

“Мир информатики”

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 14, август 2017 г.

011000100010101001001111100100011001001000101001001001011

Школа программирования

*Когда человек хочет передвинуть гору,
он начинает с того, что убирает маленькие камни.*
Восточная мудрость

Типовые задачи обработки одномерных числовых массивов

Напомним (или расскажем), что массив — это структура данных, предназначенная для хранения однородной информации. Объясним.

Рассмотрим задачу: «Известен рост 20 человек. Определить среднее значение роста».

Для решения можно в программе использовать 20 переменных величин: p_1, p_2, \dots, p_{20} — и, обращаясь к каждой из них по имени, найти сумму значений роста, а затем среднее значение. Но такой вариант неудобен в случаях, когда количество значений роста большое, и невозможен, когда это количество на момент написания программы неизвестно.

Можно найти сумму так же, как решалась задача 1 в статье [1]:

```
сумма := 0
нц для i от 1 до n
    вывод нс, "Введите значение роста очередного человека"
    ввод p
    сумма := сумма + p
кц
```

Однако если нужно определить, сколько человек имеют рост больше среднего, то придется после расчета среднего роста (см. задачу 4 в [1]) повторно вводить в программу 20 значений. Это нецелесообразно. Желательно сохранить все введенные значения, а потом их использовать для расчетов. Для этого в программе и нужно использовать массив. Массив позволяет использовать для всех 20 значений роста общее имя `Рост`. Для всех семи цветов радуги можно использовать массив с именем `Радуга`.

Обратим внимание на то, что в начале статьи использовано слово «однородной». Под однородной информацией понимается информация одного типа (числовая, текстовая и т.д.) об объектах, *одинаковых по смыслу*. Например, информацию о весе тех же 20 человек хранить в одном массиве со значениями роста нецелесообразно — её следует хранить в другом массиве¹.

¹ В языке программирования Паскаль в таких случаях можно использовать один массив данных типа «запись».

Каждое отдельное значение массива называется «элемент массива». Элементы массива при хранении массива в памяти имеют уникальные порядковые номера – «индексы» (нумерация непрерывная). Чтобы обратиться в программе к значению некоторого элемента массива, надо указать имя массива и в квадратных скобках — индекс элемента. Например:

```
вывод нс, Радуга[3]
вывод нс, Радуга[k]
если Рост[13] > Рост[12]
  то
  ...
```

В данной статье рассмотрены методы решения двадцати типовых задач, которые встречаются при обработке одномерных числовых массивов (нахождение суммы их элементов, максимального значения и др.).

Для каждой рассмотренной задачи приведен фрагмент программы ее решения на школьном алгоритмическом языке. Используются следующие основные величины:

a — имя массива;

n — общее количество элементов массива (условно принято, что нумерация элементов массива начинается с 1).

Смысл остальных величин можно легко определить по их именам.

Принято, что массив описан и заполнен значениями.

1. Нахождение суммы всех элементов массива

```
сумма := 0
нц для i от 1 до n
  сумма := сумма + a[i]
кц
|Вывод результата или использование его в расчетах
...
```

Пример. В массиве записана масса каждого из *n* предметов, загружаемых в автомобиль. Определить общую массу груза.

Решение:

```
сумма := 0
нц для i от 1 до n
  сумма := сумма + a[i]
кц
вывод нс, "Общая масса всех предметов равна ", сумма
```

2. Нахождение суммы элементов массива с заданными свойствами (удовлетворяющих некоторому условию)

```
сумма := 0
нц для i от 1 до n
  если <условие>
    то
      сумма := сумма + a[i]
  все
кц
|Вывод результата или использование его в расчетах
...
```

Примечание. Здесь условие в условном операторе (команде **если**) может определяться значением элемента массива *a[i]* либо его индексом *i*.

3. Нахождение количества элементов массива с заданными свойствами

```
количество := 0
нц для i от 1 до n
  если <условие>
    то
      количество := количество + 1
  все
кц
|Вывод результата или использование его в расчетах
...
```

Примечание. В данном случае условие в условном операторе (команде **если**) определяется значением элемента массива $a[i]$. Количество элементов, зависящих от значения индекса i , может быть найдено без использования оператора цикла (убедитесь в этом!).

4. Нахождение среднего арифметического значения элементов массива с заданными свойствами

```
сумма := 0
количество := 0
нц для i от 1 до n
  если <условие>
    то
      сумма := сумма + a[i]
      количество := количество + 1
  все
кц
|Подсчет результата
среднее_арифметическое := сумма / количество
|Вывод результата или использование его в расчетах
...
```

Обращаем внимание на то, что многократно определять значение `среднее_арифметическое` в «теле» условного оператора (команды **если**) необходимости нет. Это можно сделать один раз после окончания оператора цикла. Однако может оказаться, что чисел, удовлетворяющих заданному условию, в массиве не окажется. В этом случае при расчете будет иметь место деление на ноль, что недопустимо. Правильное оформление:

```
...
|Подсчет и вывод результата
если количество > 0
  то
    среднее_арифметическое := сумма/количество
    вывод нс, "Среднее арифметическое: ", среднее_арифметическое
  иначе
    вывод нс, "Чисел, удовлетворяющих условию, в массиве нет"
все
```

5. Изменение значений элементов массива с заданными свойствами

```
нц для i от 1 до n
  если <условие>
    то
      a[i] := ...
  все
кц
```

6. Вывод на экран элементов массива с заданными свойствами

```
вывод нс, "Элементы массива, удовлетворяющие условию: "  
нц для i от 1 до n  
  если <условие>  
    то  
      вывод a[i], " "  
  все  
кц
```

Приведенный фрагмент имеет один недостаток — если в массиве нет элементов, удовлетворяющих заданному условию, то первая команда **вывод** будет излишней. Необходимо предварительно определить количество чисел в массиве с заданными свойствами (см. задачу 3), а затем оформить фрагмент в виде:

```
если количество > 0  
  то  
    вывод нс, "Элементы массива, удовлетворяющие условию: "  
    нц для i от 1 до n  
      если <условие>  
        то  
          вывод a[i], " "  
      все  
    кц  
  иначе  
    вывод нс, "Чисел, удовлетворяющих условию, в массиве нет"  
все
```

7. Нахождение номеров (индексов) элементов массива с заданными свойствами

```
вывод нс, "Номера элементов массива, удовлетворяющих условию: "  
нц для i от 1 до n  
  если <условие>  
    то  
      вывод i, " "  
  все  
кц
```

Здесь также можно учесть замечание, сделанное при решении предыдущей задачи.

