

# «Мир информатики»

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 45, май 2020 г.

011000100010101001001111100100011001001000101001001001011

## Школа программирования

Редакцию начинает публикацию фрагментов книги Дмитрия Михайловича Златопольского «Программирование для школьников», которая готовится к изданию в издательстве «Ленанд».

## Программирование для школьников

### Введение

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

Книга, которую вы читаете, предназначена для тех, кто хочет научиться программировать. В ней системно, подробно и доступно излагаются вопросы, с которыми сталкивается человек, начинающий осваивать этот непростой, но очень захватывающий и интересный процесс – программирование. Эти вопросы – общие для всех языков программирования, так что вы можете пользоваться книгой независимо от того, какой язык изучаете.

Сначала в главе 1 рассказывается о важных понятиях программирования – «алгоритм», «программа», «транслятор» и др., после чего в главе 2 описываются особенности основных операторов языков программирования (в том числе наиболее популярных среди школьников языков Паскаль и Python).

В главе 3 рассмотрены вопросы, которые понадобятся при решении типовых задач программирования, а в главе 4 описываются методы решения ряда простых таких задач.

Ознакомившись с содержанием глав 3 и 4, читатель сможет разработать ряд простейших игр. Методика их разработки описана в главе 5.

После этого в главах 6–11 рассматриваются другие типовые задачи программирования (в том числе связанные с обработкой массивов и строк), а в главе 12 – рассказывается о методах совершенствования программ на основе использования функций и процедур.

В приложении 1 освещен еще ряд вопросов, знать которые также полезно начинающему программисту, а в завершающих приложениях 2 и 3 описаны методики разработки программ шифрования текста несколькими простейшими методами и программы, моделирующей очень интересную игру.

В книге предлагаются контрольные вопросы по темам и большое число заданий на самостоятельную разработку компьютерных программ. Обращается внимание на типовые ошибки, которые часто допускают начинающие программисты, и дается ряд полезных советов, связанных с разработкой программ.

### *Как пользоваться книгой*

Методика решения типовых задач программирования описывается с использованием так называемого «школьного алгоритмического языка» (система программирования КуМир [4]). Русский синтаксис этого языка и большое число комментариев делает методику и приводимые программы максимально понятными, и вы легко сможете разработать аналогичные программы на языке программирования, который изучаете. Автор рекомендует это делать после знакомства с тем или иным разделом книги.

И еще. Если вы хотите стать программистом, то должны знать, что одним из условий этого является большой опыт решение задач по программированию. Поэтому решайте задачи, абстрактные и содержательные, на 5 минут и на день работы. Все это обязательно вам пригодится. Чем больше опыт решения задач, тем больше вероятность того, что уже решенная задача или аналогичная ей встретится вам в будущем.

Заметим также, что многие рассмотренные в книге задачи предусмотрены Кодификатором элементов содержания и требований к уровню подготовки выпускников общеобразовательных учреждений для проведения единого государственного экзамена по информатике и ИКТ и которые используются в демонстрационных и пробных вариантах ЕГЭ по информатике последних лет, а также представлены в пробных вариантах ЕГЭ в компьютерной форме.

Книга, безусловно, будет полезна и читателям, уже имеющим определенный опыт программирования, а также учителям школ и преподавателям колледжей и ВУЗов.

## **Глава 1. Понятия «алгоритм» и «программа»**

*Алгоритм решения задачи* – точное описание порядка действий, которые надо выполнить для решения задачи.

Приведем ряд примеров.

Вот старинная задача: «Однажды крестьянину понадобилось перевезти через реку волка, козу и капусту. У крестьянина есть лодка, в которой может поместиться, кроме самого крестьянина, только одно существо или предмет – или волк, или коза, или капуста. Если крестьянин оставит без присмотра волка с козой, то волк съест козу; если крестьянин оставит без присмотра козу с капустой – коза съест капусту. Как крестьянину перевезти на другой берег все свое имущество в целостности и сохранности?»

Чтобы решить задачу, крестьянин должен действовать так:

- 1) погрузить на лодку козу;
- 2) перевезти ее на другой берег;
- 3) выгрузить ее;
- 4) вернуться;
- 5) погрузить на лодку волка;
- 6) перевезти его на другой берег;
- 7) выгрузить его;
- 8) погрузить на лодку козу;
- 9) вернуться с ней на первый берег;
- 10) выгрузить ее;
- 11) погрузить на лодку капусту;
- 12) перевезти ее на другой берег;
- 13) оставить капусту на этом берегу;
- 14) вернуться на первый берег;
- 15) погрузить на лодку козу;
- 16) перевезти ее на другой берег;
- 17) выгрузить козу;
- 18) остаться на втором берегу.

Возможен и второй вариант решения задачи (найдите его). Самостоятельно составьте также алгоритм решения такой задачи: «К реке подошла группа из 20 солдат, которым нужно переправиться на другой берег. Река глубокая и бурная, и ее без лодки переплыть невозможно. Солдаты увидели двух мальчиков с лодкой. Лодка такова, что в ней размещается только один солдат, только один мальчик или только два мальчика. Как солдатам переправиться?»

Можно говорить об алгоритме приготовления торта в домашних условиях (только в этом случае мы алгоритм называем «рецептом»).

Пример из математики. Представьте, что товарищ принес вам чертеж прямоугольника и просит сообщить ему площадь этой фигуры. Какие действия вы должны выполнить, чтобы решить поставленную перед вами задачу? Вот ответ:

- 1) узнать (измерить) длину одной из сторон прямоугольника и запомнить (или записать) её;
- 2) узнать (измерить) длину второй стороны фигуры и запомнить (или записать) её;
- 3) рассчитать площадь, перемножив два найденных значения;
- 4) сообщить результат товарищу.

Каждый алгоритм рассчитан на какого-то *исполнителя* – человека (группу людей) или устройство (робот и др.), который выполняет действия, описанные в алгоритме.

### Задание

Представьте, что на стержень *A* надеты три диска:

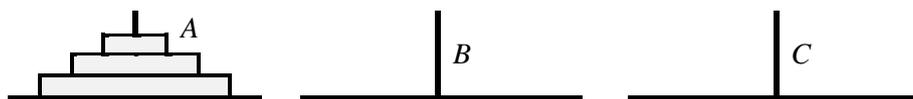


Рис. 1.1

Задача состоит в том, чтобы перенести диски со стержня *A* на стержень *C*, используя стержень *B* как промежуточный. При этом должны соблюдаться три условия:

- 1) за один «ход» можно переносить лишь один диск;
  - 2) нельзя класть больший диск на меньший;
  - 3) снятый диск нельзя отложить в сторону – он должен быть надет на один из стержней.
- Опишите алгоритм решения задачи.

Понятие «*программа решения задачи*» возникает, когда речь идет о решении задачи на компьютере. Программа решения задачи – это алгоритм решения данной задачи, записанный в виде (на языке), «понятном» данному компьютеру. А какой язык «понимает» компьютер? В нем, как в техническом устройстве, информация может быть представлена в двоичном виде – в виде последовательности условных нулей и единиц. Значит, и описание действий, которые нужно выполнить для решения (в программировании их, как правило, называют «операторами» или «инструкциями»), должно поступать в компьютер для выполнения в двоичном виде. Поэтому в первых электронных компьютерах программы представляли из себя последовательность двоичных чисел<sup>1</sup>:

1010 1101 0101 1000 ...

