”! Единственно возможное решение здесь — проведение расчетов по нескольким независимо написанным программам, в которых используются разные алгоритмы. В случае же использования компьютерной системы для управления какими-либо процессами в реальном времени всегда необходимо устанавливать простейшие (а потому и высоконадежные) устройства автоматической защиты, которые не позволяли бы переходить процессу в опасный режим.

Окончательное выявление ошибок производится уже в процессе эксплуатации. Чаще всего ошибки встречаются в программном и алгоритмическом обеспечении. Но известны случаи, когда ошибки выявлялись и в серийно производимых технических устройствах. Например, в течение некоторого времени фирма Intel выпускала микропроцессоры, которые давали сбои при поступлении на их вход определенного кода. Наиболее опасны ошибки, которые проявляются в редких случаях и никак не обнаруживают себя в обычных условиях. В частности, это касается различных средств безопасности, которые нужны только в аварийных ситуациях. Известно, например, что основной причиной Черно-

бильской катастрофы были конструктивные ошибки в последнем уровне системы аварийной защиты, но на это никто не обращал внимания, потому что до этого уровня дело никогда не доходило: срабатывали более ранние уровни системы защиты.

Использование ИТ в деструктивных целях последнее время растет такими быстрыми темпами, что в литературе уже появились термины “информационное оружие” и “информационная война”. В литературе дается следующее определение: “Информационная война — это любое действие, направленное на:

- использование, разрушение, искажение вражеской информации и ее функций;
- защиту собственной информации против подобных действий;
- использование собственных военных информационных функций”.

Информационное оружие (так же, впрочем, как и многие другие виды оружия) может быть использовано не только в военных, но и в политических, идеологических и экономических конфликтах между государствами, а также и в различных внутренних конфликтах в одном государстве. Информационное оружие может быть использовано криминалом, террористами и даже хулиганами.

Не менее быстрыми темпами развивается и противодействие информационным атакам, прежде всего на техническом уровне. Этому вопросу посвящено огромное число публикаций и в специальной литературе, и в СМИ.

Можно следующим образом охарактеризовать ситуацию, сложившуюся в области использования новых технологий в деструктивных целях и борьбы с такими действиями.

1. С одной стороны, развитие ИТ (прежде всего телевидения и Интернета) резко расширяет возможности воздействия на человеческое сознание и общественное мнение. С другой стороны, широкое внедрение ИТ в оборонные, государственные, корпоративные и технологические системы ведет ко все большей зависимости государства и общества от информационных инфраструктур и делает их уязвимыми в случае атак на эти инфраструктуры. В результате этих процессов в настоящее время информационное оружие из вспомогательного средства, обеспечивающего ведение обычных боевых действий, превращается в самостоятельное средство ведения борьбы, позволяющее достигать цели, которые раньше можно было достичь только военным путем. Примером тому могут служить революции в арабских странах, осуществить которые без современных средств массовой информации (в первую очередь — электронных) было бы затруднительно.

2. Низкая стоимость и простота применения (но не разработки!) информационного оружия позволяет использовать его не только государственным структурам, но и террористам, отдельным полити-

ческим группировкам и партиям, бизнес-структурам, криминалу и даже хулиганам.

3. Считается, что современная война — это война между высокоразвитыми и отсталыми районами мира. Относительно низкая стоимость и простота применения информационного оружия делают его удобным для применения “бедными” против “богатых”. Но, с другой стороны, бедные страны вынуждены закупать современное оружие, равно как и оборудование для стратегических объектов, в развитых странах. При этом возникает возможность поставки оружия или оборудования, содержащего специальные аппаратные или программные “закладки” (“логические бомбы”), которые при получении извне определенного сигнала могут вывести оружие или оборудование из строя. В частности, после первой войны в Персидском заливе (в начале 90-х годов) в печати было много сообщений, что система ПВО Ирака была полностью выведена из строя с помощью какой-то “закладки” (программной или аппаратной), которая была сделана при поставке этой системы Ираку. (Каких-либо технических подробностей в печати не было.)

4. Наиболее уязвимыми для информационного оружия объектами информационной инфраструктуры являются радиоканалы и сеть Интернет. Единственным эффективным средством защиты этих объектов является защита на уровне представления данных, т.е. криптография и стеганография. Что же касается наиболее ответственных военных и других систем, имеющих стратегическое назначение, то они должны быть вообще физически изолированы от Интернета.