8. Определение индекса элемента массива, равного некоторому числу

В приведенном ниже фрагменте программы величина значение — число, индекс которого (искомый_индекс) ищется.

```
нц для i от 1 до n  
  если a[i] = значение  
    то  
      искомый_индекс := i  
  все  
кц  
|Вывод результата или использование его в расчетах  
...
```

Прежде чем обсуждать представленный вариант, предлагаю читателям ответить на вопрос: индекс какого элемента будет найден, если в массиве окажутся несколько элементов, значение которых равно искомому?

Нетрудно заметить, что при таком оформлении в случае, если числа значение в массиве нет, результат будет неправильным. Чтобы учесть возможность такого случая, необходимо изменит фрагмент следующим образом:

```
искомый_индекс := 0
нц для i от 1 до n
  если a[i] = значение
    то
      искомый_индекс := i
  все
кц
|Вывод результата
если искомый_индекс > 0
  то
    вывод нс, "Индекс этого элемента: ", искомый_индекс
  иначе
    вывод нс, "Такого числа в массиве нет"
все
```

Однако и приведенный вариант нерационален — если искомое число находится, например, во втором элементе массива, все равно придется рассматривать весь массив. Желательно прекратить проверку элементов массива, как только встретится искомое значение. Для этого следует использовать оператор цикла с условием:

```
искомый_индекс := 1
нц пока искомый_индекс <= n и a[искомый_индекс] <> значение
  искомый_индекс := искомый_индекс + 1
кц
...
```

Но опять возникла уже встречавшаяся проблема — в случае, если числа значение в массиве нет, результат будет неправильным (каким именно — определите самостоятельно). Учесть возможность такого случая можно при выводе результата следующим образом:

```
...
|Вывод результата
если искомый_индекс <= n
  то
    вывод нс, "Индекс этого элемента: ", искомый_индекс
  иначе
    вывод нс, "Такого числа в массиве нет"
все
```

Интересно, что таким образом мы решили и другую задачу: «Определить, имеется ли в массиве элемент, равный некоторому значению». Её можно решить, также используя следующий фрагмент:

```
искомый_индекс := 1
есть := нет
нц пока не есть и искомый_индекс <= n
  если a[i] = значение
    то
      есть := да
    иначе Переходим к рассмотрению следующего элемента
      искомый_индекс := искомый_индекс + 1
  все
кц
```

```

если есть
то
    вывод нс, "Индекс этого элемента: ", искомый_индекс
иначе
    вывод нс, "Такого числа в массиве нет"
все

```

где `есть` — величина логического типа, принимающая истинное значение, если число значение встретится в массиве.

9. Определение индекса элемента массива, отсортированного в порядке возрастания, равного некоторому числу

Эта задача отличается от предыдущей тем, что заданный массив отсортирован. Это обстоятельство можно использовать для того, чтобы значительно уменьшить число проверяемых элементов. Идея метода аналогична идее отгадывания некоторого задуманного числа из интервала, например, от 1 до 99, путем задавания вопросов, на которые задумавший число отвечает «да» или «нет». Сначала надо спросить: «Ваше число больше 50?», а затем, в зависимости от ответа: «Ваше число больше 25?» или «Ваше число больше 75?» и т. д. При таком методе (который называется методом бинарного поиска) диапазон возможного нахождения заданного числа после каждого ответа уменьшается в 2 раза, в результате чего число вопросов, которые нужно задать для отгадывания числа, не превышает $\lceil \log_2 100 \rceil + 1$, где квадратные скобки соответствуют целой части числа, записанного между ними².

В программе решения задачи, реализующей метод бинарного поиска, используем следующие основные величины:

```

значение — число, индекс которого (искомый_индекс) ищется;
левая_граница — индекс начала части массива, рассматриваемой при поиске;
правая_граница — то же, конца рассматриваемой части массива;
середина — величина, равная среднему арифметическому значений левая_граница и
правая_граница.

```

Будем рассуждать так. Сначала имеем:

```
левая_граница := 1; правая_граница := n.
```

Находим значение величины `середина`:

```
середина := div(левая_граница + правая_граница, 2)
```

Далее рассматриваем три возможных варианта:

1) значение $> a[\text{середина}]$; в этом случае искомое число находится в правой половине массива (среди элементов с индексами от `середина + 1` до `правая_граница`);

2) значение $< a[\text{середина}]$; при этом искомое число находится в левой половине массива (среди элементов с индексами от `левая_граница` до `середина - 1`);

3) значение $= a[\text{середина}]$; это означает, что заданное число находится в массиве в ячейке с индексом `середина`, т. е. дальнейшие действия можно прекратить — задача решена (`искомый_индекс = середина`).

² Зависимость предельного числа вопросов для отгадывания числа из диапазона от 1 до n получите самостоятельно (оно не всегда равно $\lceil \log_2 n \rceil + 1$).

В двух первых вариантах необходимо аналогичные рассуждения применить к новой части массива и т. д. Действия прекращаются, когда левая_граница > правая_граница либо когда наступит третий из только что описанных вариантов (чтобы зафиксировать наступление этого варианта, в программе можно использовать величину логического типа `искомый_индекс_найден`).

Соответствующий фрагмент программы оформляется следующим образом:

```
левая_граница := 1; правая_граница := n
искомый_индекс_найден := нет
нц
  середина := div(левая_граница + правая_граница, 2)
  если значение > a[середина]
    то
      |Искомое число может находиться
      |только в правой половине массива
      левая_граница := середина + 1
    иначе
      если значение < a[середина]
        то
          |Искомое число может находиться только
          |в левой половине массива
          правая_граница := середина - 1
        иначе |значение = a[середина]
          искомый_индекс_найден := да
          искомый_индекс := середина
        все
      все
    кц при искомый_индекс_найден = да или левая_граница > правая_граница
  если искомый_индекс_найден
    то
      вывод нс, "Индекс этого числа равен "
      вывод искомый_индекс
    иначе
      вывод нс, "Такого числа в массиве нет"
  все
```

Самостоятельно определите, как будет работать описанная программа в случае, когда в массиве есть несколько элементов, равных заданному числу.

10. Определение максимального элемента массива

Алгоритм решения этой задачи аналогичен алгоритму действий человека, который определяет максимальное значение в некоторой одномерной таблице с числами. Сначала он смотрит в первую клетку таблицы и запоминает записанное там число. Затем смотрит во вторую клетку и в случае если имеющееся там число больше запомненного, то в качестве максимального запоминает новое число. Для остальных клеток действия аналогичны.

```
максимальное := a[1]
нц для i от 2 до n
  если a[i] > максимальное
    то
      максимальное := a[i]
    все
  кц
  |Вывод результата или использование его в расчетах
  ...
```

Примечание. Ясно, что при незначительном изменении приведенного фрагмента можно найти минимальный элемент массива.

