

«Мир информатики»

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 73, декабрь 2022 г.

Школа программирования

Об использовании функций

Как, конечно, известно читателю, во всех современных языках программированию имеются различные стандартные функции. Напомним, что в программировании *функция* – «самостоятельная» программа, имеющая имя и решающая какую-то частную, вспомогательную задачу. Например, стандартная функция *sqrt* языков Паскаль, Python и школьного алгоритмического языка решает задачу вычисления квадратного корня из значения, указанного в качестве её (функции) аргумента. Кроме стандартных, в программах могут использоваться и нестандартные, созданные пользователем, функции. Они также предназначены для решения некоторых задач и возвращают значение в вызывающую их программу.

Пусть, например, в программе надо рассчитать значения x :

$$x = \frac{2 + \sqrt{2}}{5 + \sqrt{5}} + \frac{5 + \sqrt{5}}{13 + \sqrt{13}} + \frac{11 + \sqrt{11}}{8 + \sqrt{8}} + \frac{14 + \sqrt{14}}{7 + \sqrt{7}}$$

Видно, что в выражении имеются похожие фрагменты – дроби. Желательно для расчета дробей оформить функцию, которая вычисляла бы значения отдельных дробей вида $\frac{a + \sqrt{a}}{b + \sqrt{b}}$ при любых значениях a и b , а затем использовать ее для расчетов.

Правила оформления функций в разных языках программирования различаются. В школьном алгоритмическом языке общий вид функций такой¹:

```
алг <тип результата> <имя функции> (<список формальных параметров>)  
нач <описания переменных величин используемых в функции>  
    <тело функции>  
кон
```

причем последним оператором тела функции должен быть оператор присваивания, в левой части которого должно быть указано служебное слово **знач**.

В нашем случае функция для расчета значения дробей указанного чуть выше вида оформляется так:

```
алг вещ Др(арг цел a, b)  
нач  
    знач := (a + sqrt(a)) / (b + sqrt(b))  
кон
```

¹ С правилами оформления функций в языке программирования, который вы изучаете, ознакомьтесь самостоятельно.

Как и стандартные, функция, созданная программистом, используется в правой части оператора присваивания². Для этого надо указать ее имя, а в скобках – значения фактических параметров:

```
x := Др(2,5) + Др(5,13) + Др(13, 8) + Др(14,7)
```

Видно, что благодаря использованию функции, оператор для расчета значения x гораздо компактнее такого варианта:

```
x := (2 + sqrt(2))/(5 + sqrt(5)) + (5 + sqrt(5))/(13 + sqrt(13)) +  
(13 + sqrt(13))/(8 + sqrt(8)) + (14 + sqrt(14))/(7 + sqrt(7))
```

Кроме того, в последнем варианте выше вероятность появления ошибки при записи оператора.

Еще одно преимущество использования собственных функций – улучшение читаемости (понятности) программы. Например, задачу «Даны 20 целых чисел. Определить, сколько из них имеют ровно 4 делителя» можно решить так:

```
алг Задача о числах с четырьмя делителями  
нач цел n, возм_дел, кол_дел, кол_4дел, номер  
    кол_4дел := 0 | Начальное значение искомой величины  
    нц для номер от 1 до 20 | Вводим и исследуем 20 чисел  
        вывод нс, "Введите ", номер, "-е число"  
        ввод n  
        кол_дел := 0 | Начальное значение количества делителей числа n  
        нц для возм_дел от 1 до n | Рассматриваем все  
            | возможные делители  
            если n mod возм_дел = 0  
                то | Число возм_дел – делитель числа n  
                    кол_дел := кол_дел + 1  
            все  
        кц  
        если кол_дел = 4  
            то  
                кол_4дел := кол_4дел + 1  
        все  
    кц  
    вывод кол_4дел  
кон
```

Если же разработать функцию, определяющую количество делителей числа n :

```
алг цел Кол_дел(арг цел n)  
нач цел возм_дел, кол_дел  
    кол_дел := 0  
    нц для возм_дел от 1 до n  
        если n mod возм_дел = 0  
            то  
                кол_дел := кол_дел + 1  
        все  
    кц  
    знач := кол_дел  
кон
```

² Конечно, нестандартная функция может быть записана в условиях (логических выражениях), в операторах вывода на экран и т. д.

и использовать её в программе решения «главной» задачи:

```
алг Задача о числах с четырьмя делителями
нач цел n, кол_4дел, номер
  кол_4дел := 0
  нц для номер от 1 до 20
    ввод n
    если Кол_дел(n) = 4
      то
        кол_4дел := кол_4дел + 1
      все
    кц
  вывод кол_4дел
кон
```

то такая программа, безусловно, будет понятнее, чем приведенная первой.

Результат, возвращаемый функцией, может быть любого типа, в том числе логического³. Например, функция, определяющая, является ли натуральное число a делителем натурального числа b , имеет вид:

```
алг лог Делитель(арг цел a, b)
нач
  | Значение функции:
  знач := b mod a = 0 | Это оператор логического присваивания
кон
```

Такая функция может быть использована в другой функции – в возвращающей количество делителей числа (см. чуть выше):

```
алг цел Кол_дел(арг цел n)
нач цел возм_дел, кол_дел
  кол_дел := 0
  нц для возм_дел от 1 до n
    если Делитель(n, возм_дел)
      то
        кол_дел := кол_дел + 1
      все
    кц
  знач := кол_дел
кон
```

Задания для самостоятельной работы

1. Рассчитать значение x , определив и использовав функцию:

$$x = \frac{\sqrt{6}+6}{2} + \frac{\sqrt{13}+13}{2} + \frac{\sqrt{21}+21}{2}$$

2. Найти периметр треугольника, заданного координатами своих вершин. (Определить функцию для расчета длины отрезка по координатам его вершин.)

3. Даны 20 чисел. Определить, сколько из них являются простыми. Использовать функцию логического типа, распознающую простые числа.

³ Если, конечно, такой тип предусмотрен в языке программирования.

Две типовые задачи программирования

Рассмотренные далее в статье задачи часто встречается в программировании. Понадобятся они и при разработке программ, связанных с головоломкой «Ханойская башня» (см. далее). Опишем методику их решения. Как принято в нашем журнале, мы используем школьный алгоритмический язык (система программирования КуМир). Русский синтаксис этого языка делает приводимые методики и программы максимально понятными, и вы сможете разработать аналогичные программы на языке программирования, который изучаете.

1. Задача 1 «Заполнение массивов неповторяющимися случайными числами»

Пусть нужно заполнить массив `мас` из 100 элементов неповторяющимися целыми числами. Логика действия такая. Нужно многократно получать случайное целое число `сл_число`. Если его нет в массиве `мас`, то будет получено новое число (его номер – номер), которой можно записать в массив с индексом `номер`.