5. Судя по опубликованным статистическим данным, наибольшее количество атак на корпоративные ИТ-системы приходится на недобросовестных служащих самой корпорации и на нападения хакеров из сети Интернет. Однако наибольший ущерб, по крайней мере в мирное время, приносят все же вредоносные программы (компьютерные вирусы, черви, троянские программы), поскольку в первом случае речь идет о единичных атаках, а вирусы и черви вызывают массовое поражение компьютеров. Известны случаи, когда один вирус поражал свыше миллиона компьютеров во всем мире. Поэтому защита от вредоносных программ является приоритетным вопросом. Недавно нашему известному “вирусологу” Евгению Касперскому был задан вопрос: “Может ли появиться такой вирус (червь), который полностью разрушит Интернет?”. Ответ был, что это крайне маловероятно, хотя теоретически исключить возможность появления червя, который забьет весь Интернет, нельзя.

Если лет пять, а тем более десять тому назад основным видом вредоносных программ были компьютерные вирусы, которые так же, как биологический вирус встраивается в ДНК клетки, встраивались в тело имеющихся на компьютере программ, то в последнее время все больше становится ком-

пьютерных червей, представляющих собой отдельные программные файлы, а также увеличивается число троянских программ, которые, как правило, тоже являются отдельными исполняемыми файлами. Это и понятно.

6. В последнее время большое внимание уделяется электромагнитному оружию, которое выводит из строя электронные средства и системы противника, не вызывая каких-либо разрушений физических объектов. Если лет тридцать тому назад единственным источником мощного электромагнитного импульса (ЭМИ), способного необратимо вывести из строя электронную аппаратуру, считался только ядерный взрыв, то сейчас созданы достаточно малогабаритные и мобильные устройства, потребляющие сравнительно небольшую мощность, которые могут создать такой импульс. Как показывают теоретические работы и проведенные за рубежом эксперименты, генераторы ЭМИ можно эффективно использовать для вывода из строя электронной и электротехнической аппаратуры, для стирания информации, хранящейся в оперативной и внешней памяти компьютеров, для вывода из строя последних, а также для вывода из строя систем электронного зажигания и других автомобильных агрегатов, подрыва или инактивации минных полей.

Следует сказать, что многочисленные публикации в обычной печати, не говоря уже о различных кинобоевиках, сильно преувеличивают уязвимость военных и стратегически важных компьютерных систем к атакам из Интернета, с которым на самом деле никто такие системы не связывает.

Что же касается четырех остальных классов опасностей, то им уделяется намного меньше внимания, хотя бороться с ними гораздо труднее, поскольку они связаны не с недостатками техники, а с недостатками человеческой природы, которая изменяется очень медленно. Причем роль этих опасностей постоянно возрастает, потому что с информационными технологиями в работе и повседневной жизни сталкивается все большее количество людей, значительная часть которых не имеет специальной подготовки в области ИТ.

Очень серьезной проблемой является явление, которое можно охарактеризовать как взаимное непонимание человека и компьютера. Дело в том, что в настоящее время практически вся обработка информации производится по заданным алгоритмам и поэтому постановка любой задачи, которая решается на компьютере, должна быть полностью формализована. Но человеческое мышление, так же, как и речь на любом естественном языке, неформальны. Многие решения человек принимает не на рациональном, а на интуитивном уровне, даже не понимая, как это делается. При подготовке любой задачи к решению ее на компьютере приходится первоначальную постановку задачи, сформулированную на естественном языке, формализо-

вать. Иначе составить машинный алгоритм решения задачи невозможно. При этом часто возникает взаимное непонимание между заказчиком, хорошо знающим предметную область, но не имеющим никакой подготовки в области ИТ, и специалистом в области ИТ — разработчиком задачи. Заказчики тех или иных автоматизированных систем обработки информации и управления (АСОИиУ) очень часто бывают далеки от современных ИТ и не могут корректно поставить задачу перед разработчиком системы. Некоторое время назад, в эпоху создания первых автоматизированных систем управления, это приводило к очень большим потерям, и многие серьезные проекты из-за этого были сорваны. В настоящее время в связи с широким внедрением ЭВМ в различные отрасли конечные пользователи стали лучше разбираться в современных ИТ, а среди разработчиков АСОИиУ, по крайней мере в областях массового применения вычислительной техники (например, в экономике, юриспруденции и управлении некоторыми технологическими процессами), появились специалисты-компьютерщики, хорошо изучившие определенную предметную область. Об этом свидетельствует появление фирм, специализирующихся на создании пакетов прикладных программ определенного назначения. Примером могут служить российские фирмы, разработавшие программные пакеты “1С” (бухгалтерия, складское хозяйство, торговля) и “Консультант+” (юриспруденция). Однако в случае внедрения информационных технологий в новые, еще не затронутые информатизацией области эта проблема может возникнуть и сейчас.