Представляете, как трудно было программисту найти нужное место в программе, чтобы что-то изменить или добавить, найти и исправить ошибку? Чтобы устранить этот недостаток, были разработаны так называемые «языки программирования высокого уровня» (ЯПВУ)<sup>2</sup> – Фортран, Алгол, Бейсик и др. Программы на ЯПВУ оформляются в привычном человеку<sup>3</sup>

<sup>1</sup> Так как числа в двоичной системе «длинные», то для записи программ использовалась также восьмеричная или шестнадцатеричная система счисления.

<sup>2</sup> Мы не рассматриваем использование для разработки программ языков ассемблера, имевшее место до применения ЯПВУ.

<sup>3</sup> ЯПВУ поэтому так и называются, что они, так сказать, «ориентированы» на человека, в отличие от машиноориентированных языков (языков двоичных кодов и ассемблеров).

виде: числа записываются как десятичные, а команды на выполнение действий (их мы будем называть «операторами») и другие служебные слова – на естественном (пусть, как правило, на иностранном, но достаточно понятном) языке: PRINT, IF, BEGIN и т. п. Однако при этом получается противоречие – человеку-программисту удобно разрабатывать и читать программу, но компьютер такую программу не «поймет»!

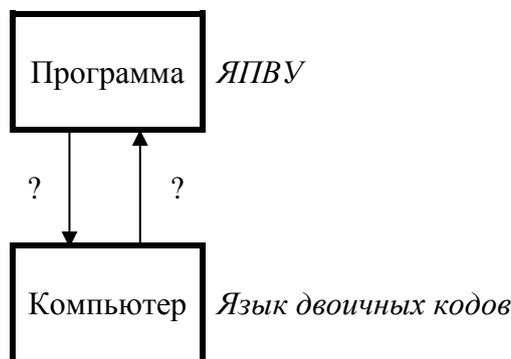


Рис. 1.2

Как же поступить, чтобы компьютер мог выполнять программу на ЯПВУ? А так, как можно общаться двум людям, один из которых знает только китайский язык, а второй – только русский: нужен переводчик! Поэтому во всех современных системах программирования предусмотрен транслятор – системная программа, осуществляющая перевод прикладной программы, написанной программистом на ЯПВУ, в язык машинных кодов и ее выполнение (см. рис. 1.3)<sup>4</sup>.

Существуют два вида трансляторов:

- 1) интерпретаторы;
- 2) компиляторы.

Интерпретатор читает очередной оператор программы и сразу его выполняет, не переводя всю программу в машинный код.

Компилятор читает всю программу целиком, делает ее перевод и создает законченный вариант программы на «машинном языке», который затем и выполняется.

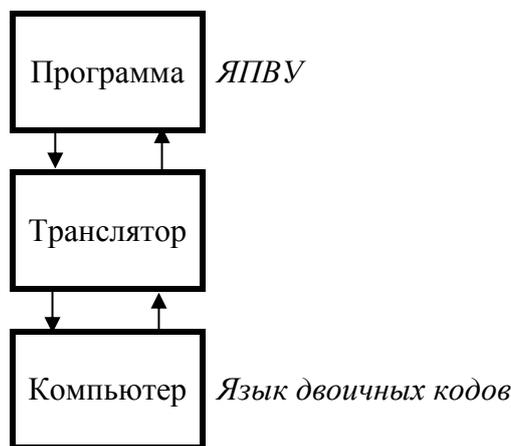


Рис. 1.3

Вопрос о преимуществах и недостатках каждого варианта транслятора здесь обсуждать не будем<sup>5</sup>.

В качестве приведем программы решения задачи расчета площади любого прямоугольника на трех ЯПВУ – так называемом «школьном алгоритмическом языке»:

<sup>4</sup> Транслятор обычно выполняет также диагностику ошибок и др.

<sup>5</sup> Подумайте над этим вопросом.

```

алг Расчет площади
нач вещь дл, шир, площ
| Ввод размеров прямоугольника
вывод нс, "Введите длину прямоугольника в см "
ввод дл
вывод нс, "Введите ширину (высоту) прямоугольника в см "
ввод шир
| Расчет площади
пл := дл * шир
| Вывод ответа на экран
вывод нс, "Площадь прямоугольника равна ", пл, " кв. см"
кон

```

на языке Паскаль:

```

var dl, shir, pl: integer;
BEGIN
{ Ввод размеров прямоугольника }
write('Введите длину прямоугольника в см ')
readln(dl);
write('Введите ширину (высоту) прямоугольника в см ');
readln(shir);
{ Расчет площади }
pl := dl * shir;
{ Вывод ответа на экран }
writeln('Площадь прямоугольника равна ', pl, ' кв. см')
END.

```

и на языке Python:

```

# Ввод размеров прямоугольника
print("Введите длину прямоугольника в см")
dl = int(input())
print("Введите ширину (высоту) прямоугольника в см")
shir = int(input())
# Расчет площади
pl = dl * shir
# Вывод ответа на экран
print("Площадь прямоугольника равна", pl, "кв. см")

```

Записи, начинающиеся символом «|» (в программе на школьном алгоритмическом языке и «#» (в программе на языке Python), а также обрамленные фигурными скобками ({ и }) в программе на языке Паскаль называют «комментариями». Они помогают читающему текст программы (в том числе и ее автору!) понять логику ее построения, смысл тех или иных действий и т. п. Опытные программисты говорят, что «много комментариев не бывает» ☺.

В настоящее время существует большое число ЯПВУ. Чтобы сделать процесс разработки программ удобным для вас и других программистов, для каждого языка разработаны *системы программирования*, включающие транслятор, текстовый редактор (позволяющий копировать, удалять и перемещать фрагменты программ), справочную систему, средства отладки (поиска ошибок в программе) и др.

Автор надеется, что читатель знает, какую систему программирования он использует, умеет запускать её, знает правила записи и выполнения программ. Кроме того, предполагается, что читатель знает, как выводить на экран текст и числа.

### **Контрольные вопросы**

1. Что такое «алгоритм решения задачи»?
2. Какие вы знаете способы записи алгоритма?
3. В чем особенности алгоритма, который называют «программой»?

4. Почему языки программирования высокого уровня так называются?
5. Что такое «транслятор»? Какие функции он выполняет?
6. Какие виды трансляторов вы знаете? В чем особенность каждого вида?
7. Что включает в себя система программирования?

## Глава 2. Основные операторы языков программирования

Во всех современных языках программирования имеются следующие основные операторы:

- оператор вывода информации на экран;
- оператор ввода данных в программу – см. раздел 2.1;
- оператор присваивания – см. раздел 2.1;
- условный оператор (несколько вариантов) – см. раздел 2.2;
- оператор цикла (также несколько вариантов) – см. раздел 2.3;
- оператор вызова процедуры – см. главу 12.

В данной главе рассмотрены особенности некоторых из них.

### 2.1. Переменные величины. Ввод данных в программу. Оператор присваивания

Переменные величины (или, короче, переменные) – величины, которые при выполнении программы могут принимать различные значения<sup>6</sup>. Эти значения (в любой момент времени – какое-то одно) хранятся в памяти компьютера и могут быть использованы<sup>7</sup>, поэтому говорят, что переменные используются в программе для хранения информации.