Для проверки наличия в массиве некоторого числа можно использовать величину `есть` логического типа (или принимающую значения 0 или 1). Вся программа:

```
алг Заполнение_массива
нач цел таб мас[1 : 100], цел i
  цел число, номер, лог есть
  номер := 0
  нц пока номер < 100
    сл_число := irand(1, 1000)
    | Проверяем
    есть := нет
    нц для i от 1 до номер
      если мас[i] = сл_число
        то
          есть := да
          выход
        все
      кц
    если не есть
      то | Получено очередное число
        номер := номер + 1 | Его номер
        | Записываем в массив
        мас[номер] := сл_число
      все
    кц
  | Вывод полученного массива
кон
```

Примечание. `irand` – функция школьного алгоритмического языка, возвращающая случайное целое число из заданного интервала значений.

Задание для самостоятельной работы

1. Разработайте аналогичную программу на языке программирования, который изучаете. В программе на языке Python для добавления нового элемента в список (массив) можно использовать метод `append()`, а проверку наличия в списке (массиве) заданного числа можно использовать оператор принадлежности `in`.

Разработаем также программу, в которой неповторяющимися случайными числами заполняются два массива.

Здесь, кроме массивов `мас1` и `мас2` по 50 элементов в каждом, используем вспомогательный массив из 100 элементов.

Сначала заполняем массив `всп_мас` так, как описано выше. Затем полученные 100 чисел распределяем в два массива. Для этого 100 раз определяем случайное число `номер_массива`, равное 1 или 2, и, в зависимости от его значения, записываем i -й элемент массива в той или иной «целевой» массив (ведя учёт количество уже записанных в них чисел). Вторая часть все программы решения задачи выглядит так:

```
кол1 := 0      | Начальные значения
кол2 := 0      | количества чисел в массивах
нц для i от 1 до 100
    номер_массива := irand(1, 2)
    если номер_массива = 1
        то | Записываем в массив мас1
            кол1 := кол1 + 1 | Номер очередного числа в массиве мас1
            мас1[кол1] := всп_мас[i]
        иначе | Записываем в массив мас2
            кол2 := кол2 + 1
            мас2[кол2] := всп_мас[i]
    все
кц
```

Примечание. Так как количество чисел, которые будут записаны в массивы `мас1` и `мас2`, неизвестно, то они должны быть описаны с «запасом».

Задания для самостоятельной работы

2. Разработайте аналогичную программу на языке программирования, который изучаете.

3. Разработайте также вариант программы, в котором вспомогательный массив не используется, а:

- сначала заполняется массив `мас1` из 50 элементов;
- после чего заполняется массив `мас2` из 50 элементов (числа в обоих массивах не должны повторяться).

2. Задача 2 «Проверка массива на упорядоченность»

Пусть дан массив `мас` из 10 элементов. Следуют определить, расположены ли все его элементы в порядке неубывания⁴, то все элементы, кроме последнего, не меньше расположенных справа от них.

Идея решения задачи достаточно очевидна – следует проверить все пары соседних элементов и, если хотя бы в одной паре «левый» элемент меньше «правого», то такой массив не является упорядоченным (отсортированным) по неубыванию. Для рассмотрения в программе всех пар элементов используем оператор цикла с параметром и команду выхода из цикла `выход`.

⁴ Возможны также проверки массивов на упорядоченность по убыванию, невозрастанию и возрастанию.

Фрагмент программы, в котором происходит проверка заполненного массива мас на упорядоченность по неубыванию и вывод ответа, имеет вид:

```
упор := да | Величина логического типа
нц для i от 1 до 9
  если стержень1[i] <= стержень1[i + 1]
    то
      упор := нет
      выход | Следующие пары не проверяем
  все
кц
если упор
  то
    вывод нс, "Массив упорядочен по неубыванию"
  иначе
    вывод нс, "Массив не упорядочен по неубыванию"
все
```

При многократной проверке массива на упорядоченность целесообразно разработать функцию логического типа, возвращающую значение **да** (истина) или **нет** (ложь). На школьном алгоритмическом языке такая функция имеет вид:

```
алг лог Отсорт
нач цел i, лог упор
  упор := да | Величина логического типа
  нц для i от 1 до n - 1 | n - количество элементов массива
    если стержень1[i] < стержень1[i + 1]
      то
        упор := нет
        выход
    все
  кц
  знач := упор | Значение функции зависит
                | от окончательного значения переменной упор
кон
```

Задания для самостоятельной работы

1. Разработайте аналогичную программу на языке программирования, который изучаете.
2. Разработайте также варианты программы, в которых проводится проверка массивов на упорядоченность по убыванию, невозрастанию и возрастанию.

Программы присылайте в редакцию. Фамилии всех приславших будут опубликованы, а авторы лучших программ будут награждены дипломами.

Ханойские башни. Часть 3⁵

Д.М. Златопольский

В двух первых частях статьи были разработаны ряд программы, связанных с головоломкой «Ханойская башня».

Решим задачу сбора дисков с соблюдением правил головоломки, когда в исходном состоянии на стержне 1 диски размещены в произвольном порядке (а не в порядке увеличения их размеров при просмотре сверху вниз).

Что можно сделать в такой ситуации?

Прежде всего, если какое-то количество верхних дисков на стержне 1 представляет из себя гипотетическую Ханойскую башню (см. рис. 1), все они, кроме самого большого⁶, могут быть перенесены на стержень 3 согласно процедуре `Вывод_диаметров_дисков_и_номеров_стержней`, разработанной в предыдущем выпуске.



Рис. 1

Количество перемещаемых дисков (имя соответствующей переменной величины – `сколько_обычных`) определяется следующим образом:

```
сколько_обычных := 0
i := кол1 - 1 | Начинаем с предпоследнего
| Проверяем остальные
нц пока стержень1[i] > стержень1[i + 1] | Имено
    сколько_обычных := сколько_обычных + 1
    i := i - 1
кц
```

После этого на первом стержне наверху окажется какое-то количество дисков, образующих «перевернутую» Ханойскую башню (см. рис. 2).



Рис. 2

Эти диски можно по одному разместить на стержне 3. В приведённом ниже фрагменте программы количество таких дисков обозначено именем `сколько_переверн`.

```
сколько_переверн := 0
i := кол1 | Начинаем рассматривать верхний диск
нц пока стержень1[i] > стержень1[i - 1]
    сколько_переверн := сколько_переверн + 1
    | Моделируем перенос одного диска со стержня 1 на стержень 2
    кол2 := кол2 + 1
    стержень2[кол2] := стержень1[i]
    | Переходим к следующему (расположенному ниже) диску
    i := i - 1
кц
| И еще один (самый нижний из подходящих дисков)
сколько_переверн := сколько_переверн + 1
кол2 := кол2 + 1
```

⁵ Части 1 и 2 – см. интернет-журнал «Мир информатики», выпуск № 71 (октябрь 2022 г.) и выпуск № 72 (ноябрь 2022 г.).

⁶ Когда «обычную» Ханойскую башню образует только один верхний диск, то его, конечно, можно учесть и перенести на стержень 3, но в целях упрощения программы и ускорения времени её работы лучше это не делать, а включить этот диск в состав «перевернутой» башни (см. чуть ниже).

```

стержень2[кол2] := стержень1[i]
| Уменьшаем значение кол1
кол1 := кол1 - сколько_переверн

```

Конечно, значение `сколько_переверн` может быть равно 1 (также убедитесь в этом!).