Если минимальное число в массиве известно, то фрагмент программы решения задачи может быть оформлен так:

```
максимальное := минимальное
нц для i от 1 до n
  если a[i] > максимальное
    то
      максимальное := a[i]
  все
кц
```

где минимальное — минимальное число в массиве.

Так, например, если известно, что в массиве не все элементы отрицательные, то можем записать:

```
максимальное := 0
нц для i от 1 до n
  если a[i] > максимальное
    то
      максимальное := a[i]
  все
кц
```

11. Определение индекса максимального элемента массива

Здесь также алгоритм решения задачи аналогичен алгоритму действий человека, определяющего номер клетки с максимальным значением в некоторой одномерной таблице с числами — сначала он запоминает первое число и номер 1, а затем рассматривает второе число. Если оно больше того числа, которое помнил, то запоминает новое число и номер 2 и переходит к следующему, в противном случае просто переходит к следующему, третьему, числу и делает то же самое и т. д.

```
максимальное := a[1]
номер_максимального := 1
нц для i от 2 до n
  если a[i] > максимальное
    то
      максимальное := a[i]
      номер_максимального := i
  все
кц
|Вывод результата или использование его в расчетах
...
```

Интересно, что при решении задачи можно обойтись без использования величины максимальное. В самом деле, если нам известно значение индекса максимального среди рассмотренных элементов, то мы знаем и значение соответствующего элемента (оно равно *a*[номер_максимального]):

```
номер_максимального := 1
нц для i от 2 до n
  если a[i] > a[номер_максимального]
    то
      номер_максимального := i
  все
кц
```

Красиво, не правда ли?

12. Определение максимального значения среди тех элементов массива, которые удовлетворяют некоторому условию

Здесь возможны два случая:

- 1) известно, что числа, удовлетворяющие заданному условию, в исследуемом массиве имеются;
- 2) чисел, удовлетворяющих заданному условию, в обрабатываемом массиве может не быть.

В первом случае задача решается следующим образом:

```
максимальное := X
нц для i от 1 до n
  если <условие>
    то |Встретилось число, удовлетворяющее заданному условию
      |Сравниваем его с величиной максимальное
      если a[i] > максимальное
        то
          максимальное := a[i]
      все
    все
  кц
```

где X — число, о котором заведомо известно, что оно не превышает минимальное из чисел, для которых определяется максимальное значение.

Если же такое значение заранее неизвестно³, задача несколько усложняется. В приведенном далее фрагменте программы ее решения используется величина логического типа впервые, определяющая, впервые ли встретилось в массиве число, удовлетворяющее заданному условию. Остальные особенности работы программы описаны в комментариях.

```
впервые := да
нц для i от 1 до n
  если <условие>
    то |Встретился элемент массива,
      |удовлетворяющий заданному условию
      если впервые |Если впервые
        то
          максимальное := a[i] |Принимаем его в качестве
          |значения максимальное
          впервые := нет |Следующие такие числа уже будут
          |встречаться не впервые
        иначе |Если не впервые
          |Сравниваем число с величиной максимальное
          если a[i] > максимальное
            то
              максимальное := a[i]
          все
        все
  все
кц
```

Вывод результата или использование его в расчетах

...

Если же допускается, что чисел, удовлетворяющих заданному условию, в обрабатываемом массиве может не быть, то для решения рассматриваемой задачи может быть использован последний приведенный фрагмент программы:

³ Вопрос о том, можно ли всегда найти такое значение, исследуйте самостоятельно.

```

впервые := да
нц для i от 1 до n
...
кц

```

но вывод ответа при этом должен быть уточнен:

```

если впервые = да
то
    вывод нс, "Чисел, удовлетворяющих условию, "
    вывод "в массиве нет"
иначе
    вывод нс, "Искомое максимальное значение: ", максимальное
все

```

13. Определение места, которое заданное число M должно занимать в массиве, упорядоченном по возрастанию (рис. 1), при сохранении его упорядоченности (известно, что число M меньше последнего элемента массива)

2	4	5	12	20	24	31	45	50	53
---	---	---	----	----	----	----	----	----	----

$M = 15$

Рис. 1

Задача может быть решена несколькими способами. Один из них заключается в подсчете количества элементов массива, меньших M (см. задачу 3). Если это количество равно k , то искомое место (индекс) — $(k + 1)$ -е. Недостаток такого способа заключается в том, что приходится рассматривать все элементы массива, в то время как может оказаться, что искомое место находится в начале массива. Более рационально применить метод, который использует человек, решающий такую задачу, — он последовательно сравнивает первый, второй и т. д. элементы с числом M , и если рассматриваемый элемент меньше M , то переходит к следующему элементу, в противном случае — прекращает поиск (искомое место найдено).

В программе это оформляется так:

```

искомый_индекс := 1
нц пока a[искомый_индекс] < M
    искомый_индекс := искомый_индекс + 1
кц
| Вывод ответа
вывод нс, "Номер элемента, в котором должно находиться число ", M
вывод "равен ", искомый_индекс

```

Самостоятельно проанализируйте:

- 1) правильно ли будет работать программа в случае, когда число M меньше первого элемента массива;
- 2) можно ли начинать проверку элементов с конца массива.

Нетрудно увидеть, что в приведенном фрагменте для того чтобы найти нужное место, в среднем необходимы порядка $n/2$ проверок условия $a[\text{искомый_индекс}] < M$. Это количество можно существенно сократить, если для нахождения искомого места применить идею метода бинарного поиска, описанную при анализе задачи 9. Программу решения данной задачи с использованием этого метода разработайте самостоятельно.

Если же о числе M известно, что оно больше первого элемента массива, то проверку можно проводить и идя от конца массива:

```

искомый_индекс := n
нц пока a[искомый_индекс] > M
    искомый_индекс := искомый_индекс - 1
кц

```

Однако после окончания работы оператора цикла найденное значение величины `искомый_индекс` должно быть уточнено (почему — определите самостоятельно):

```
искомый_индекс := искомый_индекс + 1
| Вывод ответа
...
```

Ясно, что при этом число M не должно быть больше последнего элемента массива.