Проблема может сильно обостриться в будущем, когда появится возможность решения на компьютере нечетких задач и общения человека и компьютера на естественном языке. Когда человек ставит какую-либо задачу перед другим человеком, он указывает далеко не все условия, которым должно соответствовать решение. Многие вещи подразумеваются, и люди обычно понимают друг друга правильно. При общении с “искусственным разумом” это недопустимо. Ситуация усугубляется тем, что при появлении возможности вести диалог с компьютером на естественном языке среди “постановщиков задач” безусловно резко возрастет число непрофессионалов, далеких от ИТ.

О том, к каким трагическим последствиям может привести неформализованная постановка задачи, еще в середине 50-х годов говорил “отец кибернетики” Норберт Винер. В своей книге “Кибернетика и общество” он приводит легенду об обезьяньей лапе. Ее суть такова. Существует некий колдовской амулет — обезьянья лапа, который может выполнить три любых желания его владельца. Но выполняет он их так, что после этого человек кончает жизнь самоубийством. Некий владелец обезьяньей лапы выражает желание получить миллион долларов. Тут же ему приносят извещение страховой фирмы о

том, что его отец погиб в автомобильной катастрофе и “заказчик” может получить страховую премию миллион долларов. Человек говорит: “Хочу, чтобы мой отец был здесь”. (Сказать слово “живой” он в расстройстве чувств забыл.) И ему приносят обезображенный труп отца. В ужасе человек восклицает: “Я не хочу больше жить!”. Формально все желания человека выполнены абсолютно точно, а нормы морали, которые человек подразумевает, для обезьяньей лапы не обязательны. Конечно, компьютерный алгоритм, в отличие от обезьяньей лапы, не обладает злой волей, но и никакие само собой разумеющиеся для человека истины ему неизвестны. И поэтому их необходимо специально оговаривать при постановке задачи. Придуманый американским писателем-фантастом Айзеком Азимовым 1-й закон робототехники: “Робот никогда не должен причинять вреда человеку, он хорош, но его нельзя реализовать потому, что невозможно полностью перечислить все виды вреда, в том числе такие, о которых наука еще не знает”.

Следует иметь в виду, что по мере расширения применения ИТ растет доверие к любой информации, получаемой от компьютера, что приводит к некритическому отношению к этой информации. К этому следует добавить, что при увеличении сложности расчетов обнаруживать ошибки в них будет все труднее. Несколько лет тому назад в печати был описан случай смерти человека от компьютерного вируса. Вирус поразил больничный компьютер, которым пользовались врачи для получения различной справочной информации и расчетов требуемой дозировки лекарств в зависимости от диагноза и различных параметров организма больного. К несчастью, компьютер не был полностью выведен из строя, а всего лишь незначительно была повреждена база данных. Врач обратился к компьютеру, получил неверную рекомендацию по лечению больного, но, привыкнув к тому, что компьютер не ошибается, врач назначил неверное лечение. Дело кончилось печально.

Проблема обнаружения таких ошибок в будущем может оказаться серьезным тормозом в развитии информационных технологий. Причин здесь несколько. Во-первых, с ростом сложности решаемых задач объективно растет трудность обнаружения без компьютера даже грубых ошибок в расчетах. Во-вторых, привыкнув к тому, что компьютер никогда не врет, человек отвыкает оценивать правильность полученного ответа. И, в-третьих, он разучивается делать то, что поручает компьютеру. Люди теряют умение произвести вручную грубый подсчет и оценить достоверность полученного результата, равно как и не запоминают полученные ранее результаты аналогичных расчетов с похожими данными.

В любой области техники при повышении надежности технических средств наступает момент, когда основным источником ненадежности соответствующих технических систем становится че-

ловеческий фактор, то есть вызванные теми или иными причинами ошибки оператора. Можно выделить следующие основные виды ошибок, связанных с человеческим фактором, возникающих при эксплуатации АСОИиУ:

1) ошибки конечных пользователей АСОИиУ и эксплуатационного персонала при вводе данных в процессе решения функциональных задач и управления вычислительных задач;

2) ошибки, так или иначе нарушающие безопасность системы;

3) ошибки системных программистов в процессе начальной установки и реконфигурации программного обеспечения АСОИиУ.

Ошибки и безответственная работа операторов — это очень важная проблема, значение которой будет только расти по мере развития информатизации общества и увеличения цены таких ошибок. Ведь компьютеры будут использоваться во все более важных системах управления и обработки данных. Проблема строгого соблюдения всех инструкций по эксплуатации чрезвычайно важна не только для ИТ-систем, но и для любых сложных систем.