Прежде чем представлять процедуру, моделирующую все описанные действия, заметим, что, поскольку «перевернутую» башню могут образовывать все диски до самого нижнего, то в условие для подсчета значения `сколько_переверн` следует дополнительно включить условие.

Вся процедура `Распределение_с_1_на_2_и_3`:

```

алг Распределение_с_1_на_2_и_3
нач цел сколько_обычных, сколько_переверн, i
    | Диски "обычной" башни
    | Определяю их количество
    сколько_обычных := 0
    i := кол1
    нц пока стержень1[i] < стержень1[i - 1]
        сколько_обычных := сколько_обычных + 1
        i := i - 1
    кц
    | Перемещаем
    Вывод_диаметров_дисков_и_номеров_стержней(сколько_обычных, 1, 3)
    | Диски «перевернутой» башни
    сколько_переверн := 0
    i := кол1      | Начинаем рассматривать верхний диск
    нц пока i - 1 > 0 и стержень1[i] > стержень1[i - 1]
        сколько_переверн := сколько_переверн + 1
        | Моделируем перенос одного диска со стержня 1 на стержень 2
        вывод нс, "диск диаметром ", стержень1[i],
            " со стержня 1 на стержень 2"
        кол2 := кол2 + 1
        стержень2[кол2] := стержень1[i]
        | Переходим к следующему (расположенному ниже) диску
        i := i - 1
    кц
    | И еще один (самый нижний из подходящих дисков)
    сколько_переверн := сколько_переверн + 1
    вывод нс, "диск диаметром ", стержень1[i],
        " со стержня 1 на стержень 2"
    кол2 := кол2 + 1
    стержень2[кол2] := стержень1[i]
    | Уменьшаем значение кол1
    кол1 := кол1 - сколько_переверн
кон

```

где процедура `Вывод_диаметров_дисков_и_номеров_стержней` — рекурсивная процедура, в которой происходит вывод номеров дисков, перемещаемых при решении задачи, и номеров стержней.

Для проверки правильности работы созданной процедуры целесообразно разработать отдельную программу. В ней должны быть следующие основные этапы:

1. Заполнение массива `стержень1` неповторяющимися случайными числами.
2. Вывод массива `стержень1`.
3. Вызов процедуры `Распределение_с_1_на_2_и_3`.
4. Вывод всех трёх массивов.

После выполнения процедуры Распределение на стержнях будет ситуация, показанная на рис. 3 (все формы и размеры – условные, на третьем стержне дисков может не быть).

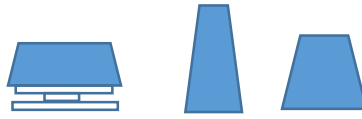


Рис. 3

Что дальше? Приведенная на рис. 3 ситуация похожа на задачу о трех «заброшенных» Ханойских башнях, которая была решена в предыдущем выпуске нашего журнала.

Отличие только в том, что на стержне 1 не все диски образуют «обычную» башню, а только выделенная их часть. Это значит, что в процедуре Неполный_сбор_дисков_с_двух_стержней, разработанной в предыдущем выпуске, при подсчете значения переменной сколько1 следует учесть дополнительное условие $\text{стержень1}[i] < \text{стержень1}[i - 1]$. Это условие соответствует выделенным дискам, кроме самого нижнего из них. Возможно, следует учесть и его – его можно перенести, если он меньше верхних дисков на стержнях 2 и 3.

Измененная с учетом этого процедура Неполный_сбор_дисков_с_двух_стержней, примет вид:

```

алг Неполный_сбор_дисков_с_двух_стержней(арг цел a1, a2)
нач цел i
  | Три варианта
  выбор
    при a1 = 1 и a2 = 2:
      если стержень1[кол1] < стержень2[кол2]
        то | Перекладываем с 1-го стержня на 2-й
          сколько1 := 0
          i := кол1
          нц пока i > 0 и стержень1[i] < стержень2[кол2]
            и стержень1[i] < стержень3[кол3]
            и стержень1[i] < стержень1[i - 1] | Новое условие
            сколько1 := сколько1 + 1
            i := i - 1
          кц
          | Проверяем, можно ли перенести еще один диск
          если стержень1[i] < стержень3[кол3]
            и стержень1[i] < стержень2[кол2]
              то
                сколько1 := сколько1 + 1
          все
          Вывод_диаметров_дисков_и_номеров_стержней(сколько1, 1, 2)
      иначе | Перекладываем с 2-го стержня на 1-й
        сколько2 := 0
        i := кол2
        нц пока i > 0 и стержень2[i] < стержень1[кол1]
          и стержень2[i] < стержень3[кол3]
          сколько2 := сколько2 + 1
          i := i - 1
        кц
        Вывод_диаметров_дисков_и_номеров_стержней(сколько2, 2, 1)
      все
    при a1 = 1 и a2 = 3:
      если стержень1[кол1] < стержень3[кол3]
        то | Перекладываем с 1-го стержня на 3-й
          сколько1 := 0

```

```

i := кол1
нц пока i > 0 и стержень1[i] < стержень3[кол3]
    и стержень1[i] < стержень2[кол2]
    и стержень1[i] < стержень1[i - 1] | Новое условие
    сколько1 := сколько1 + 1
    i := i - 1
кц
| Проверяем, можно ли перенести еще один диск
если стержень1[i] < стержень3[кол3]
    и стержень1[i] < стержень2[кол2]
    то
        сколько1 := сколько1 + 1
все
Вывод_диаметров_дисков_и_номеров_стержней(сколько1, 1, 3)
иначе | Перекладываем с 3-го стержня на 1-й
    сколько3 := 0
    i := кол3
    нц пока i > 0 и стержень3[i] < стержень1[кол1]
        и стержень3[i] < стержень2[кол2]
        сколько3 := сколько3 + 1
        i := i - 1
    кц
    Вывод_диаметров_дисков_и_номеров_стержней(сколько3, 3, 1)
все
при a1 = 2 и a2 = 3:
    если стержень2[кол2] < стержень3[кол3]
        то | Перекладываем с 2-го стержня на 3-й
            сколько2 := 0
            i := кол2
            нц пока i > 0 и стержень2[i] < стержень3[кол3]
                и стержень2[i] < стержень1[кол1]
                сколько2 := сколько2 + 1
                i := i - 1
            кц
            Вывод_диаметров_дисков_и_номеров_стержней(сколько2, 2, 3)
        иначе | Перекладываем с 3-го стержня на 2-й
            сколько3 := 0
            i := кол3
            нц пока i > 0 и стержень3[i] < стержень2[кол2]
                и стержень3[i] < стержень1[кол1]
                сколько3 := сколько3 + 1
                i := i - 1
            кц
            Вывод_диаметров_дисков_и_номеров_стержней(сколько3, 3, 2)
    все
все
кон