14. Обмен местами двух элементов массива с заданными номерами

Здесь, как и при обмене значениями двух «простых» величин, необходимо использовать вспомогательную переменную. Если индексы обмениваемых элементов — $m1$ и $m2$, то соответствующий фрагмент программы выглядит так:

```
вспомогательная := a[m1]
a[m1] := a[m2]
a[m2] := вспомогательная
```

15. Удаление из массива k -го элемента со сдвигом всех расположенных справа от него элементов на одну позицию влево

Операторы, осуществляющие сдвиг необходимых элементов:

```
a[k] := a[k + 1]
a[k + 1] := a[k + 2]
...
a[n - 1] := a[n]
```

можно оформить с использованием оператора цикла в виде:

```
нц для i от k до n - 1
    a[i] := a[i + 1]
кц
```

или

```
нц для i от k + 1 до n
    a[i - 1] := a[i]
кц
```

Внимание! После сделанных изменений следует учесть, что количество элементов исходного массива уменьшилось на 1.

Обращаем также внимание на прием, который был применен, — если сразу оператор цикла с параметром оформить сложно, следует выписать действия, являющиеся телом этого оператора, принять изменяющуюся величину в качестве параметра оператора цикла и определить его начальное и конечное значения. Этот прием будет использован и при решении нескольких следующих задач.

16. Перемещение элементов массива: первый элемент массива записать на место последнего, при этом сдвинув второй, третий, ..., последний элементы на одну позицию влево (рис. 2)⁴

5	12	8	0	14	5	1	3	23	4	8	0	9
---	----	---	---	----	---	---	---	----	---	---	---	---

а) исходный массив

12	8	0	14	5	1	3	23	4	8	0	9	5
----	---	---	----	---	---	---	----	---	---	---	---	---

б) конечный массив

Рис. 2

Ясно, что сразу записать в программе: $a[n] := a[1]$ нельзя (исходное значение $a[n]$ будет утеряно). Нельзя также сначала провести и сдвиг элементов влево ($a[1] := a[2]$ и т. д.). Как же быть? Правильно! Нужно предварительно запомнить значение первого элемента массива во вспомогательной переменной, после чего провести сдвиг необходимых элементов влево, а потом запомненное значение записать в последнюю ячейку массива:

```
вспомогательная := a[1]
нц для i от 1 до n - 1
  a[i] := a[i + 1]
кц
a[n] := вспомогательная
```

17. Вставка в массив заданного числа на k -е место со сдвигом k -го, $(k + 1)$ -го, $(k + 2)$ -го, ..., последнего элемента на одну позицию вправо

Прежде всего заметим, что для решения задачи массив должен быть описан с учетом дополнительного элемента. Еще одна особенность заключается в том, что в данном случае операторы, реализующие сдвиг элементов вправо, мы не можем записать следующим образом:

```
a[k + 1] := a[k]
a[k + 2] := a[k + 1]
...
a[n] := a[n - 1]
```

Почему не можем — понятно? Правильный вариант — сдвиг проводить, начиная с конца изменяемой части массива:

```
a[n] := a[n - 1]
a[n - 1] := a[n - 2]
...
a[k + 2] := a[k + 1]
a[k + 1] := a[k]
```

или, используя при этом оператор цикла:

```
нц для i от n до k + 1 шаг -1
  a[i] := a[i - 1]
кц
```

После сдвига можно в k -ю ячейку записать заданное число:

```
a[k] := ...
```

⁴ В общем случае задача формулируется так: « s -й элемент массива записать на место k -го, при этом сдвинув $(s + 1)$ -й, $(s + 2)$ -й, ..., k -й элементы на одну позицию влево ($s < k$)».

18. Перемещение элементов массива: последний элемент записать на место первого, при этом сдвинув первый, второй, ..., предпоследний элементы на одну позицию вправо (рис. 3)⁵

12	3	0	23	5	7	13	7	7	45	8	2	1
----	---	---	----	---	---	----	---	---	----	---	---	---

а) исходный массив

1	12	3	0	23	5	7	13	7	7	45	8	2
---	----	---	---	----	---	---	----	---	---	----	---	---

б) конечный массив

Рис. 3

Вспомнив решения задач 16 и 17, можем записать:

```

вспомогательная := a[n]
нц для i от n до 2 шаг -1
  a[i] := a[i - 1]
кц
a[1] := вспомогательная
  
```

19. Проверка массива на упорядоченность по неубыванию, т. е. определение того, верно ли, что каждый его элемент, начиная со второго, не меньше предыдущего

Здесь, как и при решении ряда предыдущих задач, всегда рассматривать весь массив нецелесообразно — желательно применить оператор цикла с условием, например, с постусловием. Условие окончания работы оператора: встретился элемент, меньший предыдущего, или рассмотрены все элементы массива:

```

i := 1
нц
  i := i + 1
кц_при a[i] < a[i - 1] или i > n
  
```

После этого ответ можно получить по значению величины i :

```

если i <= n
  то
    вывод нс, "Массив не упорядочен"
  иначе
    вывод нс, "Массив упорядочен"
все
  
```

20. Проверка наличия в массиве одинаковых элементов

Можно подсчитать количество в массиве каждого i -го элемента k_i (см. задачу 3). Если одинаковых элементов в массиве нет, что сумма всех таких количеств равна общему числу элементов массива (убедитесь в этом!). Соответствующая программа:

```

сумма_всех_количеств := 0
нц для i от 1 до n | Для каждого i-го элемента
  | определяем количество таких значений в массиве
  количество := 06
  нц для j от 1 до n | j - индексы элементов, с которыми
  
```

⁵ В общем случае задача формулируется так: “ s -й элемент массива записать на место k -го, при этом сдвинув k -й, $(k + 1)$ -й, ..., $(s - 1)$ -й элементы на одну позицию вправо ($s > k$)”.

⁶ Здесь начальное присваивание нулевого значения для каждого нового i является обязательным!

```

|сравнивается значение a[i]
если a[j] = a[i]
  то
    количество := количество + 1
все
кц
|Учитываем это количество в общей сумме
сумма_всех_количеств := сумма_всех_количеств + количество
кц
если сумма_всех_количеств = n
  то
    вывод нс, "В массиве нет одинаковых элементов"
  иначе
    вывод нс, "В массиве есть одинаковые элементы"
все

```

Количество сравнений значений элементов (которое в приведенной программе равно n^2) можно сократить в 2 раза, если каждый элемент сравнивать с элементами, расположенными правее него:

```

сумма_всех_количеств := 0
нц для i от 1 до n - 1
  количество := 0
  нц для j от i + 1 до n
    если a[j] = a[i]
      то
        количество := количество + 1
    все
  кц
сумма_всех_количеств := сумма_всех_количеств + количество
кц

```

В этом случае ответ определяется следующим образом:

```

если сумма_всех_количеств = 0    Обратите внимание на условие!
  то
    вывод нс, "В массиве нет одинаковых элементов"
  иначе
    вывод нс, "В массиве есть одинаковые элементы"
все