Анализ специальной литературы и личный опыт авторов показывают, что наибольшее количество успешных информационных атак были связаны не с плохим качеством технических средств и не с появлением каких-то “неотразимых” методов нападения, а с неграмотной эксплуатацией средств защиты, а то и просто с их безответственным отключением. Здесь действует психологический фактор, имеющий место в любых системах безопасности: реализация угрозы, будь то информационная атака, пожар, взрыв или ДТП, являются редкими случайными событиями, и люди надеются на “русский авось”. Известен, например, случай, когда очень серьезная система обработки информации была на четыре дня выведена из строя из-за того, что местные “рационализаторы” для повышения быстродействия отключили на нескольких серверах антивирусные программы. Да ведь и основной причиной Чернобыльской катастрофы были грубые нарушения эксплуатационных инструкций. Проводя некоторые испытания, люди просто отключили мешающие им системы автоматической защиты, кроме одной, которая, как уже было сказано выше, была спроектирована с ошибками. К сожалению, важность этой проблемы очень часто недооценивается, и человеческому фактору уделяется недостаточное внимание при проектировании и эксплуатации средств ИТ.

Очень важной проблемой, которая до сих пор изучена недостаточно, является влияние тех или иных ИТ на здоровье людей. Например, традиционно считалось, что главным фактором вредности при работе перед экраном монитора являются создаваемые им электромагнитные излучения, начиная от мягкого рентгеновского диапазона и кончая

радиодиапазоном, а также внешне незаметное мигание экрана с частотой кадровой развертки. При переходе от мониторов на электронно-лучевых трубках к плазменным или жидкокристаллическим индикаторам ситуация изменяется в лучшую сторону. Однако в последнее время в печати появились сообщения, что рост близорукости у детей, которые слишком рано знакомятся с компьютером, в первую очередь связан с необходимостью читать на экране слишком мелкие надписи. Если это верно, нужно по-другому подходить к разработке графического интерфейса. Ясности в этом вопросе нет, и необходимо проводить серьезные медицинские исследования.

Недавно в Интернете появилось сообщение об исследованиях, проведенных европейскими учеными, которые показали, что под воздействием излучения мобильных телефонов происходят необратимые изменения ДНК клеток, что может привести к возникновению раковых опухолей у человека. Однако эти исследования проводились только на клеточных культурах, и окончательные выводы делать еще рано.

Имеются сообщения о возможности создания компьютерных вирусов, могущих причинить существенный вред здоровью человека или даже убить его. Вирус использует прием, давно применявшийся в рекламной сфере, при котором в выводимые на экран кадры изредка вставляются кадры, которые не воспринимаются сознанием человека, но воздействуют на его подсознание, вызывая у него повышение кровяного давления. Насколько такое сообщение достоверно с медицинской точки зрения, неизвестно. Однако один случай воздействия на здоровье человека телевизионного изображения известен. Некоторое время назад в Японии была зафиксирована эпидемия эпилепсии среди детей, смотревших по телевизору фильм. Заметим, что в медицинской литературе давно были описаны случаи, когда мигание света с определенной частотой вызывало припадки у людей, больных эпилепсией. Однако нет никаких сведений о влиянии такого мигания на здоровых людей. С тех пор, когда появились эти публикации, прошло более десяти лет. Никаких сведений о появлении таких вирусов до сих пор нет. Почему? Оптимистический ответ на этот

вопрос состоит в том, что их невозможно создать вообще. А пессимистический ответ будет: эти вирусы давно уже созданы соответствующими военными лабораториями или спецслужбами, а никто их до сих пор не использовал, потому что еще не настал час “X”.

Массовое внедрение информационных технологий оказывает все большее влияние не только на производственную и научную сферу деятельности человечества, но и на обучение и воспитание молодежи и на изменение характера и образа мысли людей. Массовое внедрение даже таких простых машин, как калькуляторы, в последние годы привело не только к потере навыков устного счета среди молодежи, но и к потере умения выполнять арифметические действия “в столбик”. На первый взгляд эти навыки сейчас вроде бы и не нужны. А какое влияние это явление оказывает на общее интеллектуальное развитие, не знает никто.