```

Разработаем также аналогичную процедуру неполного сбора дисков с двух стержней для случая, когда на третьем стержне дисков нет (такая ситуация, естественно, возможна, и соответствующая процедура понадобится нам в будущем). В новой процедуре, которой дадим имя Неполный_сбор_дисков_с_двух_стержней_2, не должно быть условий, связанных с третьим («пустым» стержнем, а не обязательно стержнем с номером 3).

```

алг Неполный_сбор_дисков_с_двух_стержней_2(арг цел a1, a2)
нач цел i
  | Три варианта
  выбор
    при a1 = 1 и a2 = 2:
      если стержень1[кол1] < стержень2[кол2]
      то | Перекладываем с 1-го стержня на 2-й
        сколько1 := 0
        i := кол1
        нц пока i > 0 и стержень1[i] < стержень2[кол2]
          и стержень1[i] < стержень3[кол3] | Удаляется
          и стержень1[i] < стержень1[i - 1]
          сколько1 := сколько1 + 1
          i := i - 1
        кц
        | Проверяем, можно ли перенести еще один диск
        если стержень1[i] < стержень2[кол2]
          и стержень1[i] < стержень3[кол3]
          то
            сколько1 := сколько1 + 1
        все
        Вывод_диаметров_дисков_и_номеров_стержней(сколько1, 1, 2)
      иначе | Перекладываем с 2-го стержня на 1-й
        сколько2 := 0
        i := кол2
        нц пока i > 0 и стержень2[i] < стержень1[кол1]
          и стержень2[i] < стержень3[кол3]
          сколько2 := сколько2 + 1
          i := i - 1
        кц
        Вывод_диаметров_дисков_и_номеров_стержней(сколько2, 2, 1)
      все
    при a1 = 1 и a2 = 3:
      если стержень1[кол1] < стержень3[кол3]
      то | Перекладываем с 1-го стержня на 3-й
        сколько1 := 0
        i := кол1
        нц пока i > 0 и стержень1[i] < стержень3[кол3]
          и стержень1[i] < стержень2[кол2]
          и стержень1[i] < стержень1[i - 1]
          сколько1 := сколько1 + 1
          i := i - 1
        кц
        | Проверяем, можно ли перенести еще один диск
        если стержень1[i] < стержень3[кол3]
          и стержень1[i] < стержень2[кол2]
          то
            сколько1 := сколько1 + 1
        все
        Вывод_диаметров_дисков_и_номеров_стержней(сколько1, 1, 3)
      иначе | Перекладываем с 3-го стержня на 1-й
        сколько3 := 0
        i := кол3
        нц пока i > 0 и стержень3[i] < стержень1[кол1]
          и стержень3[i] < стержень2[кол2]
          сколько3 := сколько3 + 1
          i := i - 1
        кц

```

```

        Вывод_диаметров_дисков_и_номеров_стержней(сколько3, 3, 1)
    все
при a1 = 2 и a2 = 3:
    если стержень2[кол2] < стержень3[кол3]
    то | Перекладываем с 2-го стержня на 3-й
        сколько2 := 0
        i := кол2
        нц пока i > 0 и стержень2[i] < стержень3[кол3]
            и стержень2[i] < стержень1[кол1]
            сколько2 := сколько2 + 1
            i := i - 1
        кц
        Вывод_диаметров_дисков_и_номеров_стержней(сколько2, 2, 3)
    иначе | Перекладываем с 3-го стержня на 2-й
        сколько3 := 0
        i := кол3
        нц пока i > 0 и стержень3[i] < стержень2[кол2]
            и стержень3[i] < стержень1[кол1]
            сколько3 := сколько3 + 1
            i := i - 1
        кц
        Вывод_диаметров_дисков_и_номеров_стержней(сколько3, 3, 2)
    все
все
кон

```

Напомним, что при решении задачи о трех «заброшенных» Ханойских башнях использовалась процедура Неполный_сбор_дисков_с_трех_стержней:

```

алг Неполный_сбор_дисков_с_трех_стержней
нач цел стержень_с_макс_дискком
    | Определяем, где самый большой верхний диск
    если стержень1[кол1] > стержень2[кол2]
        и стержень1[кол1] > стержень3[кол3]
    то
        стержень_с_макс_дискком := 1
        Неполный_сбор_дисков_с_двух_стержней(2, 3)
    иначе
        если стержень2[кол2] > стержень1[кол1]
            и стержень2[кол2] > стержень3[кол3]
        то
            стержень_с_макс_дискком := 2
            Неполный_сбор_дисков_с_двух_стержней(1, 3)
        иначе
            стержень_с_макс_дискком := 3
            Неполный_сбор_дисков_с_двух_стержней(1, 2)
    все
все
кон

```

Итак, если в ситуации, показанной на рис. 3, использовать процедуру Неполный_сбор_дисков_с_трех_стержней, то будут два варианта состояния стержней:

- 1) диски имеются на всех трёх стержнях;
- 2) диски есть только на каких-то двух стержнях.

Соответствующие варианты действий запишем в процедуре с аналогичным именем:

```
алг Варианты_действий
нач
  если кол1 > 0 и кол2 > 0 и кол3 > 0
  то
    Неполный_сбор_дисков_с_трех_стержней
  иначе | Диски есть только на двух стержнях
    если кол1 > 0 и кол2 > 0 или кол1 > 0 и кол3 > 0
    или кол2 > 0 и кол3 > 0
    то
      выбор
        при кол1 > 0 и кол2 > 0:
          Неполный_сбор_дисков_с_двух_стержней_2(1, 2)
        при кол1 > 0 и кол3 > 0:
          Неполный_сбор_дисков_с_двух_стержней_2(1, 3)
        при кол2 > 0 и кол3 > 0:
          Неполный_сбор_дисков_с_двух_стержней_2(2, 3)
      все
    все
  все
кон
```

Все эти возможные действия (по процедуре Варианты_действий) должны повторяться пока все диски не будут собраны на одном из стержней⁷.

```
алг Неполный_сбор_с_трех_стержней
нач
  Распределение_с_1_на_2_и_3
  нц пока не ((кол1 > 0 и кол2 = 0 и кол3 = 0)
    или (кол2 > 0 и кол1 = 0 и кол3 = 0))
  Варианты_действий
  . кц
кон
```

В результате:

– если диски будут собраны на стержне 2, то они обязательно будут собраны в нужном порядке⁸;

– если диски будут собраны на стержне 1, то это не обязательно.

Во втором случае надо повторять процедуру Неполный_сбор_с_трех_стержней, пока и на стержне 1 диски не будут упорядочены в соответствии с требованиями. Это значит, что основная часть программы должна быть оформлена так:

```
| Глобальные переменные
цел n, кол1, кол2, кол3
цел таб стержень1[1 : 30], стержень2[1 : 30], стержень3[1 : 30]
цел сколько1, сколько2, сколько3

алг Башня
нач цел i
  | Ввод количества дисков
  вывод нс, "Введите количество дисков "
  ввод n
  Размещение_дисков
  Вывод_дисков
  нц пока не Отсорт(1) и не Отсорт(2)
```

⁷ Это могут быть только стержни 1 или 2. Доказательство этого мы оставляем за рамками статьи.