```

Но и новый вариант программы имеет недостаток, заключающийся в том, что, если в массиве одинаковыми являются, например, уже второй и третий элементы, рассматривать все равно приходится весь массив. Желательно прекратить проверку, как только встретятся два одинаковых элемента. Это можно сделать, используя в программе вместо операторов цикла с параметром операторы цикла с условием:

```

сумма_всех_количеств := 0
i := 1
нц пока i <= n - 1
  количество := 0
  j := i + 1
  нц пока j <= n
    если a[j] = a[i]
      то
        количество := количество + 1
    все
    j := j + 1
  кц
сумма_всех_количеств := сумма_всех_количеств + количество

```

```
i := i + 1
кц
```

Однако и в приведенном варианте по-прежнему рассматриваются все элементы. Чтобы в программе прекратить действия в нужный момент, введем переменную логического типа `есть_одинаковые` — как только встретятся два одинаковых элемента, присвоим ей значение `да`; при этом используем эту переменную в условиях:

```
i := 1; есть_одинаковые := нет
нц пока i <= n - 1 и не есть_одинаковые
  j := i + 1
  нц пока j <= n и не есть_одинаковые
    если a[j] = a[i]
      то
        есть_одинаковые := да
    все
  j := j + 1
кц
i := i + 1
кц
```

Искомый ответ можно получить по значению величины `есть_одинаковые`.

В заключение заметим, что еще одной достаточно распространенной задачей обработки числовых массивов является задача их сортировки, т. е. перерасположения элементов в некотором порядке, как правило, в порядке возрастания или убывания. Рассмотрению различных методов решения этой задачи будет посвящена одна из будущих публикаций.

Задачи для самостоятельной работы⁷

1. Известно сопротивление каждого из восьми элементов электрической цепи. Все элементы соединены последовательно. Определить общее сопротивление цепи.
2. Известны данные о стоимости каждого из 12 товаров. Найти общую стоимость тех товаров, которые стоят дороже 1000 рублей (количество таких товаров неизвестно).
3. Известны данные о количестве осадков, выпавших за каждый день января. Определить общее количество осадков, выпавших второго, четвертого и т. д. числа этого месяца. Оператор цикла с параметром, шаг изменения которого отличается от 1 или -1 , не использовать.
4. Известны оценки по информатике каждого из 22 учеников класса. Определить количество пятерок.
5. Известен рост каждого из 25 учеников класса. Рост мальчиков условно задан отрицательными числами. Определить средний рост мальчиков и средний рост девочек.
6. Все отрицательные элементы массива заменить на их абсолютную величину.
7. Вывести на экран элементы целочисленного массива, оканчивающиеся на 49.
8. Определить порядковые номера элементов массива, больших своих «соседей» (расположенных справа и слева от них).
9. Найти индекс элемента массива, равного 13. Если такого числа в массиве нет, то вывести на экран соответствующее сообщение.
10. В массиве записано общее число учащихся в каждой параллели школы (с 5-го по 11-й классы). Найти номер «самой старшей» параллели, в которой учатся более 120 учащихся.
11. В массиве хранится рост каждого ученика класса (значения расположены в порядке возрастания). Определить, есть ли в классе ученик, рост которого равен 170 см.
12. Известны расстояния от Москвы до 15 городов. Найти расстояние от Москвы до самого близкого от нее города из представленных в списке.

⁷ Все задачи решить с использованием массивов.

13. Известны данные о температуре воздуха в течение августа месяца. Определить максимальную температуру этого месяца.

14. Известны данные о количестве осадков, выпавших за каждый день февраля. Какого числа выпало самое большое число осадков? Если таких дней несколько, то должна быть найдена дата последнего из них.

15. Дан массив из 20 целых чисел. Определить максимальное четное число в нем. Рассмотреть два возможных случая:

- 1) при котором известно, что четные числа в массиве имеются;
- 2) при котором допускается, что четных чисел в массиве может не быть.

16. Решить задачу 13 для случаев, при которых:

1) число M может быть больше последнего элемента массива, но точно известно, что оно больше первого элемента;

2) число M может быть меньше первого элемента массива, но точно известно, что оно меньше последнего элемента.

17. Значения элементов массива упорядочены по убыванию. Определить, на какое место в массиве следует записать число 100, чтобы упорядоченность значений сохранилась. Известно, что минимальное число в массиве меньше 100.

18. В заданном массиве поменять местами имеющиеся в нем в единственном числе числа 5 и 15.

19. В массиве записаны значения роста 13 юношей класса. В середине учебного года ученик, рост которого был записан в третьей ячейке массива выбыл в другую школу. Учесть это изменение в массиве.

20. Изменить массив

1	3	10	70	35	47	53	57	58	65	68	72	81
---	---	----	----	----	----	----	----	----	----	----	----	----

таким образом, чтобы его элементы были расположены в порядке возрастания (при просмотре слева направо).

21. В массиве записаны значения роста каждого из 20 учеников класса. В середине учебного года в класс принят новый ученик, рост которого должен быть записан в пятую ячейку массива. Учесть это изменение в массиве.

22. Изменить массив

1	3	10	23	35	47	53	57	58	65	18	72	81
---	---	----	----	----	----	----	----	----	----	----	----	----

таким образом, чтобы его элементы были расположены в порядке возрастания (при просмотре слева направо).

23. Проверить, заданный массив на упорядоченность по невозрастанию, т. е. определить, верно ли, что каждый его элемент, начиная со второго, не больше предыдущего.

Литература

1. Типовые задачи обработки последовательности чисел. // «Мир информатики», № 3 (август 2016 г.).

При1ы

Диалог двух блондинок по телефону

— Люся! У меня компьютер не видит принтер. Я уже и монитор на него повернула, а он всё равно пишет, что его не видит. Что делать?

— Пальцем покажи!

Ответы, решения, разъяснения

к заданиям, опубликованным в журнале “Мир информатики” ранее

Головоломка “Карточная пирамида”

Напомним, что была приведена “пирамида” из 36 карт (нижние карты перевернуты “рубашкой” вверх) и следовало определить, какие две карты лежат по краям нижнего ряда, если известно, что в каждой из колонок нет карты той же масти или того же достоинства, что и перевернутая карта.

Решение

Прежде всего можно определить, какие карты находятся в нижнем ряду:



Из них две карты — дама пик и туз пик — не могут занимать других мест, кроме крайних. Они и являются искомыми.

Ответ: дама пик и туз пик.