Вспомним «математическую» программу из главы 1. В ней ее пользователь должен ввести размеры сторон прямоугольника, которые могут быть любыми; эти значения будут использованы для расчета площади. Значение площади, которое тоже может быть любым, надо запомнить, а затем вывести на экран. Можем сказать, что в этой программе используются три переменные.

Каждая переменная величина характеризуется именем и типом.

В именах переменных можно использовать латинские буквы<sup>8</sup>, цифры (но имя не может начинаться с цифры) и знак подчеркивания «\_». Примеры: n,

#### СОВЕТ № 1

Желательно давать переменным «говорящие» имена, чтобы можно было сразу понять, зачем нужна та или иная переменная. Например, имя `diskr` помогает понять, что эта переменная хранит значение дискриминанта квадратного уравнения.

Есть набор слов, которые нельзя использовать в качестве имен переменных, так как эти слова «зарезервированы» в языке для определенных целей (эти слова называют «зарезервированными», или «служебными», или «ключевыми»). С перечнем таких слов в используемом вами языке программирования ознакомьтесь самостоятельно.

Основные типы данных приведены в табл. 2.1.

<sup>6</sup> Существуют также так называемые «константы» – так в программировании называют величины, значения которых во время выполнения программы не меняются.

<sup>7</sup> О том, как используются значения переменных, вы, наверное, уже поняли из приведенных в главе 1 примеров их вывода на экран, – по имени (**вывод** пл; `print (pl)` и т. п.).

<sup>8</sup> В школьном алгоритмическом языке можно и русские.

Таблица 2.1

Тип	К нему относятся	Примеры	Примечание
Целый	Целые числа (положительные и отрицательные, а также 0)	4, -45, 0, 687	
Вещественный	Вещественные числа <sup>9</sup> (могут быть с дробной частью)	1.45, 0.00453, -3.789,	Видно, что разделителем целой и дробной частей вещественного числа является точка
Логический	Величины, которые могут принимать значения True (Истина, <b>да</b> ) или False (Ложь, <b>нет</b> )		Величины логического типа возможны не во всех языках программирования (см. раздел 3.3).
Строковый	Последовательность (строка) символов, в том числе один символ или пустая строка (""/'')	"Школа", 'красный', "ф", ""	Вид кавычек зависит от используемой системы программирования

Тип величины определяет:

– какие значения может принимать величина (область допустимых значений переменной), например, значением переменной строкового типа может быть только строка (последовательность) символов;

– какие операции можно проводить над переменной (множество допустимых операций с величиной); например, над величинами вещественного типа можно проводить операции сложения, вычитания, умножения, деления и возведения в степень<sup>10</sup>, а над величинами целого типа – еще две операции (см. далее);

– какой объем памяти компьютера требуется для хранения значения данной переменной и в каком формате будут храниться данные. Например, переменные вещественного типа, как правило, занимают 8 байтов.

Значение переменной хранится в каком-то месте памяти, которое можно смоделировать в виде прямоугольника (ячейки), рядом с которым указано имя этой переменной:



Рис. 2.1

Как значение «попадает» в соответствующий прямоугольник? Это происходит с помощью двух операторов:

- 1) оператора присваивания;
- 2) оператора ввода данных.

Начнем со второго. Он используется для ввода данных в программу в ходе ее выполнения с помощью клавиатуры (как говорят в программировании – «с клавиатуры»).

Язык программирования	Оператор
Школьный алгоритмический язык	<b>ввод</b>
Паскаль	readln, read
Python	input ()

<sup>9</sup> Их называют также «числами с плавающей точкой» (или «числами с плавающей запятой»).

<sup>10</sup> В языке программирования Паскаль операция возведения в степень не предусмотрена.

Например, чтобы ввести значение переменной *a* строкового типа, нужно записать в программе:

Язык программирования	Оператор
Школьный алгоритмический язык	<b>ввод</b> <i>a</i>
Паскаль	<code>readln(a)</code>
Python	<code>a = input()</code>

При выполнении этой строки на экране появится курсор, и система будет ожидать ввода значения с клавиатуры. Когда пользователь введет его и нажмет клавишу **<Enter>**, система запишет это значение в память в переменную *a*. До нажатия клавиши **<Enter>** можно удалять символы с помощью клавиши **<Backspace>**.

В программах на школьном алгоритмическом языке и языке Паскаль ввод значений переменных *числового* типа проводится с помощью операторов, указанных в таблице чуть выше. В языке Python имеется особенность выполнения оператора `input`. Дело в том, что при нажатии клавиши на клавиатуре (в том числе клавиш с цифрами) в компьютер поступает код клавиши, то есть код соответствующего символа (об этом будет рассказано в главе 10). И входные данные воспринимаются оператором `input` именно как последовательность символов, в результате чего переменная получит значение строкового типа, даже если была введена последовательность цифр. Чтобы преобразовать полученное значение в целое число, оператор должен быть оформлен так:

```
a = int(input())
```

*a* в вещественное число:

```
a = float(input())
```

### **Внимание!**

Во многих языках программирования все используемые в программе переменные величины должны быть предварительно объявлены (описаны) с указанием их типа (см. программу Расчет площади в главе 1). С особенностями объявления переменных в используемом вами языке программирования ознакомьтесь самостоятельно.

Когда приходится вводить значения нескольких величин, целесообразно перед оператором ввода записать оператор вывода на экран с сообщением-подсказкой<sup>11</sup>, чтобы пользователь программы знал, какое значение вводится в тот или иной момент ее выполнения:

Язык программирования	Оператор
Школьный алгоритмический язык	<b>вывод</b> <i>нс</i> , "Введите фамилию: " <b>ввод</b> <i>фам</i> <b>вывод</b> <i>нс</i> , "Укажите имя: " <b>ввод</b> <i>им</i>
Паскаль	<code>write('Введите фамилию: ');</code> <code>readln(fam);</code> <code>write('Укажите имя: ');</code> <code>readln(im);</code>
Python	<code>print("Введите фамилию")</code> <code>fam = input()</code> <code>print("Укажите имя: ")</code> <code>im = input()</code>  или <code>fam = input("Введите фамилию: ")</code> <code>im = input("Укажите имя: ")</code>

<sup>11</sup> О том, как вывести на экран сообщение в программе на используемом вами языке, узнайте самостоятельно.

Можно также вывести на экран общее сообщение:

Язык программирования	Оператор
Школьный алгоритмический язык	<b>Вывод</b> <b>нс</b> , "Введите фамилию, а затем имя " <b>Ввод</b> фам <b>Ввод</b> им
Паскаль	write('Введите фамилию, а затем имя '); readln(fam); readln(im);
Python	print("Введите фамилию, а затем имя: ") fam = input() im = input()

и (в некоторых языках программирования) указывать в операторе ввода два (и более) имени переменных:

Язык программирования	Оператор
Школьный алгоритмический язык	<b>Вывод</b> <b>нс</b> , "Введите фамилию, а затем имя " <b>Ввод</b> фам, им
Паскаль	write('Введите фамилию, а затем имя '); readln(fam, im);

*Примечание.* Не выполняя программу (на языке программирования, который вы изучаете), определите, как будут выводиться приведенные в ней сообщения.

Теперь об операторе присваивания. Этот оператор позволяет изменить (или задать впервые) значение переменной величины.