⁸ Утверждаем это также без доказательства.

```

        Неполный_сбор_с_трех_стержней
    кц
    Вывод_дисков
кон

```

Осталось подготовить функцию, проверяющую правильность размещения дисков на стержне (проверяющую упорядоченность элементов массива по убыванию). Эта функция логического типа аналогична такой же функции, описанной в статье «Две типовые задачи программирования» в начале данного выпуска. Но так как проверять надо упорядоченность одного из двух массивов, то это должна быть функция с аргументом (с параметром), соответствующим номеру стержня. Поэтому в ней должны быть рассмотрены два варианта в зависимости от этого номера (номер_стержня). Кроме того, следует учесть, что проверяемый массив может быть пустым. В этом случае массив должен считаться неупорядоченным (значение функции равно **нет**).

Вот соответствующая функция

```

алг лог Отсорт(арг цел номер_стержня)
нач цел i, лог упор
выбор
    при номер_стержня = 1:
        если кол1 = 0
            то
                знач := нет
            иначе
                упор := да
                нц для i от 1 до кол1 - 1
                    если стержень1[i] < стержень1[i + 1]
                        то
                            упор := нет
                            выход
                все
            кц
            знач := упор
        все
    при номер_стержня = 2:
        если кол2 = 0
            то
                знач := нет
            иначе
                упор := да
                нц для i от 1 до кол2 - 1
                    если стержень2[i] < стержень2[i + 1]
                        то
                            упор := нет
                            выход
                все
            кц
            знач := упор
        все
    все
кон

```

Всю программу (на языке программирования, который вы изучаете), решающую задачу сбора всех дисков, изначально расположенных на стержне 1 в произвольном порядке, согласно правилам решения головоломки «Ханойская башня», разработайте самостоятельно. Программы присылайте в редакцию. Все приславшие правильные программы будут награждены дипломами.

Ответы, решения, разъяснения

к заданиям, опубликованным в журнале «Мир информатики» № 70
(сентябрь 2022 г.)

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

«Стремление решить задачу заложено в самой природе человека».

Джордж Пойя

Ребусы по информатике

Ответы

Ребус № 1: КОДИРОВАНИЕ. Ребус № 2: ОБРАБОТКА. Ребус № 3: МОНИТОР.
Ребус № 4: ПЕРЕДАЧА.

Задача «Дмитрий и новые коллеги»

Напомним, что требовалось по приведенной информации определить имена коллег программиста Дмитрия и цвет их волос.

Решение

Человек, который работает в компании дольше всех, — не рыжий и не брюнет. Значит, он блондин. У этого же человека нет братьев и сестёр, поэтому он точно не Пётр. Рыжий коллега тоже не может им быть. Получается, что Пётр — брюнет.

Брюнет Пётр пришёл в компанию раньше рыжего коллеги и позже Игоря. Следовательно, Игорь — блондин, который работает дольше всех, а Михаил — рыжий.

Ответ: Игорь — блондин, Пётр — брюнет, Михаил — рыжий.

Оригинальная система счисления

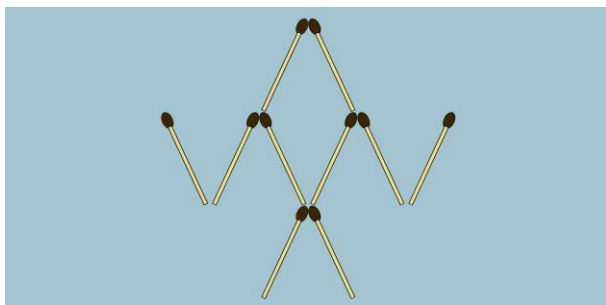
Напомним, что следовало определить числовой эквивалент символов, использованных в некоторой непозиционной системе счисления, в которой цифры обозначаются геометрическими фигурами. Были представлены некоторые числа этой системы счисления и соответствующие им числа десятичной системы счисления:

Неизвестная система	Десятичная система
	9
 	11
  	19
  	190
  	1900

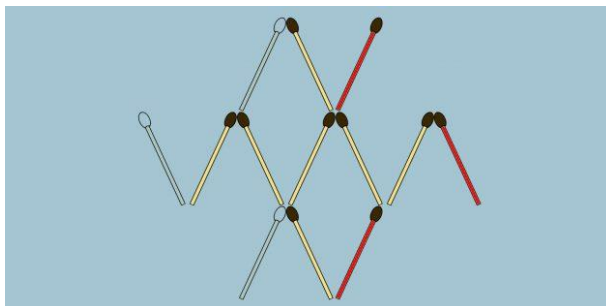
Ответ:  = 1,  = 10,  = 100,  = 1000. Чтение многозначных чисел во многом аналогично числам, представленным в римской системе счисления.

Головоломка «Рак из спичек»

Напомним, что требовалось переставить три спички так, чтобы изображенный из спичек рак полз вниз, а не вверх.



Решение



Перемещаемые спички оформлены серым цветом, «новые» – красным.

Задача «Девять чисел в таблице»

Напомним условие: «В клетки таблицы 3×3 вписаны числа от 1 до 9. Катя нашла сумму чисел в каждом из квадратов 2×2 , а затем сложила полученные суммы. Какова наименьшая возможная сумма этих четырёх сумм?»

Решение

Запишем «координаты» клеток таблицы, как это делается в компьютерных программах, которые называют «электронными таблицами» (см. рисунок).

	A	B	C
1	9	5	6
2	4	1	2
3	8	3	7

Тогда сумма всех четырёх сумм будет равна $4 \times B2 + 2 \times (B1 + A2 + B3 + C2) + (A1 + C1 + A3 + C3)$. Это значит, что наименьшее число (1) нужно записать в клетку B2, а следующие четыре наименьших числа в клетки B1, A2, B3, C2 (на рисунке приведен один из возможных вариантов). Дальнейшее решение очевидно.

Ответ: 62.

Задача «Числа-палиндромы»

Напомним, что следовало найти такое десятичное число X , не превышающее 10, которое является палиндромом в трёх различных системах счисления (число-палиндром — это число, одинаково читаемое как слева направо, так и справа налево, например, 707_8 , 2112_4 и т. п.).

Ответ: это число 10, которое является палиндромом в троичной (101_3), четверичной (22_4) и девятеричной (11_9) системах счисления.

Задача «Какая буква пропущена?»

Напомним, что требовалось определить, что за последовательность приведена ниже и выяснить, какую пропущенную букву нужно вставить вместо квадратика.

П В С П С В

Ответ

В последовательности приведены первые буквы дней недели (П – понедельник, В – вторник и т. д.). Пропущена буква «Ч» (четверг).

Валико Думаян и Ростислав Колисниченко, ученики средней школы рабочего поселка Пинеровка, Саратовская обл., Балашовский р-н (учитель **Пичугин В.В.**), предложили такую задачу: «Дана последовательность О, Д, Т, Ч, ... Продолжить её еще пятью буквами». Решите, пожалуйста, и её.

Задача «Робинзона Крузо и таблетки»

Напомним условие: «Однажды на необитаемом острове Робинзона Крузо укусила очень ядовитая змея.