Ответы представили:

— Тамошин Михаил, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

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

— Кукушкина София, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Лукичев Никита, г. Пенза, школа № 73, учитель **Диков А.В.**;

— Филиппова Елена, Республика Татарстан, Мамадышский р-н, совхоз “Мамадышский”, политехнический колледж, преподаватель **Порываева Н.С.**

Задание “Еще раз о нумерации домов”

Напомним, что было предложено (повторно) ответить на вопрос, на каком проспекте в Москве нумерация домов не соответствует общепринятым правилам (если встать в начале улицы, то с левой стороны должны быть дома с нечётными номерами, с правой — с чётными; в обоих случаях — в порядке возрастания).

Правильный ответ (Солнцевский проспект) прислали:

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

— Петрова Софья, школа № 1, г. Зеленокумск Ставропольского края, учитель **Букина Е.Ю.**;

— Чередниченко Николай, г. Пенза, школа № 73, учитель **Диков А.В.**

Учитывая сложность задания, редакция решила наградить Марию, Софью и Николая дипломами.

Дипломами будут награждены также Гашимов Геннадий и Султаншин Ансар, средняя школа г. Пионерский Калининградской обл. (учитель **Багрова О.А.**), правильно решившие задачу, опубликованную в мартовском выпуске журнала “Мир информатики”, связанную с системами счисления.

Поздравляем всех награжденных!

Ребусы по информатике, опубликованные в мартовском выпуске журнала “Мир информатики”, правильно решили:

— Исламов Тимур, г. Пенза, школа № 73, учитель **Диков А.В.**;

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

Толстоногова Дарья из Совхозной средней школы представила ответы на задание “Круглые зверушки”.

Пятеро друзей

В поездке пятеро друзей — Антон, Борис, Вадим, Дима и Гриша — познакомились с попугачицей. Они предложили ей отгадать их фамилии, причём каждый из них высказал одно истинное и одно ложное утверждение.

Дима сказал: «Моя фамилия — Мишин, а фамилия Бориса — Хохлов».

Антон сказал: «Мишин — это моя фамилия, а фамилия Вадима — Белкин».

Борис сказал: «Фамилия Вадима — Тихонов, а моя фамилия — Мишин».

Вадим сказал: «Моя фамилия — Белкин, а фамилия Гриши — Чехов».

Гриша сказал: «Да, моя фамилия Чехов, а фамилия Антона — Тихонов»

Какую фамилию носит каждый из друзей?

Банкир и конверты

Некому банкиру нужно было встретиться с важным клиентом, которому он должен выдать наличными заранее неизвестную сумму от 1 до 1 000 000 у. е. Чтобы не тратить время на отсчитывание денег, банкир дал указание своим кассирам заготовить некоторое количество конвертов с деньгами, на которых написаны содержащиеся в них суммы, чтобы потом просто отдать клиенту один или несколько конвертов, в которых и будет содержаться требуемая им сумма.

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

Ответы (можно не на все вопросы) присылайте в редакцию.

Литература

1. *Гаишков С. Б.* Занимательная компьютерная арифметика. Математика и искусство счета на компьютерах и без них. – М.: URSS, 2012.

Пять карточек

Имеются пять карточек, на которых записана одна из букв А, Б, Е, Ж, И. Надо составить последовательность из k карточек, соблюдая такие правила:

- 1) любая последовательность начинается буквой А;
- 2) после гласной буквы не может снова идти гласная, а после согласной — согласная;
- 3) буквы в последовательности не должны повторяться.

Запишите все допустимые последовательности карточек для:

- 1) $k = 3$;
- 2) $k = 4$;
- 3) $k = 5$.

Цифры и страницы

Для нумерации страниц в учебнике понадобилось 534 цифры. Сколько страниц в учебнике?

Как обычно, в этой рубрике проводится разбор задач, решение которых вызвало трудности.

Курение очень-очень вредно

Напомним условие: “Добрый доктор Айболит сказал Змею Горынычу, что если Змей будет выкуривать по 6 сигарет в день, то проживёт ещё 10 лет, а если по 17, то 5 лет. Сколько лет проживёт Змей Горыныч, если бросит курить?”

Решение (начало)

Примем, что годы одинаковой длины, а каждая выкуренная сигарета сокращает жизнь на одно и то же время.

Пусть в году x дней. Тогда, выкуривая по 6 сигарет в день, за 10 оставшихся ему жить лет Змей выкурит всего $6 \cdot 10x = 60x$ сигарет, и они сократят ему жизнь на столько, что ему останется прожить эти самые 10 лет. А если Змей будет выкуривать по 17 сигарет в день, то аналогично можем получить $17 \cdot 5x = 85x$ сигарет для 5 лет. Значит $85x - 60x = 25x$ выкуренных сигарет сокращают жизнь на $10 - 5 = 5$ лет, или каждые $5x$ сигарет — на 1 год. Можно также сказать, что каждые $5x$ “невкуренных” им сигарет продлевает ему жизнь на 1 год.

Итак, если Горыныч выкурит $60x$ сигарет, он проживет 10 лет; если $60x - 5x = 55x$ сигарет — $10 + 1 = 11$ лет; если $50x$ сигарет — $10 + 2 = 12$ лет, ...

Учитывая сказанное, самостоятельно найдите ответ на поставленный в условии вопрос и пришлите его в редакцию.

Задача “Кофе по утрам”

Напомним условие: “По утрам пятеро друзей (Макс, Дора, Боб, Иван и Борис) встречаются за чашкой кофе. Один из них выпивает в день одну чашку кофе, другой — 4, третий — 5, четвертый — 6, пятый — 8, однако кто-то пьет кофе без сахара, другие кладут в свою чашку по 1, 2, 4, 6 кусочков сахара, и несколько человек пьют свой кофе с молоком. Выясните, сколько чашек кофе выпивает за день каждый; сколько кусочков сахара он кладет в кофе, и кто пьет кофе с молоком, а кто — нет, если известно следующее:

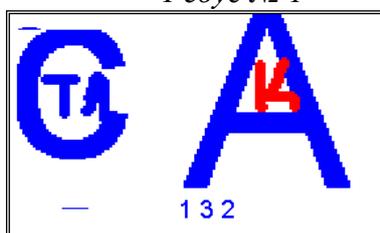
- а) Иван кладет в кофе втрое больше кусочков сахара, чем тот, кто выпивает за день 4 чашки кофе;
- б) трое, включая того, кто кладет в кофе 4 кусочка сахара, пьют кофе без молока;
- в) Боб выпивает только одну чашку кофе в день, пьет кофе без молока и без сахара;
- г) Дора пьет кофе и с молоком, и с сахаром;
- д) Макс, который пьет кофе без молока, кладет в него вдвое меньше кусочков сахара, чем тот, кто выпивает за день вдвое больше него кофе;
- е) Борис выпивает на 2 чашки кофе больше, чем Иван — но Иван кладет в кофе на два куска сахара больше, чем Борис”.