Например, когда присваивается значение величине числового типа, оператор имеет вид:

– в школьном алгоритмическом языке и языке Паскаль:

*<имя переменной> := <арифметическое выражение>*

– в языке Python:

*<имя переменной> = <арифметическое выражение>*

где *<арифметическое выражение>* – одно или несколько чисел, имен переменных величин<sup>12</sup> или имен функций<sup>13</sup>, соединенных знаками арифметических операций. Примеры:

Школьный алгоритмический язык, Паскаль	Python
c := 12	c = 12
m := n	m = n
сумма := a + b	sum = a + b
x := 2 * a - 3.6 * b/c	stepen3 = a ** 3
sred := (a + b) / 2	x = 2 * a - 3.6 * b/c
	sred = (a + b) / 2

Обратим внимание на знак деления («/»), а также на использование в последнем примере круглых скобок для изменения порядка действий (во всех языках программирования выражения записываются в строчку<sup>14</sup>, без «многоэтажных» дробей).

Распространенной ошибкой начинающих программистов является запись операторов присваивания в виде:

c := 2a

<sup>12</sup> Это также говорит о том, что к значению переменной величины мы обращаемся по ее имени (см. выше о выводе значения на экран).

<sup>13</sup> О том, что такое «функция», см. Дополнение «О функциях» в конце раздела.

<sup>14</sup> Такую запись называют «линейной».

или

$$c = 2a$$

(правильный вариант:  $c := 2 * a$  или  $c = 2 * a$ ; вариант оператора  $c := a2$  или  $c = a2$  возможен, если  $a2$  – имя переменной, значение которой задано).

Когда в операторе присваивания записаны знаки нескольких операций, то они (операции) выполняются в определенном порядке. При определении порядка действий учитывается приоритет (старшинство) операций. Они выполняются в следующем порядке:

- операции в скобках;
- возведение в степень, справа налево (!);
- умножение (\*) и деление (/), слева направо;
- сложение и вычитание, слева направо.

Таким образом, умножение и деление имеют одинаковый приоритет, более высокий, чем сложение и вычитание. Поэтому в последнем приведенном примере ( $sred := (a + b) / 2$ ) запись без скобок привела бы к тому, что сначала выполнилось бы деление значения  $b$  на  $2$ , а затем сложение этого частного и значения переменной  $a$ .

Над величинами целого типа, кроме операций сложения, вычитания, умножения, деления и возведения в степень, можно выполнять также еще две операции:

1) операция определения целой части частного («целочисленного частного») от деления одного целого числа на другое (её называют «целочисленное деление», или «деление нацело»);

2) операция определения остатка от целочисленного деления.

Это сделано потому, что задачи, связанные с использованием этих операций, очень часто встречаются в программировании.

Знаки этих двух операций показаны в таблице:

Язык программирования	Целочисленное деление	Определение остатка	Примеры
Паскаль	div	mod	$ost := a \text{ mod } 2$ $a := b \text{ div } (c + 1)$
Python	//	%	$ch = n // d$ $ost = 1000 \% m$

Они имеют такой же приоритет, как и «обычные» умножение и деление.

В школьном алгоритмическом языке для определения целочисленного частного от деления одной величины на другую и остатка от такого деления используются не знаки операций, а функции `div` и `mod`, соответственно:

```
частн := div(a, b)
ост := mod(a, 10)
```

Как работает оператор присваивания? Объясним на примере:

```
средн := (a + b) / 2
```

Транслятор (см. главу 1) «посмотрит», что «лежит» в ячейках  $a$  и  $b$ , сложит соответствующие значения и сумму разделит на  $2$ . Результат «положится» в ячейку `средн`:

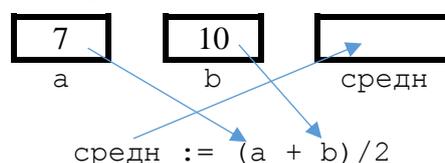


Рис. 2.2

Это означает, что в программе может быть запись:

```
x := x + 10
```

невозможная в математике. Ее смысл в том, что нужно к имеющемуся в данный момент выполнения программы значению переменной  $x$  прибавить 10, а результат присвоить той же переменной.

Мы рассмотрели особенности присваивания значения величинам числового типа.

Оператор присваивания значений величинам логического типа будет рассмотрена в разделе 3.3, а строкового типа – в главе 10.

Завершая обсуждение переменных величин, заметим, что в программах могут использоваться также так называемые «константы» – величины, которые в ходе выполнения программы не меняют заданное им единожды значение. Константы может создать (описать) программист, и, кроме того, во многих языках программирования имеются стандартные (предусмотренные в системе программирования) константы (см., например, раздел 2.3). С последними можно ознакомиться в справочной системе системы программирования.

### Контрольные вопросы

1. Какие величины в программе называют «переменными»?
2. Чем характеризуется каждая переменная?
3. Каковы правила присвоения имен переменным?
4. Почему желательно переменным давать «говорящие» имена?
5. Какие типы переменных вы знаете?
6. Что определяет тип переменной?
7. Какие значения может иметь переменная логического типа?
8. Как можно смоделировать хранение значения переменной в памяти компьютера?
9. Как обратиться к значению (использовать значение) переменной величины в программе?
10. С помощью какого оператора можно ввести в программу значение переменной в ходе ее выполнения?
11. В чем заключается особенность ввода в программу в ходе ее выполнения значений переменных?
12. Почему желательно выводить на экран подсказку перед вводом данных?
13. С помощью какого оператора можно изменить (или задать впервые) в программе значение переменной? Как он оформляется в общем случае? Как он оформляется применительно к переменным числового типа?
14. Возможна ли в изучаемом вами языке программирования операция возведения в степень? Если возможна, то какой знак используется для нее?
15. Какой знак используется в изучаемом вами языке программирования для операций целочисленного деления и определения остатка?
16. Что такое приоритет операций? Зачем он нужен? Перечислите арифметические операции в порядке уменьшения приоритета.
17. В каком порядке выполняются операции, если они имеют одинаковый приоритет?
18. Зачем в операторе присваивания используются скобки?
19. Что такое «константа»? Какие виды констант возможны в программировании?

### Задачи для самостоятельной работы

1. Получите линейную запись следующих выражений:

а)  $\frac{-1}{x^2}$ ;    г)  $\frac{(a+b)}{2}$ ;    ж)  $\frac{-b + \frac{1}{a}}{\frac{2}{c}}$ ;    й)  $2^{m^n}$ .

$$\begin{array}{lll} \text{б)} \frac{a}{bc}; & \text{д)} 5,45 \frac{(a+2b)}{2-b}; & \text{з)} \frac{1}{1+\frac{a+b}{2}}; \\ \text{в)} \frac{a}{b}c; & \text{е)} \frac{-b+\sqrt{b^2-4ac}}{2a}; & \text{и)} \frac{1}{1+\frac{1}{2+\frac{1}{2+\frac{3}{5}}}}; \end{array}$$

2. Запишите следующие выражения, представленные в линейной записи, в обычную форму:

а)  $a/b/c$ ;

б)  $a+b/c$ ;

в)  $(a+b)/c$ ;

г)  $a*b/(c+2)$

д)  $a/b*(c-3)/d$

е)  $(a/3+b/2)/(b+c/(2+b))$

Рассмотрим простейшие примеры задач на разработку программ. Начнем с задач с величинами целого типа<sup>15</sup>.