Случайно, у Робинзона были с собой четыре таблетки: две таблетки лечебные, а две – для усиления лечебного эффекта. Причём эти таблетки необходимо было принимать только попарно: одну лечебную вместе с дополнительной, которая сама по себе не спасает от яда.

Первый раз надо принять одну лечебную таблетку и одну – дополнительную к ней. Также надо поступить и на следующий день.

Но, к несчастью, все четыре таблетки оказались в одной баночке, и проблема в том, что они по внешнему виду совершенно не отличаются друг от друга. Теперь непонятно, где лечебные, а где – дополнительные к ним.

Как может поступить Робинзон Крузо, чтобы спастись?»

Решение

Чтобы спастись, Робинзон Крузо должен поступить так. Разделить все таблетки пополам и отложить одну половину каждой таблетки (всего их будет 4) в одну группу, а вторую половину каждой таблетки – в другую. В первый день он должен принять две половинки из первой группы и две — из второй. Аналогично – и на следующий день.

В ряде ответов предлагался такой вариант: «Стереть в порошок все таблетки, хорошо перемешать их и поделить на 2 части. Одну часть выпить сразу, а вторую на следующий день». Считаем такое решение «опасным для Робинзона Крузо», так как при перемешивании дозировки лечебных и дополнительных порошков могут оказаться разными. Использование же половинок таблеток гарантирует правильный приём лекарства.

Вторая задача «Робинзона Крузо и таблетки»

По другой версии у Робинзона были с собой не четыре таблетки, а восемь – четыре таблетки лечебные, а четыре – для усиления лечебного эффекта. Но принимать их, как и прежде, нужно только попарно: одну лечебную вместе с дополнительной, которая сама по себе не спасает от яда. Первый раз надо принять одну лечебную таблетку и одну – дополнительную к ней. Также надо поступить и в течение следующих трех дней. Как должен поступить Робинзон Крузо, чтобы спастись в этом случае?»

Задачи о «шутниках» и «серьёзных». Часть 2

Мы продолжаем (см. предыдущий выпуск интернет-журнала «Мир информатики») публиковать задачи об учениках одного класса, которые для игры разбились на две «партии»: серьёзных, отвечающих правдиво на любой вопрос, и шутников, дающих только неправильные ответы. Полезные советы по решению таких задач также даны в предыдущем выпуске нашего журнала.

5. Митя сказал Вите: «Ты должен меня назвать шутником». К какой «партии» относится каждый из них?

6. Учитель спросил каждого из двух учеников, к какой «партии» относится его товарищ. Оба ответили одинаково. Можно ли определить, кто из учеников шутник, а кто – серьёзный?

7. Учитель спросил каждого из двух учеников, к какой «партии» относится его товарищ. Может ли случиться так, что их ответы будут разными?

8. Мог ли состояться такой диалог: «Митя сказал Вите: “Ты должен меня назвать серьёзным”, а тот ответил: “А ты меня – шутником”?»

9. Митя сказал Вите: «Ты должен меня назвать серьёзным». Тот ответил: «И ты меня тоже». Можно ли определить, к какой «партии» относится каждый из них?

10. Митя сказал Вите: «Ты должен меня назвать шутником». Тот ответил: «И ты меня тоже». Можно ли определить принадлежность ребят к той или иной «партии»?

11. Митя сказал Вите: «Ты должен меня назвать шутником. Тот ответил: «А ты меня – серьёзным». Возможен ли такой диалог между учениками?

Шестнадцать предложений

Какие из следующих предложений являются высказываниями:

- а) Санкт-Петербург — столица России;
- б) Студент физико-математического факультета педагогического университета;
- в) Треугольник ABC подобен треугольнику A'B'C';
- г) Луна есть спутник Марса;
- д) $2 + 2 = 5$;
- е) Кислород — газ;
- ж) Каша — вкусное блюдо;
- з) Математика — интересный предмет;
- и) Картины Пикассо слишком абстрактны;
- к) Железо тяжелее свинца;
- л) «Да здравствует знание!»;
- м) Треугольник называется равнобедренным, если все его стороны равны;
- н) Если в треугольнике все углы равны, то он равнобедренный;
- о) Сегодня плохая погода;
- п) В романе А. С. Пушкина «Евгений Онегин» 136 245 букв;
- р) Река Ангара впадает в озеро Байкал

Поровну

Если число 3400 записать в двоичной системе счисления, получится такое выражение:
110101001000

В этой записи участвуют всего две цифры. Но их неодинаковое количество. Нулей больше, чем единиц.

Найдите *две* такие системы счисления, что в каждой из них число 3400 выражается также при помощи только двух цифр. Причём, в равных количествах.

Три друга

На одном заводе работают три друга — слесарь, токарь и сварщик. Их фамилии Борисов, Иванов и Семенов. У слесаря нет ни братьев, ни сестер. Он — самый младший из друзей. Семенов, женатый на сестре Борисова, старше токаря. Назовите фамилии слесаря, токаря и сварщика.

Шахматный турнир

В шахматном турнире принимали участие шесть партнеров разных профессий: токарь, слесарь, инженер, учитель, врач и шофер.

Известно, что:

- 1) в первом туре Андреев играл с врачом, учитель — с Борисовым, а Григорьев — с Евдокимовым; в первом туре 3 партии;
- 2) во втором туре Дмитриев играл с токарем, а врач — с Борисовым; во втором туре 2 партии;
- 3) в третьем туре Евдокимов играл с инженером; в третьем туре 1 партия; тем временем токарь обсуждал с учителем ход игры во втором туре;
- 4) по окончании турнира места распределились так: Борисову присудили 1-е место, Григорьев и инженер поделили 2-е и 3-е места, Дмитриев занял 4-е место,

У кого какая профессия?

Длина кратчайшего пути

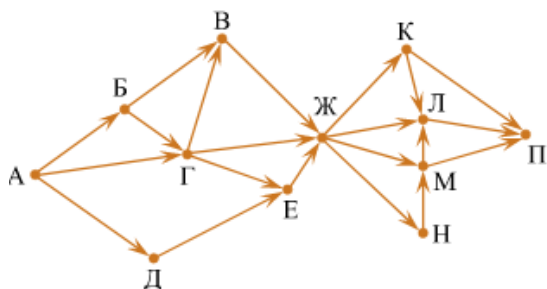
Между населёнными пунктами А, В, С, D, E, F построены дороги, протяжённость которых (в километрах) приведена в таблице:

	A	B	C	D	E	F
A		3	5			15
B	3		3			
C	5	3		5	2	
D			5			3
E			2			7
F	15			3	7	

Определите длину кратчайшего пути между пунктами А и F. Передвигаться можно только по дорогам, протяжённость которых указана в таблице.

Из города А в город П

На рисунке ниже приведена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, К, Л, М, Н, П. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.



Сколько существует различных путей из города А в город П, проходящих через город Н?