Имеются сообщения о случаях маниакальной зависимости от компьютерных игр и даже о появлении у некоторых молодых людей маниакального желания работать в Интернете. Игромания была известна еще в XIX веке, но тогда это было очень редкое, можно даже сказать — экзотическое душевное заболевание, а после появления компьютеризированных игровых автоматов она стала настоящим бедствием. Это явление можно было бы предвидеть заранее и принять соответствующие меры (Пушкина и Достоевского ведь читали многие!). Но спохватились и приняли решение выгнать “одноруких бандитов” в специально построенные “Лас-Вегасы” только тогда, когда бедствие достигло огромных масштабов.

Какое влияние на качество знаний может оказать массовая замена обычного обучения автоматизированным, дистанционным, не знает никто. Некоторыми авторами высказывается мнение, что постоянное общение детей с компьютером с малых лет может привести к дегуманизации и ожесточению общества. В частности, под подозрение ставятся столь любимые среди молодежи игры “стрелялки”. Хотя другие авторы считают, что эти игры “снимают” агрессию.

Все это говорит о необходимости проведения серьезных медицинских, психологических, педагогических, а возможно, и социологических исследований, посвященных проблемам информатизации.

ЭТО ПОЛЕЗНО ЗНАТЬ

Память калькулятора

При расчетах на калькуляторе часто приходится запоминать или записывать промежуточные результаты, с тем чтобы использовать их в дальнейших вычислениях. Например, продавцы хозяйственных товаров, работающие без компьютеризированных кассовых аппаратов, при продаже, например, пяти кусков мыла по 22 рубля каждый,

трех коробок стирального порошка стоимостью 53 рубля каждый и двух бутылок жидкого мыла по 73 рубля каждая должны провести три умножения, после чего сложить три полученных произведения.

У большинства калькуляторов для хранения промежуточных результатов (в описанном случае — трех произведений) имеется специальное запоминающее устройство — регистр памяти. Операции с регистром памяти производят, как правило, с помощью следующих клавиш:

M+	При помощи этой клавиши в регистр памяти вводится число, изображенное на индикаторе калькулятора. Если там уже было какое-то число, то новое прибавляется к нему.
M-	При нажатии на эту клавишу число, изображенное на индикаторе калькулятора, вычитается из числа, хранящегося в регистре памяти.
MR	Название этой клавиши (“ Memory Recall ”) можно перевести как “из памяти”. При нажатии на нее число, хранящееся в регистре памяти, высвечивается на индикаторе калькулятора. Если перед этим была нажата клавиша какого-либо действия, то число из памяти будет вторым операндом этого действия (результат получается при нажатии на клавишу $\langle = \rangle$).
MC	Это клавиша сброса, или стирания, числа из регистра памяти (“ Memory Clear ”). У некоторых калькуляторов ее нет и, чтобы очистить регистр памяти, нажимают последовательно клавиши $\langle MR \rangle$ и $\langle M-\rangle$. Иногда клавиши $\langle MR \rangle$ и $\langle MC \rangle$ “объединяют” в одну. В этом случае на ней написано $\langle M \frac{R}{C} \rangle$ — первое нажатие на эту клавишу соответствует нажатию на $\langle MR \rangle$, второе — на $\langle MC \rangle$.

В калькуляторе, имеющемся в компьютере (в операционной системе Windows), предусмотрена также клавиша $\langle MS \rangle$. Она предназначена для занесения в память числа, изображенного на индикаторе калькулятора. При этом, в отличие от нажатия клавиши $\langle M+ \rangle$, это число не прибавляется к уже имеющемуся в памяти, а заменяет его.

Прежде чем предлагать задания для самостоятельной работы, выскажем ряд рекомендаций по проведению расчетов с использованием регистра памяти калькулятора.

1. Помните, что в регистре памяти может храниться только одно число, поэтому тщательно продумывайте последовательность своих действий.

2. При вычислении дробей со сложными выражениями в числителе и знаменателе сначала вычисляют знаменатель и вводят результат в регистр памяти. Затем рассчитывают числитель и делят полученный результат на число, хранящееся в памяти.

3. Перед началом новых вычислений регистр памяти калькулятора очищают от “старых” значений.

Задания для самостоятельной работы

- Вычислите значение $\frac{3,4+25,9}{10,5-6,1}$.
- Получите таблицу умножения на 16 (внесите число 16 в регистр памяти):

1×16
2×16
...
10×16
- Вычислите значение $93,1 - (4,3 \times 6,8) - (15,2 - 3,4)$.
- Что лучше занести в регистр памяти при решении следующих примеров?

- $1 \times 9 + 2 =$
 $12 \times 9 + 3 =$
 $123 \times 9 + 4 =$
 $1234 \times 9 + 5 =$
- $143 \times 2 \times 7 =$
 $143 \times 3 \times 7 =$
 $143 \times 4 \times 7 =$
 $143 \times 5 \times 7 =$
 $143 \times 6 \times 7 =$

5. Переведите расстояния 85 км, 100 км, 115 км, 32 км и 270 км в мили, если 1 километр — это $\frac{5}{8}$ мили.

6. Задача “Турпоход”

Предположим, что вы, уважаемый читатель, идете в туристический поход. Ваш рюкзак вместе с палаткой и спальным мешком весит 6,75 кг. Еще вам нужно положить в рюкзак: спиртовку, кружку и ложку (все это весит 2 кг), два пакета с рисом (по 430 г каждый), семь пакетов суповых концентратов (по 85 г каждый), три пачки чая (по 113 г каждая), пять пакетиков с сухим молоком (по 21 г каждый).

Но ведь нужно еще взять с собой шоколад! Каждая плитка весит 103 г. С помощью калькулятора, не записывая результаты промежуточных вычислений на бумаге, определите, сколько плиток шоколада вы можете взять с собой, если за плечами у вас не должно быть больше 12 кг.

Ответы к задачам 4 и 6, пожалуйста, присылайте в редакцию.

.....
Уважаемые коллеги!

Редакция обращается к вам с просьбой поделиться на страницах журнала своим опытом использования материалов раздела “В мир информатики”. Пожалуйста, напишите, как статьи, задачи и условия конкурсов “доходят” до учащихся (в виде раздаточных материалов, в виде информации на стенде и т.п.).

Мудр не тот, кто знает много, а тот, чьи знания полезны.
Эсхил
 Образованность можно скрыть ото всех, невежество — только от себя.
В.З. Черняк

Заполнение ячеек порядковыми номерами

При решении многих практических задач требуется создавать столбец или строку с порядковыми номерами. Это можно сделать разными способами.

Опишем их на примере диапазона ячеек A3:A100.

1. Автозаполнение

1. Ввести в ячейку A3 значение 1 (начальное), в ячейку A4 — 2.

2. Выделить эти ячейки.

3. Поставить курсор мыши на правый нижний угол прямоугольника, ограничивающего выделенные ячейки, — курсор примет вид +. Это так называемый “маркер заполнения”.

4. “Отбуксировать” мышь вниз на требуемое количество ячеек (при этом рядом с маркером появится маленький квадратик, в котором будет отображаться меняющееся значение текущего порядкового номера, — рис. 1).

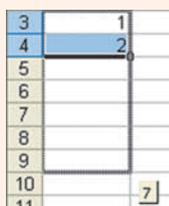


Рис. 1

Если вы отпустили кнопку мыши раньше, чем требуется, можно повторить описанные действия, используя на этапе 2 две последние заполненные ячейки.

Другой способ автозаполнения.

1. Ввести в ячейку A3 значение 1 (начальное).

2. Выделить ячейку A3.

3. Поставить курсор мыши на правый нижний угол прямоугольника, ограничивающего выделенные ячейки, — появится маркер заполнения.

4. Нажать клавишу **Ctrl**, не отпуская ее, “отбуксировать” мышь вниз на требуемое количество ячеек.

2. Использование формул

1. В ячейку A3 ввести значение 1 (начальное).

2. В ячейку A4 ввести формулу $=A3+1$.

3. Используя маркер заполнения, распространить (скопировать) формулу на остальные ячейки.

3. Использование прогрессии

В ячейке A3 набрать значение 1 (начальное) и ввести его, оставшись в этой же ячейке (для этого после набора нажать на клавишу **Enter**), а щелкнуть мышью на кнопке с “галочкой”, на панели инструментов расположенной справа от адреса ячейки). Можно, конечно, после нажатия на клавишу **Enter** вернуть указатель активной ячейки на A3.2.

Выбрать вкладку **Главная**, в группе Редактирование щелкнуть на пиктограмме **Заполнить** (↓) и выбрать **Прогрессия**.

В появившемся диалоговом окне **Прогрессия** (рис. 2):

— в поле **Предельное значение** указать значение 98 (в данном случае);

— в группе радиокнопок **Расположение** указать значение по столбцам (в данном случае);

— нажать кнопку **ОК**.

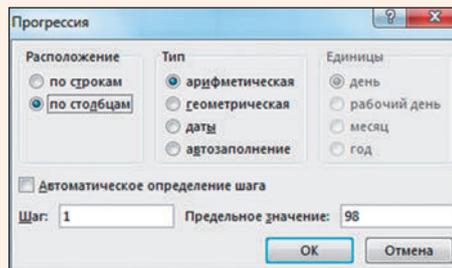


Рис. 2

Если перед вызовом меню выделить диапазон для ввода номеров, то предельное (последнее) значение задавать не нужно.