**Внимание!**

Как уже отмечалось, в школьном алгоритмическом языке, который мы будем использовать далее при описании методики решения задач, для определения результата целочисленного деления и определения остатка применяются не операции, а специальные функции `div` и `mod`. Но для единообразия при описании мы будем условно записывать их как знаки операций ( $10 \text{ div } 3, 10 \text{ mod } 3$  и т.п.).

**Задача 1.** Дано двузначное натуральное число. Найти сумму его цифр.

*Решение*

Как, имея, например, число 47, получить число 4? – Надо разделить 47 на 10, причем нацело (иначе получится 4,7). Итак, первая цифра заданного числа  $n$ :

$$\text{перв} := n \text{ div } 10$$

А вторая? Она может быть определена как остаток от деления  $n$  на 10:

$$\text{втор} := n \text{ mod } 10$$

Начинающие программисты после нахождения одной из цифр часто определяют другую, используя найденную:

$$\begin{array}{l} \text{перв} := n \text{ div } 10 \\ \text{втор} := n - 10 * \text{перв} \end{array}$$

или

$$\begin{array}{l} \text{втор} := n \text{ mod } 10 \\ \text{перв} := (n - \text{втор}) \text{ div } 10 \end{array}$$

При решении данной задачи такой вариант возможен, но в других случаях, когда количество цифр в заданном числе неизвестно, он не может быть применен (см. пункт 2.3.2).

---

<sup>15</sup> Более сложные задачи с целыми числами рассмотрены в разделе П1.4 приложения 1.

**Задача 2.** Дано трехзначное натуральное число. Найти сумму его цифр.

*Решение*

Первая цифра заданного числа  $n$  (количество сотен в нем):

перв :=  $n \operatorname{div} 100$

последняя (количество единиц):

трет :=  $n \operatorname{mod} 10$

А как определить вторую цифру (количество десятков). Одним действием это сделать нельзя. Потребуется два действия.

Можно сначала рассчитать двузначное число, образованное второй и третьей цифрами числа  $n$  (например, при  $n = 318$  получить 18):

втор\_трет :=  $n \operatorname{mod} 100$

а затем в нем выделить первую цифру (1):

втор :=  $\text{втор\_трет} \operatorname{div} 10$

Можно также определить двузначное число, образованное первой и второй цифрой числа заданного числа (например, при  $n = 318$  получить 31):

перв\_втор :=  $n \operatorname{div} 10$

а в нем выделить последнюю, вторую, цифру:

втор :=  $\text{перв\_втор} \operatorname{mod} 10$

Самостоятельно убедитесь в том, что во всех случаях результат будет правильным даже в случае, когда «средняя» цифра числа  $n$  равна нулю.

Мы же обратим внимание на то, что последняя цифра как двузначного, так и трехзначного числа равна остатку от деления заданного числа на 10.

### **Задачи для самостоятельной работы**

1. Даны два целых числа. Определить:

- их сумму;
- разность между первым и вторым числом;
- их произведение.

2. Составить программу вычисления значения функции:

- $y = 17x^2 - 6x + 13$  при любом значении  $x$ ;
- $z = 2x^3 - 4xy + 3x^2 - 7y + 2$  при любых значениях  $x$  и  $y$ ;

3. С некоторого момента прошло 234 дня. Сколько полных недель прошло за этот период?

4.  $n$  школьников делят  $k$  яблок поровну так, чтобы каждому достались только целые яблоки, остальные яблоки остаются в корзинке. Определить, сколько яблок достанется каждому школьнику и сколько яблок останется в корзинке.

5. Дан прямоугольник с размерами  $543 \times 130$  мм. Сколько квадратов со стороной 130 мм можно отрезать от него?

6. Из трехзначного числа  $x$  вычли его последнюю цифру. Когда результат разделили на 10, а к частному слева приписали последнюю цифру числа  $x$ , то получилось число 237. Найти число  $x$ .

7. С начала суток прошло  $n$  секунд. Определить:

- сколько полных часов прошло с начала суток;
- сколько полных минут прошло с начала очередного часа;

в) сколько полных секунд прошло с начала очередной минуты.

8. Напишите программу, в которой вычисляется сумма, разность и произведение двух целых чисел, введенных с клавиатуры. Например, при вводе чисел 2 и 7 должен быть получен ответ вида:

```
2+7=9
2-7=-5
2*7=14
```

Рассмотрим теперь задачу, так сказать, «с особенностями»: «Даны два целых числа. Определить частное от деления первого на второе».

Если программу на языке Паскаль оформить аналогично тому, как решался один из вариантов задачи 1:

```
Var a, b, chastn: integer;
BEGIN
  write('Задайте первое число ');
  readln(a);
  write('Задайте второе число ');
  readln(b);
  chastn := a/b;
  writeln;
  writeln('Частное от деления первого числа на второе равно ', chastn)
END.
```

– то при выполнении появится сообщение об ошибке (см. рис. 2.3).

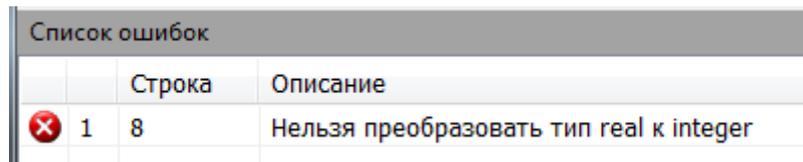


Рис. 2.3

Оно связано с оператором:

```
chastn := a/b;
```

В нем целое число  $a$  делится на целое же  $b$  и результат присваивается переменной `chastn` целого типа (как описано в начале программы). В то же время операция деления (знак операции «/») дает вещественный результат даже в случаях, когда она проводится над целыми числами, в том числе в случаях  $25/5$ ,  $48/12$  и т. п. А переменные вещественного типа требуют для своего хранения в памяти компьютера больше места, чем переменные целого типа (см. выше). Поэтому переменная `chastn` должна быть описана как имеющая вещественный тип. Это замечание должно быть учтено и в программе на школьном алгоритмическом языке.

Другая особенность заключается в виде результата. Например, если частное, описанное как величина вещественного типа, не определяется точно, то оно выводится с большим количеством цифр после десятичной точки:

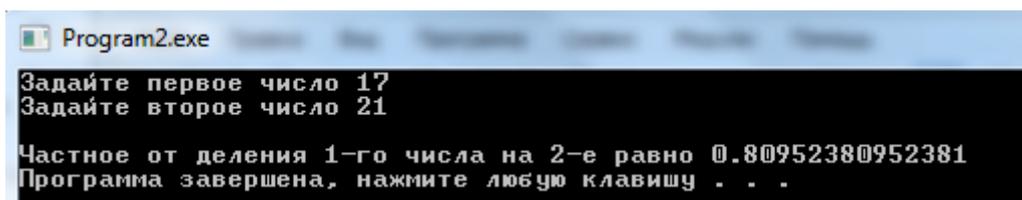


Рис. 2.4

Количество цифр в дробной части результата можно ограничить. Как это сделать в используемом вами языке программирования, установите самостоятельно.

### Задачи для самостоятельной работы

1. Число  $\pi$  примерно равно 3,1415926. Вывести на экран это число с тремя цифрами в дробной части. Текст "3.142" не использовать.
2. Число  $e$  (так называемое «основание натурального логарифма») приблизительно равно 2,71828. Вывести на экран это число с точностью до десятых. Текст "2.7" не использовать.
3. Известны объем и масса тела. Определить плотность материала этого тела в  $\text{г/см}^3$ .
4. Составить программу решения линейного уравнения  $ax + b = 0$  ( $a \neq 0$ ).