Благодаря Базанова Илью, Куклеву Лидию, Медведеву Анастасию и Серова Георгия, учеников школы № 11 г. Струнино Владимирской обл. (учитель Волков Ю.П.), и Мухитдинову Даяну, приславших правильные ответы, приведём начало решения.

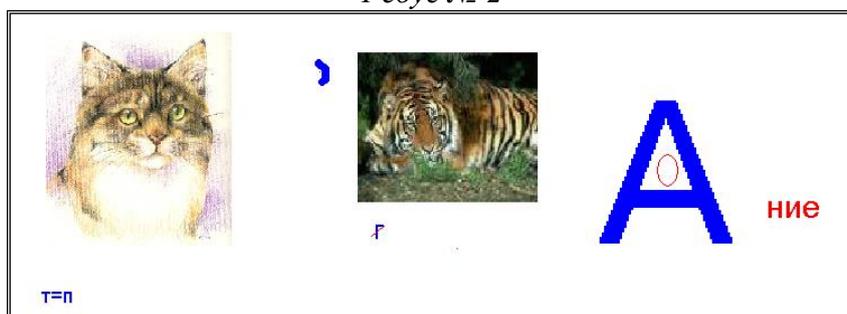
По условию, один из участников “кофепития” кладет в кофе втрое больше кусочков сахара, чем кто-то другой. Кто это может быть? Ответ — Иван (почему — подумайте). Одновременно можно установить, что тот, кто кладет в кофе втрое меньше кусочков сахара, чем Иван, выпивает за день ... чашки кофе. Число вместо многоточия установите самостоятельно и пришлите решение всей задачи в редакцию. Фамилии всех приславших правильные ответы будут опубликованы.

Ребусы по информатике

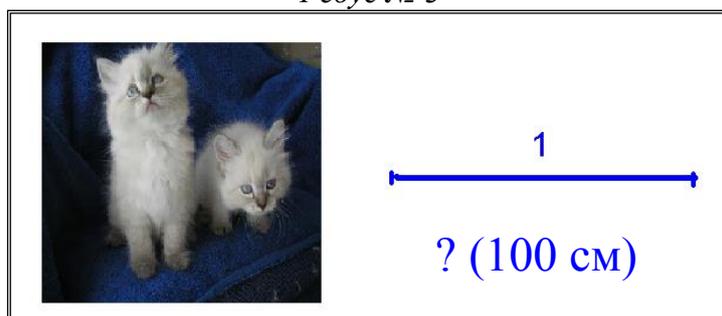
Ребус № 1



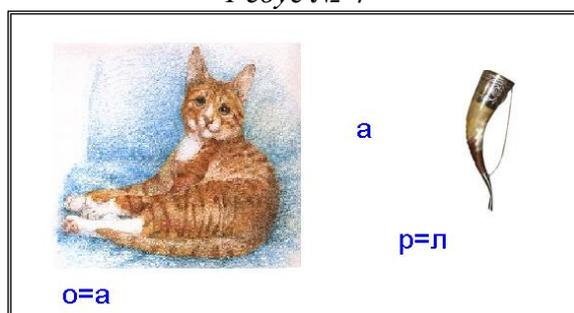
Ребус № 2



Ребус № 3



Ребус № 4



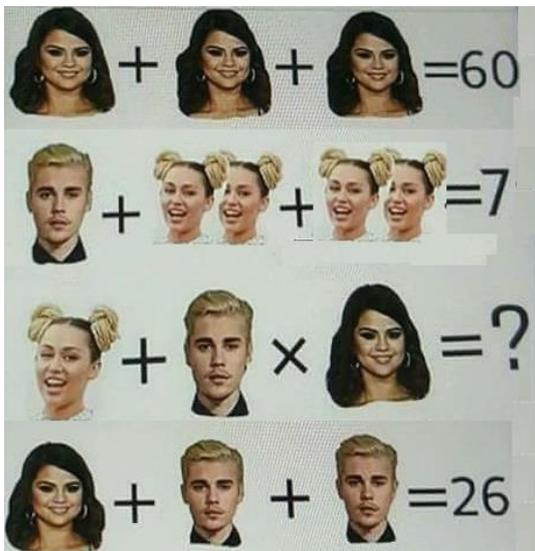
В ответах, пожалуйста, прокомментируйте найденные термины в контексте информатики.

Про дядю Фёдора

В день рождения дяди Фёдора почтальон Печкин хочет выяснить, сколько тому лет. Шарик говорит, что дяде Фёдору больше 11 лет, а кот Матроскин утверждает, что больше 10 лет. Сколько лет дяде Фёдору, если известно, что ровно один из них ошибся?

Числовой ребус из лиц (☺)⁸

Предлагаем учащимся 1 – 7-х классов определить число вместо знака вопроса и прислать ответ в редакцию.



Который час?

На циферблате электронных часов высвечивается текущий день месяца, номер месяца, часы, минуты, секунды:



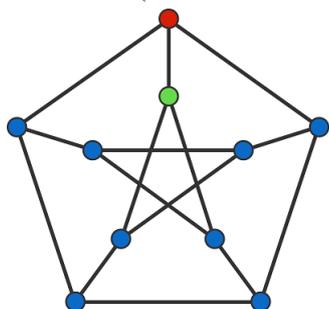
Однажды часы показывали такую дату и время, что:

- а) все 10 цифр были разные;
- б) номер месяца равен корню квадратному из числа секунд;
- в) число минут равно сумме номера месяца и количества секунд;
- г) номер дня равен половине количества минут.

Определите все значения.

От красной точки к зелёной

Через какое максимальное количество синих точек можно пройти по дороге от красной точки к зелёной при условии, что ни по какой линии между точками нельзя проходить дважды? (Можно ходить только по прямым линиям и синим точкам.)



Ответ обоснуйте, обозначив точки буквами и приведя соответствующий маршрут.

Одновременно предлагаем читателям ответить на вопрос: «Как называют конечное число точек на плоскости, соединённых отрезками линий?»

⁸ Ребус и следующую задачу предложил Валерий Рубанцев, <http://rvgames.de/ru/index.htm>.

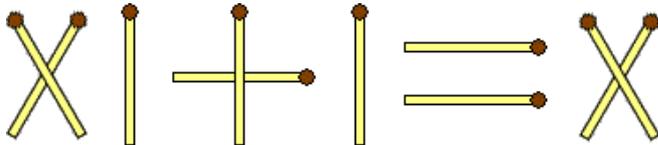
Числовые ребусы в троичной системе

В приведенных ниже ребусах зашифрованы цифры троичной системы счисления. Одинаковым буквам соответствуют одинаковые цифры. Звездочкой может быть любая цифра. Решите, пожалуйста, ребусы.

$$\begin{array}{r} 1. \\ * \\ + * \\ \hline * \end{array} \quad \begin{array}{r} 2. \\ * \\ + A \\ \hline A * \end{array} \quad \begin{array}{r} 3. \\ * \\ + B \\ \hline * * \end{array}$$

Как исправить ошибку?