Ответы и решения задач и головоломок, опубликованных в сентябрьском выпуске интернет-журнала «Мир информатики», представили:

- Арсланова Алина, г. Пенза, школа № 73, учитель **Диков А.В.**;
- Барсков Максим, Изотов Павел, Корнилова Виктория, Назарян Эмил, Павлов Алексей, Пушкарев Захар, Пыров Макар, Смирнова Полина, Степанов Никита и Сутырина Ксения, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;
- Валуев Иван, Гаязов Рашид и Хорькова Анна, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;
- Вахрина Оксана, Владимирская обл., г. Александров, школа № 13, учитель **Дуванский И. А.**;
- Ващенко Светлана и Хомутова Евгения, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;
- Думанян Валико и Колисниченко Ростислав, средняя школа рабочего поселка Пинеровка, Саратовская обл., Балашовский р-н, учитель **Пичугин В.В.**;
- Муарисов Данила, Пантюшкин Георгий, Попкова Мария, Резцова Анастасия, Сангинов Хабиббуло и Ягиева Гуля, средняя школа села Ириновка, Новобурасский р-н Саратовской обл., учитель **Брунов А.С.**;
- Сухомлинов Владислав и Тихомирова Виктория, Москва, школа № 1530 «Школа Ломоносова», учитель **Каменецкая Т.С.**

Авторы лучших ответов будут награждены дипломами.

Программы решения задачи, предложенных для самостоятельной работы в статье «Методы заполнения массивов», прислали:

- Бояринов Алексей, г. Пенза, школа № 73, учитель **Диков А.В.**;
- Степанов Никита, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;
- Сухомлинов Владислав, Москва, школа № 1530 «Школа Ломоносова», учитель **Каменецкая Т.С.**

Валико Думанян и Ростислав Колисниченко из посёлка Пинеровка Саратовской обл. написали в редакцию о том, что статья о календаре из рубрики «Это полезно знать» заставила их задуматься о возможности алгоритмического вычисления дат. Оказалось, есть много разработанных алгоритмов «вечного календаря». После обсуждения с учителем на кружке разных подходов к определению дня недели по заданной дате они разработали соответствующие программы на языке программирования Python. Спасибо, Валико и Ростислав!

Напомним, что в этой рубрике редакция проводит разбор задач, решение которых вызвало трудности.

Десять монет и клоп

Напомним условие: «Есть 10 монет. Шесть из них весом ровно 10 г и четыре — ровно 11 г. Монеты выглядят совершенно одинаково. Имеются очень точные двухчашечные весы, но, к сожалению, на результат взвешивания влияет вредный невидимый клоп, который при каждом взвешивании садится на чашу весов, но на какую именно — неизвестно. Вес клопа ровно 1 г. (Клоп не садится на весы, если на них ничего не взвешивается.)

За какое минимальное число взвешиваний без гирь можно обнаружить вредное насекомое, если монеты нельзя взвешивать повторно?»

Анализ решения

Прежде всего, заметим, что «минимальное количество взвешиваний» в условии означает, что должны рассматриваться самые неблагоприятные варианты, а не варианты типа «А вдруг повезёт!».

Например, если сравнить на весах две любые монеты, и они окажутся одного веса (обе по 10 г или обе по 11 г), то клоп будет находиться на той чаше весов, которая перевесит.

А если монеты будут разного веса? В этом случае возможны два варианта:

- 1) клоп сядет на чашу с более тяжелой монетой, и при этом чаша также перевесит;
- 2) клоп сядет на чашу с монетой весом 10 г, то есть в этом случае будет равновесие. Но на какой именно чаше будет находиться насекомое, определить будет нельзя.

Последний вариант и является неблагоприятным, и именно его следует учесть при анализе. Итак, первое взвешивание не позволит найти ответ.

Дальнейший анализ (второе взвешивание и т. д.) проведите самостоятельно. Ответ (минимальное число взвешиваний) направьте в редакцию.

Решите также похожую задачу с двумя монетами весом 10 и 11 г. Но в ней, в отличие от рассмотренной, клоп, чтобы «скрыться» поочередно садится то на одну чашу весов, то на другую. За какое минимальное число взвешиваний без гирь можно обнаружить вредное насекомое в этом случае?

Карло Анчелотти

Чем знаменит как тренер Карло Анчелотти (Италия), кроме того, что он:

- первый и единственный тренер, выигравший футбольные чемпионаты пяти стран Европы, а также 4 кубка Лиги чемпионов УЕФА;
- единственный тренер, руководивший командами в пяти финалах Лиги чемпионов (2003, 2005, 2007, 2014, 2022 годы);
- дважды выигрывал Клубный чемпионат мира;
- является одним из семи человек, которые выигрывали Кубок европейских чемпионов и как игрок, и как тренер.

Получить верное равенство

Как, переложив всего одну спичку, получить верное равенство?

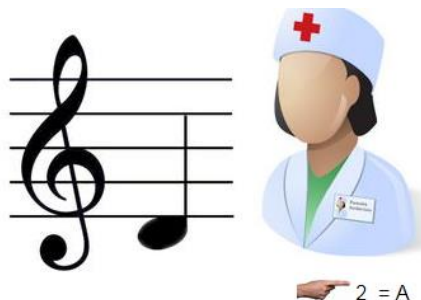


Пять ребусов

Решите, пожалуйста, ряд ребусов.

Ребус № 1 (два слова)

Т, 3! ЪЩ



2 = А

Ребус № 2 (два слова)

Э, [bread] К [cake] [faucet] Я

4 = Н

Т [antler] [fox]

3 = Ц

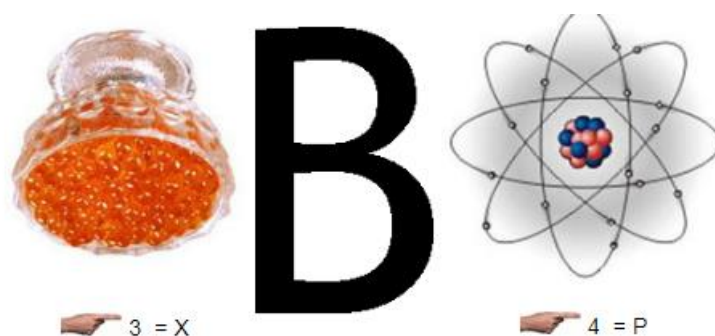
Ребус № 3 (два слова)



Ребус № 4



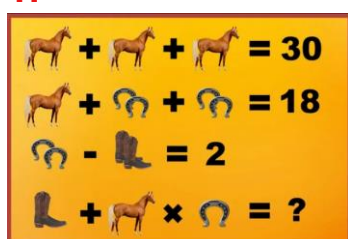
Ребус № 5



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

Две головоломки без слов

1.