### Дополнение «О функциях»

Возникает вопрос: «А можно ли в программах извлекать квадратный корень?» Можно. Для этого следует использовать соответствующую функцию. В программировании *функция* – «самостоятельная» программа, имеющая имя и решающая какую-то частную, вспомогательную задачу<sup>16</sup>. Функции, как правило, бывают двух видов:

- 1) стандартные («встроенные» в систему программирования и доступные без всяких условий);
- 2) нестандартные – созданные пользователем. Подробно о таких функциях рассказано в главе 12.

В школьном алгоритмическом языке и в языке Паскаль функция, определяющая квадратный корень, является стандартной. Её имя – `sqrt`.

Любая функция вызывается для выполнения по её имени. Например, для функции `sqrt()` это делается так:

```
koren := sqrt(...)
```

где вместо ... указывается арифметическое выражение (см. выше):

```
kor2 = sqrt((a + b)/2)
writeln(sqrt(n))
if sqrt(x) > 2 then ...
```

Функция с таким именем имеется и в языке программирования Python. Но она относится к функциям особого вида – предусмотренным в системе программирования, но для вызова (использования) которых необходимо подключить к разрабатываемой программе модуль<sup>17</sup>, в который входят такие функции<sup>18</sup>. Для использования в программе функции `sqrt` следует в начале программы записать:

```
import math
```

Пример:

```
koren = math.sqrt(...) # Используется так называемая "точечная запись"
```

где вместо ... указывается арифметическое выражение.

Можно подключить к программе только нужные функции:

```
from math import sqrt, sin, cos
```

---

<sup>16</sup> Во всех языках программирования имеются также вспомогательные программы другого вида – так называемые «процедуры» (в языке Python оба вида вспомогательных программ называют «функциями»). См. главу 12).

<sup>17</sup> Модуль – элемент системы программирования, включающий в себя группу некоторых функций.

<sup>18</sup> Автор называет такие функций «полустандартными».

В этом случае все функции модуля `math`, кроме перечисленных (`sqrt`, `sin`, `cos`), в программе будут недоступны. Перечисленные при импорте конкретные функции можно вызывать без использования точечной записи:

```
r = sqrt((a + b)/2)
print(sin(n))
if sqrt(x) > 2:
```

Полустандартные функции предусмотрены и в языке программирования Паскаль. Например, к ним относится функция `keypressed` (см. разделы 2.3 и главу 5). Для их использования в начале программы следует записать служебное слово `Uses` и указать имя модуля/модулей с используемыми функциями (функция `keypressed` входит в модуль `CRT`).

Обратим внимание на то, что во всех языках функция для расчета квадратного корня возвращает вещественный результат даже в случаях, когда ее аргумент равен 1, 4, 49 и т. п.

### Задачи для самостоятельной работы

1. Даны катеты прямоугольного треугольника. Найти его гипотенузу.
2. Даны координаты на плоскости двух точек. Найти расстояние между этими точками.
3. Составить программу вычисления значения функции  $y = \frac{a^2 + 10}{\sqrt{a^2 + 1}}$  при любом значении

а.

Найдите ошибку в записи операторов присваивания для расчетов корней в программе решения квадратного уравнения с коэффициентами `a`, `b`, `c` ( $a \neq 0$ ):

```
x1 := (-b + sqrt(diskr))/2 * a
x2 := (-b - sqrt(diskr))/2 * a
```

где `diskr` – имя переменной величины, соответствующей дискриминанту квадратного уравнения, `sqrt` – функция, возвращающая квадратный корень из своего аргумента.

### Внимание!

В программах на языке Паскаль неправильно определяется остаток от деления отрицательных чисел. Так, например,  $-1 \bmod 24$  вернет результат, равный  $-1$ , в то время как по правилам математики он равен 23.

Продолжение — в следующем выпуске.

Книжная полка

## Новые книги издательства ДМК-Пресс



## Ответы, решения, разъяснения

к заданиям, опубликованным в журнале «Мир информатики» № 41 (январь 2020 г.)

### Числовые ребусы в пятеричной системе счисления. Часть 4

Ответы

Ребус № 10	$\begin{array}{r} + 3 3 \\ 3 3 \\ \hline 1 2 1 \end{array}$	Ребус № 11	$\begin{array}{r} + 4 4 \\ 4 4 \\ \hline 1 4 3 \end{array}$	Ребус № 12	$\begin{array}{r} + 1 1 \\ 1 2 \\ \hline 2 3 \end{array}$
------------	---	------------	---	------------	---

### Задание «Шесть вопросов» (рубрика «Поиск информации»)

Ответы

1. Хвойное дерево, плакучую разновидность которого вывели в Никитском ботаническом саду, — кедр.
2. Домашнее прозвище Владимира Маяковского — «Щенок».
3. Герой романа американского писателя Джона Апдайка «Кролик разбогател» торгует автомобилями марки «Тойота».
4. Русский вариант испанского имени Диего — Яков.
5. Советский разведчик Рудольф Абель добывал секреты американской атомной бомбы под псевдонимом «Марк».
6. Сухопутный гигант, близкими родственниками которого приходятся ламантин и дюгонь, — слон.

### Задача «В одном купе»

Напомним условие: «Как-то раз судьба свела в одном купе поезда известного историка, поэта, прозаика и драматурга. Это были Антонов, Борисенко, Симонян и Дмитриев. Когда поезд тронулся, они углубились в чтение. Оказалось, что каждый из них взял с собой книгу, написанную одним из пассажиров купе. Антонов и Борисенко, дочитав каждый свою книгу, условились на завтра обменяться ими. На следующий день поэт читал пьесу. Прозаик, очень молодой человек, выпустивший свою первую книгу, говорил, что он в жизни не читал и не читает ничего по истории. Историк Борисенко читал произведение Дмитриева. И, конечно, никто из пассажиров не читал книгу, написанную им самим.

Что читал каждый из пассажиров? Каковы профессии каждого из спутников?»

Задача решается табличным методом:

	Профессии				Кто какую книгу читал			
	Историк	Поэт	Прозаик	Драматург	Исторические книги	Стихи	Проза	Пьеса
<b>Антонов</b>	—	+	—	—	—	—	+	+
<b>Борисенко</b>	+	—	—	—	—	—	+	+
<b>Симонян</b>	—	—	—	+	+	—	—	—
<b>Дмитриев</b>	—	—	+	—	—	+	—	—

Ответ: поэт Антонов и историк Борисенко читали пьесу и прозу, драматург Симонян читал историческую книгу, прозаик Дмитриев читал стихи.

### Пять ребусов

Ответы

Ребус № 1: БИТ. Ребус № 2: СИМВОЛ. Ребус № 3: АЛФАВИТ. Ребус № 4: МОЩНОСТЬ АЛФАВИТА. Ребус № 5: ИНФОРМАЦИОННЫЙ ОБЪЕМ.

### Задача «День рождения»

Напомним условие: «Алина и Алена были очень правдивыми девушками и говорили неправду в виде исключения только в день своего рождения. 26 октября кто-то спросил каждую из них: “Когда твой день рождения?” Алина сказала: “Он был вчера”, Алена ответила: “Он будет завтра”. На следующий день их спросили то же самое, и, как ни странно, девушки ответили точно так же. Когда родилась каждая из них?».