На листе бумаги из спичек сложено “неправильное равенство”:



Как, переложив одну спичку, сделать так, чтобы равенство стало верным, вы, конечно, уже поняли. А как это сделать, не перекладывая, не поджигая и не ломая спички?

Фальшивая сторублевка (старинная задача)

Один господин зашел в часовой магазин и купил часы за 25 рублей. В уплату он отдал 100-рублевый кредитный билет. Так как у хозяина магазина не было мелких денег, чтобы отдать сдачу, он послал эту сторублевку соседу для размена. Получив мелкие деньги, он отдал покупателю 75 рублей сдачу, и тот ушел. Через некоторое время к хозяину пришел сосед и заявил, что разменянная им сторублевка оказалась фальшивой. Тогда хозяин отдал ему настоящую сторублевку.

Сколько денег потерял хозяин часового магазина?

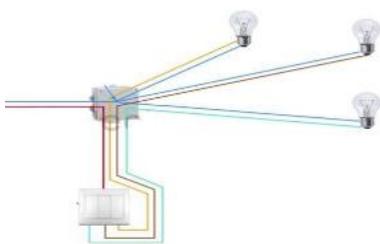
Вкусные ломтики

Мама очень вкусно поджаривает ломтики хлеба, пользуясь специальной маленькой сковородкой. Для готовности каждый ломтик должен быть поджарен с двух сторон. Поджаривание каждой стороны ломтика длится 30 секунд, причем на сковороде умещается рядом только два ломтика.

За какое минимальное время при этих условиях мама может приготовить:

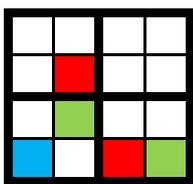
- а) четыре ломтика хлеба?
- б) пять ломтиков?

Три лампочки



Есть две изолированные друг от друга комнаты. В одной из них расположены три лампочки, в другой — трехклавишный выключатель к этим лампочкам, каждая клавиша которого включает одну и только одну из них, но какую именно — неизвестно. Изначально все лампочки выключены. Требуется за минимальное число переходов из одной комнаты в другую определить, какая клавиша какой лампочкой управляет. Как это сделать?

Цветные клетки



На клетчатом поле размером 4×4 , разбитом на четыре квадрата размером 2×2 , некоторые клетки закрашены. Необходимо закрасить оставшиеся клетки так, чтобы в каждой строке, в каждом столбце поля и в каждом из четырех квадратов размером 2×2 все цвета были разными. Четвертый цвет выберите сами.

Внимание! Конкурс!

Конкурс № 7

В качестве задания этого конкурса предлагаем решить задачу: “Число $A = (32^{32} + 4^4 - 1) \times 16^{16} + 8^8 - 1$ перевели из десятичной системы счисления в двоичную. Сколько единиц получилось в двоичной записи числа?”

Ответы с обоснованием направьте в редакцию по адресу mirinfo@infojournal.ru. Срок представления ответов — 30 сентября.

Конкурс № 8

В качестве задания этого конкурса предлагаем выполнить задания (можно не все) для самостоятельной работы, предложенные в первой статье данного выпуска. Срок представления программ не ограничен.

Пожалуйста, укажите в письме свои фамилию и имя, населённый пункт и номер школы, а также фамилию, имя и отчество учителя информатики.

Итоги конкурса № 5 “Круглые зверушки”

Напомним, что было предложено нарисовать изображения так называемых “круглых зверушек” — изображения зверей, полученные из кругов.

Победителями конкурса признаны:

— Алиев Тимур, Желтоухов Владислав, Королёва Марианна, Масимов Тимур, Родионов Александр и Слостухина Анастасия, Владимирская обл., г. Александров, Детский клуб “Крошка Ру”, преподаватель **Волков А.Ю.**;

— Аксёненко Ирина и Чумаков Илья, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Антонова Анна и Знаменкина Вера, Санкт-Петербург, лицей № 445, учитель **Зорина Е.М.**;

— Ананьев Александр, Волошин Марк, Дощик Константин, Лазуренко Глеб, Семёнов Дмитрий, Сушенцова Анастасия и Чунин Павел, средняя школа г. Пионерский Калининградской обл., учитель **Багрова О.А.** (все они получили изображения, разработав компьютерную программу);

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

— Филимонова Галина, г. Пенза, школа № 73, учитель **Диков А.В.**

Все перечисленные читатели будут награждены дипломами. Поздравляем!

Учебник по программированию на языке Python

В издательстве ДМК-Пресс вышла книга Д.М. Златопольского «Основы программирования на языке Python» (<http://dmkpress.com/catalog/computer/programming/python/978-5-97060-552-3/>)



Эта книга для тех, кто хочет научиться программировать на языке Python, который в последнее время становится популярным у нас в стране и за рубежом. Её отличие в том, что в ней главное – не описание языка программирования, которое представлено в большинстве книг по программированию. В данной книге рассматриваются особенности разработки программ, описываются методы решения типовых задач, встречающихся при разработке, распространенные алгоритмы, даются советы и т. п. Это учебник, в котором в доступной форме излагаются вопросы, с которыми сталкивается человек, начинающий осваивать этот непростой, но захватывающий и очень интересный процесс – программирование.

В первой части книги (разделы 1 – 6) рассматриваются особенности разработки компьютерных программ и соответствующие инструкции языка Python. После их изучения читатель сможет разработать ряд простейших игр, описанных в разделе 7. В завершающей части (разделы 8 – 16) рассматриваются основные структуры данных языка Python (строки, списки, словари), вопросы, связанные с работой с файлами, а также с совершенствованием программ на основе использования функций.

По всем темам, рассмотренным в книге, приводятся большое число примеров и задач, подробная методика их решения и соответствующие программы с комментариями. При подборе задач автор опирался на Кодификатор элементов содержания и требований к уровню подготовки выпускников образовательных организаций для проведения ЕГЭ по информатике и ИКТ.

В приложении впервые системно и популярно описаны особенности разработки программ на языке Python с графическим пользовательским интерфейсом.

С оглавлением книги можно ознакомиться по адресу: <http://dmkpress.com/files/PDF/978-5-97060-552-3.pdf>

Безграмотными в 21 веке будут не те,
кто не умеет читать и писать, а те,
кто не умеет учиться, разучиваться
и переучиваться.

Элвин Тоффлер