Другой способ использования прогрессии.

1. В ячейку A3 ввести значение 1 (начальное).

2. Поставить курсор мыши на маркер заполнения.

3. “Отбуксировать” мышь вниз на несколько ячеек, нажав ее правую кнопку.

4. После окончания выделения появится контекстное меню, в котором нужно выбрать пункт **Прогрессия** и в появившемся диалоговом окне с таким названием выполнить действия, описанные выше.

Если же в этом окне ничего не заполнять, а сразу щелкнуть на кнопке **ОК**, то порядковыми номерами будет заполнен выделенный диапазон.

Опишем также методику заполнения ячеек порядковыми номерами с точкой.

Пример. На листе должна быть представлена информация о результатах тестирования (рис. 3):

	A	B	C	...
1	Результаты тестирования			
2		Фамилия И.О.	Тест 1	
3	1.	Васильев Н.В.		
4	2.	Бойко К.С.		
54	52.	Абрамов А.И.		

Рис. 3

Решение

1. Получить в диапазоне A3:A54 порядковые номера одним из описанных выше способов.

2. Выделить соответствующий диапазон.

3. Установить в нем так называемый “пользовательский формат”.

Для этого на вкладке **Главная**, в группе **Ячейки** выбрать пункт **Формат | Формат ячеек** и в появившемся диалоговом окне **Формат ячеек** (см. рис. 4) — вкладку **Число**, а в ней в списке **Числовые форматы** — пункт **(все форматы)**.

В поле **Тип** в правой части окна ввести формат вида: $\#._)$

и щелкнуть на кнопке **ОК**.

Диалоговое окно **Формат ячеек** можно также вызвать, набрав комбинацию клавиш **Ctrl** + **1**.

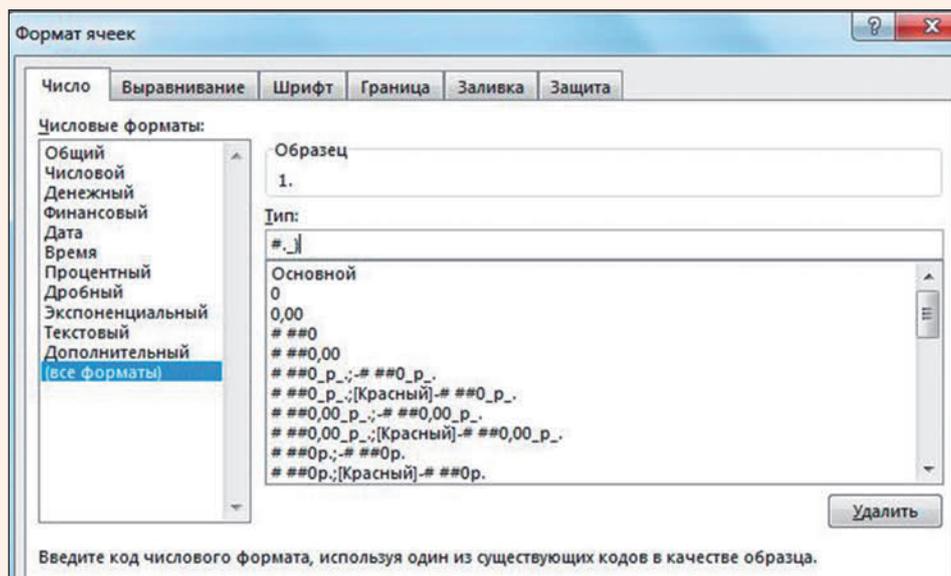


Рис. 4

ИСТОРИЯ ИНФОРМАТИКИ

Вычислительная машина Гана

В 70-е годы XVIII века в Германии вюртембергский пастор М.Ган сконструировал вычислительную машину, которая, в отличие от машин Паскаля и Лейбница, имевших вид продолговатых ящиков, представляла собой цилиндр. На верхнем основании цилиндра было расположено 14 пластинок с цифрами от 0 до 9. Им соответствовало столько же штифтов с теми же цифрами. В середине цилиндра была вставлена ось с ручкой. Поворотом этой ручки приводились во вращение ступенчатые валики, которые располагались вертикально. Машина Гана производила четыре арифметических действия, при этом результат не должен был превышать 14-значного числа.

Две машины Гана были на лондонской выставке в 1876 г. В отчете об этой выставке относительно машины Гана сказано: “Прибор священника Гана в Эхтердингене (Штутгарт) изобретен в 1770–1776 гг. Изготовлен (четвертый экземпляр) его сыном — придворным техником в Штутгарте в 1809 г., а в 1876 г. отправлен на выставку в Лондон герцогинею Урах”.



На этой же выставке была еще одна счетная машина, которая числилась как “арифметическая машина прошлого столетия”. Полагают, что это был первый экземпляр машины Гана. На выставку его привез директор Берлинской промышленной академии профессор Рело. Он приобрел эту машину в числе других вещей, оставшихся после кончины физика и химика Бейреса. К прибору было приложено старинное описание его, сделанное, по видимому, самим Ганом.

В описании прибора, опубликованном в 1878 г., было сказано: “У одного из здешних граждан находится прибор священника Гана, прибор этот действует вполне исправно”. Об этом приборе сам Ган говорит: “Когда я был занят вычислениями над колесами астрономических часов, то мне пришлось иметь дело с громаднейшими дробями и делать умножения и деления над весьма большими числами, от которых даже мои мысли останавливались, так что эта работа могла нанести ущерб моим прямым обязанностям. Тут я вспомнил, что когда-то читал о Лейбнице, что он занимался изобретением арифметической машины, на которую тратил много времени и денег, но удовлетворительного результата не достиг. У меня родилась мысль также поработать в этом направлении. Нечего говорить, что мною также потрачено много времени и средств над различными опытами и над устранением неудач и затруднений при проектировании и устройстве прибора. Наконец мне удалось устроить прибор достаточно совершенный и прочный. Более всего затруднений я встретил над изобретением способа переноса накопившихся десяти единиц на десятки”.

Литература

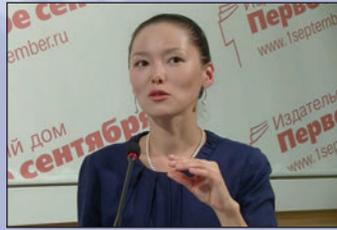
1. Апокин И.А., Майстров Л.Е. История вычислительной техники. М.: Наука, 1990.

НОВЫЙ ПРОЕКТ

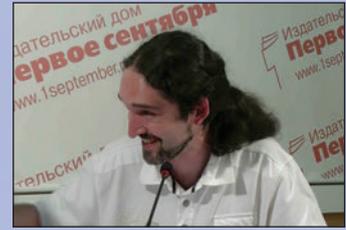
«Первого сентября»



А. Минченков
«Счастливая семья: миф или реальность, или Как построить гармоничные отношения в семье»



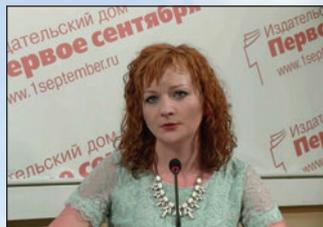
Т. Басанова
«Путь к успеху, или Как достичь успеха в личной жизни и в карьере»



О. Хухлаев
«Манипуляторы и манипуляции, или Как противостоять скрытому давлению»

СПЕЦИАЛИСТЫ-ПРАКТИКИ
СВИДЕТЕЛЬСТВО УЧАСТНИКА **О ВОСПИТАНИИ**
ОБ ОТНОШЕНИЯХ **О МАНИПУЛЯТОРАХ** **ОБ УСПЕХЕ**
АБОНЕМЕНТЫ **О САМООЦЕНКЕ** **УДОБНОЕ** **О ЧУВСТВЕ ВИНЫ**
О РАБОТЕ **О МИРОВОСПРИЯТИИ** **ВРЕМЯ** **О ДЕТЯХ**
ВЕБИНАРЫ
ОБ ИМИДЖЕ **ВОСТРЕБОВАННЫЕ**
ОБ ОПТИМИЗМЕ **О КОНФЛИКТАХ** **ТЕМЫ**
О СЕМЬЕ **ВИДЕОЗАПИСИ**
ДОСТУПНАЯ СТОИМОСТЬ **О ЛИЧНЫХ КРИЗИСАХ**
О ВЫГОРАНИИ **О КАРЬЕРЕ** **О ВЗАИМОПОНИМАНИИ** **О СТРЕССЕ**
ВОПРОСЫ И ОТВЕТЫ ОНЛАЙН **О ЦЕННОСТЯХ**

М. Чибисова
«Мифы и рифы демократического воспитания, или Как найти "золотую середину" в воспитании»



И. Телегина
«Компьютерный капкан, или Как помочь себе и близким людям не попасть в зависимость от компьютера»

Подробности смотрите на сайте
webinar.1september.ru