*Решение*

Допустим, что день рождения Алины	Тогда её высказывание	
	сделанное 26 октября	сделанное 27 октября
26 октября	— ложное, то есть её день рождения не 25 октября	— истинное, то есть её день рождения 26 октября
27 октября	— истинное, то есть её день рождения 25 октября. Но это противоречит допущению	

Итак, день рождения Алины — 26 октября.

Аналогичные рассуждения показывают, что Алёна родилась 27 октября.

### Задача “Сколько стоило мороженое?”

Напомним, что необходимо было определить “старинную” стоимость мороженого, если когда-то Павлику не хватало на мороженое 19-ти копеек, а Вове — одной, а когда они сложили свои деньги вместе, то на покупку все равно не хватило.

*Решение*

Обозначим сумму денег, имевшихся у Павлика, —  $P$ , у Вовы —  $B$ , а стоимость мороженого —  $C$ .

Тогда можем записать, что:

$$1) C = P + 19; \quad (1)$$

$$2) C = B + 1. \quad (2)$$

Так как  $P + B$  меньше  $C$ , то, подставив вместо  $C$  сумму из (1), получим:

$$P + B < P + 19,$$

откуда  $B < 19$ . Но это нам ничего не дает ☹.

А если использовать  $C$  сумму из (2), то:

$$P + B < B + 1, \text{ или } P < 1, \text{ то есть } P = 0.$$

Иными словами, у Павлика денег совсем не было. А поскольку ему не хватало на мороженое 19 копеек, то это значит, что стоимость этого “деликатеса” и была равна 19 копеек.

Можно также решить задачу методом рассуждений.

Прежде всего, ясно, что мороженое не могло стоить менее 19 копеек (у Павлика не могла быть отрицательная сумма денег).

Допустим, что стоимость мороженого — 20 копеек. В этом случае у Павлика была 1 копейка, а у Вовы — 19. Но тогда общая сумма была бы достаточной для покупки. Аналогично и при стоимости в 21 копейку и более. Значит, истинная стоимость — 19 копеек. (Вот бы сейчас так! ☺)

## Кроссворд

### Ответы

По горизонтали: 1. ТРИ 4. ОР 5. КОНТРОЛЬ 7. ТИП 10. ДНО 11. ПРОЦЕССОР  
13. ДИАЛОГ 14. ОПРОС 15. СТАТУС 16. КЛОН 17. ТЕЛО 19. РАСТР 22. СОСТАВ  
24. ФОРМАТИРОВАНИЕ 25. СТИЛЬ 26. АРЕНА

По вертикали: 1. ТРАНЗИСТОР 2. ИКС 3. ПРОЦЕНТ 6. ЛИСТИНГ 8. ПРОТОКОЛ  
9. БОДО 12. РЕГИСТР 16. КЛАВИША 18. ОСНОВА 20. АДРЕС 21. ДАННЫЕ 23. ТИШЬ.

## Задача «Цепочки цифр»

Напомним условие: «Строки (цепочки цифр) создаются по следующему правилу.

Первая строка состоит из одной цифры 0. Каждая из последующих цепочек создается такими действиями: в очередную строку сначала записывается цифра, на 1 большая последней цифры в предыдущей строке, к ней слева дважды подряд приписывается предыдущая строка.

- (1) 0
- (2) 001
- (3) 0010012
- (4) 001001200100123.

Какие из приведенных ниже строчек соответствуют описанным правилам?

- 1) 00100120010012300100120010012340010012001001230010120010012345
- (2) 001001200100123001001200100123400100120010012300100120010123456
- (3) 001001200100123001001200100123400100120010012300100120010012345
- (4) 000100120010012300100120010012340010012001001230010012001012345
- (5) 001001200100123400100120010012300100120010012300100120010012345
- (6) 001001200100123400100120010012300100120010012300100120010001345

### Решение

Так как первые четыре формируемой строки известны, то искомая строка должна начинаться так:

001001200100123

Этому требованию соответствуют только 3-я и 5-я строки. Их проверка на соответствие правилу в условии показывает, что оно соблюдается только в 3-й строке. Можно для проверки также поступить так:

- отбрасывать последнюю цифру;
- делить оставшуюся строку пополам и сравнивать её половины:  
если они одинаковые, то поступать аналогично.

Так поступать до тех пор, когда либо обнаружится расхождение, либо дойдем до строки “0” (после “001”). Например, для 5-й строки:

001001200100123400100120010012300100120010012300100120010012345  
0010012001001234001001200100123  
0010012001001230010012001001234 (расхождение половин).

Ответ — строка 3.

## Задача “Чернявый, Кудрявый и Лысый”

Напомним, что необходимо было определить, кто из трех участников преступной группировки — Чернявый, Кудрявый и Лысый — совершил преступление, если из двух заявлений каждого:

Чернявый: “Я не делал этого. Это сделал Лысый”;

Кудрявый: “Лысый не виновен. Это сделал Чернявый”;

Лысый: “Я этого не делал. Кудрявый этого тоже не делал”

— суд установил, что один из них дважды солгал, другой дважды сказал правду, а третий — один раз солгал, а другой раз сказал правду.

*Решение*

Рассмотрим возможные варианты для двух правдивых заявлений.

1. Предположим, что их сделал Лысый. Из его слов следует (что ни он, ни Кудрявый не виноват) следует, что виноват Чернявый. Но тогда оба утверждения Кудрявого также верны, а этого быть не может. Следовательно, наше предположение неверно.

2. Допустим, что два раза сказал правду Кудрявый. В этом случае оба утверждения Лысого тоже верны, что также противоречит условию.

3. Из предположения, что правдивыми являются заявления Чернявого (то есть преступление совершил Лысый), следует, что Кудрявый дважды солгал, а Лысый в первый раз солгал, а во второй раз сказал правду. То есть такой вариант допустим.

*Ответ:* преступление совершил Лысый.

### Головоломка «Как получить 3 квадрата?»

Решение показано на рисунке:



### Ответы и решения представили:

— Абдувахидова Алина, Абдувахидова Софья, Баранов Матвей, Добрыднева Ксения, Изотов Павел, Камышанская Анастасия, Лебедев Сергей, Лефтерев Влад, Лопатин Тимофей и Хозин Марат, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Боровиков Иван, Костюк Максим, Кравец Сергей и Фунтиков Роман, г. Пенза, школа № 73, учитель **Диков А.В.**;

— Великородных Светлана и Королёва Марина, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Гуртуева Дарья, Медведева Мария Сангинов Набиджон и Устинов Виталий, средняя школа села Ириновка, Новобураский р-н Саратовской обл., учитель **Брунов А.С.**;

— Серякова Елена и Фомин Никита, Республика Башкортостан, г. Стерлитамак, гимназия № 4, учитель **Орлова Е.В.**;

— Торопов Александр, Филимонова Надежда и Юмашев Сергей, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**

Читатели, приславшие наибольшее число правильных ответов, будут награждены дипломами. Поздравляем всех!

### Крепкий орешек

Как всегда, в этой рубрике редакция проводит разбор задач, решение которых вызвало трудности.