$$6+4=210$$

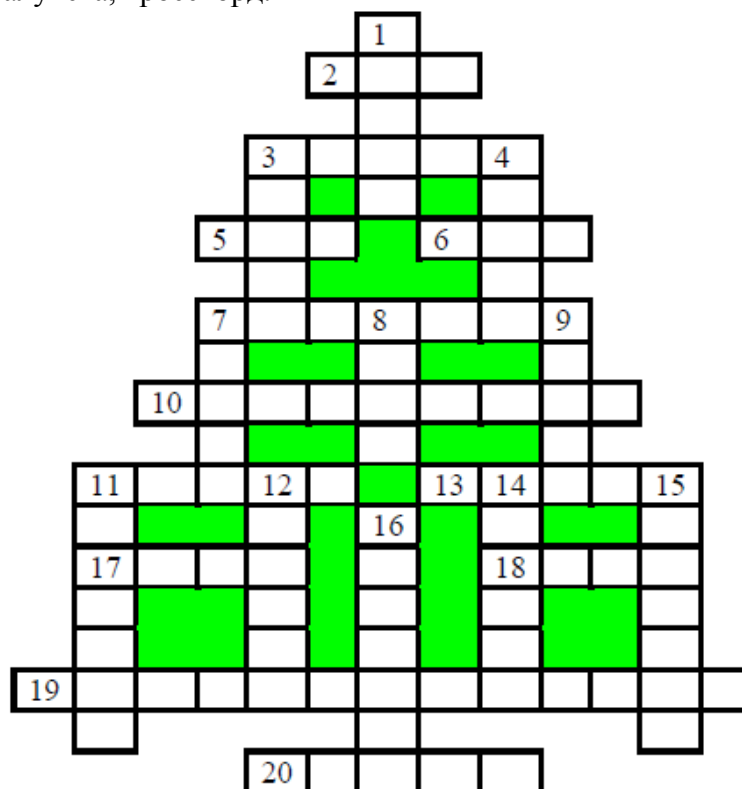
$$9+2=711$$

$$8+5=313$$

2. $15+3=...$

Новогодний кроссворд-“елочка”

Решите, пожалуйста, кроссворд.



По горизонтали

2. Число в системе условных обозначений символов.
3. Прямоугольник, ограничивающий меню или т.п.
5. ... координат.
6. Цифра восьмеричной системы счисления.
7. Последовательная смена состояний, стадий развития.
10. Строение, внутреннее устройство.
11. Неисправность в работе компьютера.
13. Величина, с помощью которой осуществляется счет.



17.

18. Так называют любую запись на странице в Интернете (сообщение, вопрос, ответ, комментарий и т.п.).

19. Один из двух видов транслятора — системной программы, которая «переводит» программу с языка программирования высокого уровня в машинные команды.

20. Единица измерения высоты шрифта.

По вертикали

1. Один из объектов баз данных, а также внешнее очертание, наружный вид предмета.
3. Массив точек, из которых формируется изображение на экране.
4. Координаты ячейки в электронной таблице.
7. Непрерывная последовательность данных.
8. Алгоритмическая конструкция, обеспечивающая повторение одних и тех же операций.
9. Отмена всех установленных параметров компьютера.

11. (Почти ребус)



$O_2 = H$

12. Устройство управления работой шины персонального компьютера, а также спортивный судья.

14. Ввоз товаров из-за рубежа, а также вставка в документ приложений Windows объектов из других приложений.

15. Один из результатов целочисленного деления.

16. Язык программирования.



Старый магнитный конструктор

Митя обнаружил на чердаке старый магнитный конструктор, на котором был выложен прямоугольник из цветных элементов.

Почему именно так выложены элементы?

Итоги конкурса № 51

Напомним, что в качестве задания этого конкурса было предложено ответить на вопрос: «Какие математические инструменты изображены на картине с выдающимся русским учёным Михаилом Васильевичем Ломоносовым, сидящим за столом?»

Ответ

Речь идёт о портрете И.В. Ломоносова, автором которой является неизвестный художник (ниже приведена копия художника Г. К. И. фон Преннера, 1755 г., с несохранившегося оригинала).



На столе у учёного имеются циркуль, транспортир и угольник.

Правильные ответы прислали и стали победителями конкурса:

— Борисенко Степан и Глухова Марина, г. Пенза, школа № 73, учитель **Диков А.В.**;

— Вахрина Оксана, Владимирская обл., г. Александров, школа № 13, учитель **Дуванский И. А.**;

— Каминская Елизавета и Приходько Матвей, г. Петрозаводск, Державинский лицей, учитель **Флотская Т.В.**;

— Лазуткина Мария, Макаренко Тарас и Шестерякова Владислава, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Пушкарев Захар и Сутырина Ксения, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Сухомлинов Владислав, Москва, школа № 1530 «Школа Ломоносова», учитель **Каменецкая Т.С.**

Итоги конкурса № 52

Напомним, что в качестве задания этого конкурса было предложено ответить на ряд вопросов.

Ответы

1. Б. Луна (холодная Луна светит отражённым светом, остальные сами являются источником света).

2. Б. Лишней является дробь $\frac{3}{6}$ (это же $\frac{1}{2}$).

3. В. 3 (Hello, world!). Традиция использования фразы «Hello, world!» («Привет, мир!») появилась в 1978 году и поддерживается до сих пор.

4. Единственный язык, напрямую выполняемый ЭВМ, — это машинный язык (вариант ответа В).

5. Цепочку цифр 1000011000111... продолжают цифры 00111... (вариант Б). В группах из пяти цифр количество единиц растёт, а нулей — уменьшается: 100001100011100111...

6. Чтобы стать Frontend-разработчиком, то нужно освоить HTML, CSS и JavaScript (вариант ответа Б).

7. Среди представленных фрагментов кода на языке программирования Basic написан фрагмент PRINT "Привет" (вариант В).

Остальные языки программирования (по порядку): C++, Java, и последний, созданный российскими программистами язык Kotlin. Между прочим, объявленный в 2019 году Google приоритетным языком программирования для разработки Android-приложений.

8. Если предположить, что птицы с рождения умеют программировать, некоторые птицы — жаворонки и все жаворонки обязательно знают три языка программирования, то из этого следует, что любая птица, знающая меньше трех языков, жаворонком не будет (вариант ответа А).

9. Из утверждений «Некоторые программисты любят математику» и «Некоторые любители математики любят слушать Баха». Из этого утверждений нельзя с уверенностью сказать, есть ли среди программистов поклонники великого композитора (вариант Г).

Наибольшее число правильных ответов прислали и признаны победителями конкурса:

— Архипов Никита, Бондаренко Дарья и Копин Андрей, Ардатовский областной многопрофильный техникум, посёлок Ардатов Нижегородской обл., преподаватель **Зудин В.П.**;

— Барсков Максим, Изотов Павел и Сысуев Илья, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Думанян Валико и Колисниченко Ростислав, средняя школа рабочего посёлка Пинеровка, Саратовская обл., Балашовский р-н, учитель **Пичугин В.В.**;

— Игнатов Даниил, Назарьева Альбина, Суховерхова Софья и Чеканова Ольга, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;

— Каминская Елизавета и Приходько Матвей, г. Петрозаводск, Державинский лицей, учитель **Флотская Т.В.**

Конкурс № 55 “Детский набор кубиков”.

В детском наборе кубиков-цифр остались лишь кубики с цифрами 4, 5, 8, 9 и 7. Каким образом из данных цифр можно изобразить число 3400?



Укажите в письме свои фамилию и имя, населенный пункт, номер школы и фамилию, имя и отчество учителя информатики. Срок представления ответов — 31 января 2023 г.