### Задача «Кто выше?»

Напомним условие: «Учащиеся школы построены в виде каре (прямоугольника). В каждой колонне выбрали самого высокого школьника, а из них — самого низкого. Им оказался Петя Иванов. Затем в каждой шеренге выбрали самого низкого школьника, а из них — самого высокого. Им оказался Ваня Петров. Кто выше — Ваня или Петя?».

Приведем начало решения<sup>19</sup>.

Упомянутое в условии каре (прямоугольник) будем моделировать в виде таблицы, строки которой соответствуют шеренгам, а столбцы – колоннам (каждый школьник моделируется клеткой этой таблицы). Пусть Пете Иванов стоял в некоторой колонне, выделенной синим, а Ваня Петров находился в некоторой шеренге, выделенной зеленым (рис. 1).


ПИ

Рис. 1

Рассмотрим место на пересечении указанных колонны и шеренги.

Допустим, на этом месте был Ваня Петров (рис. 2). В этом случае он — ниже ростом, чем Петя Иванов, так как Петя — самый высокий в синей колонне.

		ВП	

ПИ

Рис. 2

Рассмотрите еще два случая:

- 1) на этом месте стоял Петя Иванов;
- 2) на этом месте стоял кто-то “третий” (пусть это был Дима Сидоров) и установите окончательный ответ на заданный в условии вопрос.

Ответ присылайте в редакцию.

**Внимание! Конкурс!**

## Итоги конкурса № 24

Напомним, что в качестве задания этого конкурса было предложено разработать программы для шифрования текста тремя методами шифрования.

Программы представили и победителями конкурса стали:

- Муллахметов Булат, Республика Татарстан, Мамадышский р-н, совхоз “Мамадышский”, политехнический колледж, преподаватель **Порываева Н.С.**;
- Соколинский Максим, г. Пенза, школа № 73, учитель **Диков А.В.**

## Итоги конкурса № 26

Напомним, что в качестве задания этого конкурса было предложено решить числовой ребус  $КУБ = Б^Y + K$ . В нем одинаковыми буквами зашифрованы одинаковые цифры, разными буквами — разные цифры. Вспомогательные расчеты следовало провести с помощью электронной таблицы Microsoft Excel или подобной программы.

<sup>19</sup> Ответ (к сожалению, без обоснования) прислали Серякова Елена и Фомин Никита, Республика Башкортостан, г. Стерлитамак, гимназия № 4, учитель **Орлова Е.В.**

### Решение

Прежде всего, можно установить, что показатель степени ( $У + К$ ) не может быть меньше трех, поскольку даже  $9^2$  — не трехзначное число, и не больше 9, так как  $2^{10} > 999$ .

После этого определим значения показателя степени, дающие трехзначное значение  $Б^{У + К}$ . Для этого может быть использована электронная таблица Microsoft Excel. В приведенном ниже фрагменте листа в диапазоне В3:G10 получены значения  $Б^{У + К}$ .

	A	B	C	D	E	F	G	H
1	<b>Показатель степени У + К</b>							
2	<b>Б</b>	3	4	5	6	7	8	9
3	2	8	16	32	64	128	256	512
4	3	27	81	243	729	2187	6561	
5	4	64	256	1024	4096	16384	65536	
6	5	125	625	3125	15625	78125	390625	
7	6	216	1296	7776	46656	279936	1679616	
8	7	343	2401	16807	117649	823543	5764801	
9	8	512	4096	32768	262144	2097152	16777216	
10	9	729	6561	59049	531441	4782969	43046721	

Формула в ячейке В3 может быть оформлена так, что ее можно распространить (скопировать на остальные ячейки диапазона В3:G10). Вид такой формулы установите самостоятельно.

Допустимые варианты для анализа приведены в таблице:

	<b>Показатель степени У + К</b>						
<b>Б</b>	3	4	5	6	7	8	9
2					128	256	512
3			243	729			
4		256					
5	125	625					
6	216						
7	343						
8	512						
9	729						

Из нее видно, что значение цифры **Б** совпадает с последней цифрой результата в шести случаях:

- 1)  $512 = 2^9$ ;
- 2)  $243 = 3^5$ ;
- 3)  $125 = 5^3$ ;
- 4)  $625 = 5^4$ ;
- 5)  $216 = 6^3$ ;
- 6)  $729 = 9^3$ ,

из которых с точки зрения значений **У** и **К** подходят только третий и пятый варианты.

Итак, ребус имеет два решения:

- 1) **Б** = 5, **У** = 2, **К** = 1 ( $125 = 5^{2+1}$ );
- 2) **Б** = 6, **У** = 1, **К** = 2 ( $216 = 6^{1+2}$ ).

Правильные и обоснованные ответы прислали и победителями конкурса стали:

— Соколинский Максим, г. Пенза, школа № 73, учитель **Диков А.В.**

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

## Задание 19 демонстрационного варианта ЕГЭ по информатике и ИКТ 2020 года

### Условие

В программе используется одномерный целочисленный массив  $A$  с индексами от 0 до 11. Значения элементов равны 20, 19, 17, 41, 23, 12, 24, 16, 4, 13, 6, 15 соответственно, то есть  $A[0] = 20$ ,  $A[1] = 19$  и т. д. Определите значение переменной  $s$  после выполнения следующего фрагмента этой программы (записанного ниже на четырех языках программирования).

Алгоритмический язык	Паскаль
<pre> s := 0 n := 0 нц для i от 0 до 11   если A[i] &lt;= A[n]     то       s := s + i       t := A[i]       A[i] := A[n]       A[n] := t   все кц </pre>	<pre> s := 0; n := 0; for i := 0 to 11 do   if A[i] &lt;= A[n] then     begin       s := s + i;       t := A[i];       A[i] := A[n];       A[n] := t     end; end; </pre>
Python	Бейсик
<pre> s = 0 n = 0 for i in range(12):   if A[i] &lt;= A[n]:     s = s + i     t = A[i]     A[i] = A[n]     A[n] = t </pre>	<pre> s = 0 n = 0 for i = 0 to 11   if A[i] &lt;= A[n] then     s = s + i     t = A[i]     A[i] = A[n]     A[n] = t   end if next i </pre>

### Решение

Анализ показывает, что в «теле» условного оператора:

- 1) переменная  $s$  увеличивается на  $i$ ; эта переменная суммирует индексы  $i$ , при которых условие истинно;
- 2) происходит обмен значениями элементов  $A[i]$  и  $A[n]$ , то есть текущий элемент, меньший начального, становится начальным.

Обратим также внимание на то, что в цикле происходит сравнение  $i$ -го элемента не с последним, как может показаться (из-за имени  $A[n]$ ), а с начальным элементом массива (так как  $n = 0$ ).

Это означает, что подсчеты  $s$  и изменения массива будут происходить, когда текущий элемент меньше начального. Для заданного массива это будет иметь место для значений 20, 19, 17, 12 и 4 (при рассмотрении значения 6 начальный элемент будет равен 4). Сумма индексов пяти перечисленных элементов равна:  $0 + 1 + 2 + 5 + 8 = 16$ .

*Ответ:* 16.

## Уважаемые коллеги!

Для поощрения самых активных участников наших конкурсов редакция может в мае-июне направить вам по запросу макет грамоты. Оформление грамот будет проводиться в школах. Запрос, пожалуйста, направляйте в редакцию по адресу [mirinfo@infojournal.ru](mailto:mirinfo@infojournal.ru).

## Наше пожелание выпускникам:

