

«Мир информатики»

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 52, январь 2021 г.

Games.exe

Головоломка «Меледа»

Головоломка «Меледа» — старинная русская головоломка, происходящая, предположительно, из Китая. Она состоит из замкнутой проволочной «вилки» («челнока»), имеющей вид длинной шпильки для волос и воткнутой обоими свободными концами в рукоятку, и нескольких колец, связанных между собой довольно сложным образом (см. рис. 1–3, 5). Задача состоит в том, чтобы снять всю систему колец с вилки или надеть её обратно.



Рис. 1



Рис. 2



Рис. 3

Французское название головоломки — *baguenaudier*, немецкое — *das magische Ringspiel*. Калмыцкий вариант головоломки называется *нярн шинж* (буквально — «тонкое соображение»).



Рис. 4

Приведем правила, которые необходимо соблюдать при решении головоломки. Последнее (на рис. 5 — крайнее правое) кольцо можно всегда снять или надеть на вилку. Любое другое кольцо можно снять или надеть только тогда, когда кольцо непосредственно справа от него надето, а все остальные справа — сняты.

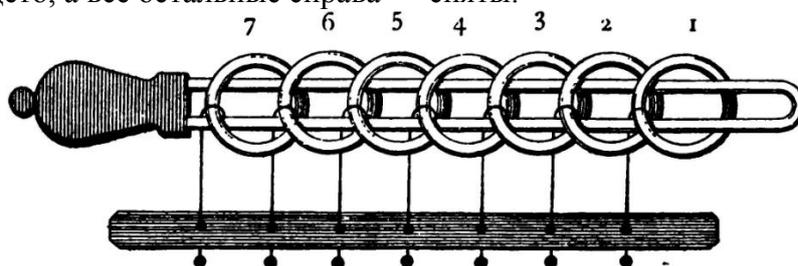


Рис. 5

Предлагаем читателям снять кольца с модели меледы с четырьмя кольцами. Её можно моделировать в виде 4-значного числа, в котором 1 соответствует надетому кольцу, а 0 — снятому. Так, в исходном состоянии модель головоломки будет иметь вид 1111, при этом последним кольцом условимся считать крайнюю правую единицу.

Сразу же заметим, что при большом количестве колец задача становится сложной и длительной даже при известном алгоритме ее решения. Недаром название головоломки связано со старым русским глаголом *меледиться* — заниматься пустым и мешкотным делом.

Решение головоломки оформите в виде:

1. Снять ... кольцо
2. Надеть ... кольцо
3. Снять ... кольцо
- ...

или в виде меняющего 4-значного числа:

1111 (исходное состояние)
 ...
 0000 (конечное состояние)

Постарайтесь снять все кольца за минимальное число операций. Ответы присылайте в редакцию.

Для решения вы можете использовать виртуальный вариант головоломки, опубликованный на сайте издательства «Информатика и образование» (доступ через страницу данного выпуска журнала).

Желающие могут также подготовить собственный виртуальный вариант. Для этого ознакомьтесь со статьей в рубрике «Microsoft Excel углубленно».

Методика разработки компьютерной программы, моделирующей головоломку «меледа», приведена в рубрике «Школа программирования».

Число-исполин

Для измерений в пределах солнечной системы астрономы считают единицей длины среднее расстояние от Земли до Солнца (149 500 000 км). Это так называемая “астрономическая единица”. Но для измерения расстояний до других солнц-звезд и туманностей эта мера слишком мала. Поэтому введены в употребление более крупные единицы:

- 1 астрономическая единица = 149 500 000 километров;
- 1 парсек = 206 265 астрономических единиц, или 30 800 000 000 000 километров;
- 1 килопарсек = 1000 парсеков;
- 1 мегапарсек = 1 000 000 парсеков.

Чтобы хоть отдаленно представить себе это число-исполин (мегапарсек), дадим одно сопоставление: тончайшая паутинная нить, протянутая от Москвы до Санкт-Петербурга, весила бы 10 граммов, от Земли до Луны — не более 8 килограммов, от Земли до Солнца — 3 тонны. Но протянутая на длину одного мегапарсека она весила бы...

Каков будет вес паутинной нити? Ответ получите, используя электронную таблицу. Значение, полученное с помощью этой программы, присылайте в редакцию.

Литература

1. На досуге. Сборник занимательных задач. — Ростов-на-Дону. Ростовское книжное изд-во, 1959.

Цепочки цифр

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

Ответ обоснуйте.

Цветные карандаши

У Пети три карандаша — зеленый, красный и черный. Назовите самый короткий и самый длинный карандаши, если известно, что:

- а) зеленый карандаш короче красного, а черный — короче зеленого;
- б) зеленый карандаш длиннее черного, а красный короче черного.

Задача предназначена для учащихся 1–7-х классов.

Рыбалка

Даниил, Максим и Лев ловили рыбу. Каждый из них поймал щуку или окуня. Известно, что если:

- Даниил и Максим поймали одинаковых рыб,
- Даниил и Лев — разных,
- Лев поймал щуку.

Какую рыбу поймал каждый из рыбаков?



Три обезьяны

На трех клетках в зоопарке висят таблички. На первой табличке написано, что в клетке сидит горилла, на второй — что в ней находится орангутанг или шимпанзе, а на третьей — что в клетке сидит шимпанзе. Известно, что надписи на всех табличках не соответствуют «содержимому» клеток. Какая обезьяна в какой клетке сидит?

Школьные учителя

В школе работают три учителя: Велиев, Симонов и Корниенко. Каждый из них преподает по два предмета, так что в расписании у них всего шесть предметов: математика, физика, химия, история, литература и английский языки. Корниенко — самый молодой из преподавателей. Учитель химии старше учителя истории. Все трое — учитель химии, учитель физики и Симонов — занимаются спортом. Когда между учителями литературы и английского языка возникает спор, Коршунов тоже принимает участие в споре. Симонов не преподает ни английский язык, ни математику. Кто какие предметы преподает?

Существует ли система счисления?

Существует ли такая система счисления, в которой одновременно соблюдаются два равенства:

- а) $3 + 4 = 10$ и $3 \times 4 = 15$;
- б) $3 + 2 = 5$ и $3 \times 2 = 11$.

Восстановить числа¹

Некоторое десятичное число X перевели в системы счисления с основаниями 16, 8, 4. Часть символов при записи была утеряна. Позиции утерянных символов обозначены символом “*”:

$$X = *7*_{16} = 5*6_8 = ***1*_4$$

Восстановите все числа и определите число X .

¹ Автор задачи — Е.А. Мирончик, учитель информатики лицея № 11, г. Новокузнецк Кемеровской обл.

Программируем модель «меледы»

В данной статье мы опишем методику разработки программы, моделирующей старинную русскую головоломку «меледа» (конструкция головоломки и правила обращения с ней описаны в статье в рубрике «Games.exe»). Как принято в нашем журнале, программу будем создавать на школьном алгоритмическом языке (система программирования КуМир). Русский синтаксис языка делает описание понятным, и вы сможете разработать соответствующую программу на языке программирования, который изучаете.

Примем, что головоломку в программе будем моделировать в виде схемы, в которой надетые кольца изображены над линией в виде буквы О, снятые — под ней, а номера колец показаны над схемой. Например, на схеме ниже можно снять кольцо 3 или надеть кольцо 1:

```

  4   3   2   1
      О   О
-----
  О           О

```

Состояние головоломки в любой момент будем хранить в виде массива с именем *меледа* из четырех элементов, значения которого могут быть равны 1 (соответствует одетому *i*-му кольцу) или 0 (соответствует снятому *i*-му кольцу). Поскольку нумерация элементов массива возрастающая, а на схеме выше кольца пронумерованы в обратном порядке, то получается, что следующая ситуация на головоломке:

снято	одето	снято	снято
4-е кольцо	3-е кольцо	2-е кольцо	1-е кольцо

будет моделировать массивом вида 0100. Это надо учесть при использовании массива *меледа*.

Общая структура программы:

1. Вывод заголовка и правил решения головоломки
2. Заполнение массива *меледа* начальными значениями
3. Вывод схемы, моделирующей головоломку в исходном состоянии
4. Цикл – многократное повторение «ходов» (снятия и надевания колец)
 - 4.1. Ввод пользователем номера кольца, состояние которого он хочет изменить
 - 4.2. Изменение состояния меледы.
 - 4.3. Вывод на экран нового состояния меледы
- Конец цикла
5. Вывод сообщения о решении головоломки.

Этап 1 простой, но объемный:

Вывод "ГОЛОВОЛОМКА =МЕЛЕДА="

Вывод **НС**

Вывод **НС**, "Программа моделирует старинную русскую головоломку =меледа="

Вывод **НС**, "Она состоит из замкнутой проволочной вилки, имеющей вид длинной"

Вывод **НС**, "шпильки для волос, воткнутой обоими свободными концами в рукоятку"

Вывод **НС**, "и нескольких колец, связанных между собой довольно сложным образом"

Вывод **НС**, "Задача состоит в том, чтобы снять всю систему колец с вилки "

Вывод **НС**

```

Вывод НС, "Правила, которые необходимо соблюдать при решении
           головоломки:"
Вывод НС, "- последнее (крайнее правое) кольцо можно всегда снять"
Вывод НС, "или надеть на вилку;"
Вывод НС, "- любое другое кольцо можно снять или надеть
           только тогда,"
Вывод НС, "когда кольцо непосредственно справа от него надето,"
Вывод НС, "а все остальные справа – сняты"
Вывод НС
Вывод НС, "Головоломка моделируется в программе в виде схемы "
Вывод НС, "На схеме надетые кольца изображены над линией,
           снятые – под ней "
Вывод НС, "Номера колец показаны над схемой "
Вывод НС, "Например, на схеме ниже можно снять кольцо 3
           или надеть кольцо 1"
Вывод НС, "4 3 2 1"
Вывод НС, " 0 0"
Вывод НС, "-----"
Вывод НС, "0          0"
Вывод НС
Вывод НС, "Попробуйте!"
Вывод НС, "Если готовы – введите любое число и нажмите клавишу Enter
"
Ввод ход | Условно. О переменной ход будет рассказано далее
Вывод НС

```

Заполнение массива *меледа* начальными значениями (этап 2) проводится так:

```

нц для i от 1 до 4
    меледа[i] := 1
кц

```

Так как вывод на экран схемы, моделирующей текущее состояние головоломки, будет проводиться многократно и соответствующий фрагмент программы будет несколько громоздким, то удобно для такого вывода разработать вспомогательную процедуру² с именем `вывод_меледы`. В ней будет четыре части:

- 1) вывод номеров колец;
- 2) вывод надетых колец;
- 3) вывод линии, имитирующей вилку (челнок);
- 4) вывод снятых колец.

Вторая и четвертая части зависят от состава массива *меледа*.

Прежде чем приводить процедуру, обратим внимание на то, что элементы массива надо рассматривать в обратном порядке (см. чуть выше).

Итак, указанная процедура:

```

алг вывод_меледы
нач цел i
    вывод НС
    | Номера колец
нц для i от 4 до 1 шаг -1
    вывод i, " "
кц
вывод НС

```

² С правилами оформления вспомогательных процедур и функций (см. далее о функции можно) в языке программирования, который вы изучаете, ознакомьтесь самостоятельно.

```

| Надетые кольца
нц для i от 4 до 1 шаг -1
  если меледа[i] = 1
    то
      вывод "О "
    иначе
      вывод " " | Пустые места
  все
кц
вывод нс
| Линия
нц для i от кол_колец до 1 шаг -1
  вывод "=="
кц
вывод нс
| Снятые кольца
нц для i от кол_колец до 1 шаг -1
  если меледа[i] = 0
    то
      вывод "О "
    иначе
      вывод " "
  все
кц
вывод нс
кон

```

На этапе 4 (см. выше общую структуру программы) остановимся подробно.

При вводе пользователем номера кольца, состояние которого он хочет изменить (п. 4.1 структуры программы) следует предусмотреть возможность того, что пользователь введет недопустимый номер (меньше 1 или больше 4). Поэтому соответствующий фрагмент надо оформить, например, с использованием оператора цикла с постусловием:

```

нц
  вывод нс, "Введите номер кольца, состояние которого
    Вы хотите изменить "
  ввод номер_кольца
  если номер_кольца < 1 или номер_кольца > 4 | Если номер ошибочный
    то
      вывод нс, "Неправильный номер! Попробуйте еще"
      вывод нс
      вывод меледы
    все
  кц_при номер_кольца >= 1 и номер_кольца <= 4

```

Однако понятно, что не при каждом «правильном» номере кольца можно указанное кольцо снять или надеть. Для этого еще должны соблюдаться правила обращения с головоломкой (см. их в начале статьи). Разработаем функцию `можно`, проверяющую, можно ли снять или надеть кольцо с заданным номером `номер_кольца`. Она будет возвращать результат логического типа:

```

алг лог можно (цел номер_кольца)
нач цел i
  если номер_кольца = 1
    то | Это кольцо можно всегда надеть или снять
      знач := да | Служебное слово знач означает значение функции
    иначе
      если номер_кольца = 2

```

```

то | Кольцо справа (с номером 1) должно быть надето
если меледа[1] = 1
    то
        знач := да
    все
иначе | Остальные кольца
    если меледа[номер_кольца - 1] = 0 | Если справа нет кольца
        то
            знач := нет
        иначе | Справа есть кольцо. Но этого еще ничего не значит
            знач := да | Сначала принимаем, что можно
            | Проверяем остальные кольца справа
            нц для i от номер_кольца - 2 до 1 шаг -1
                если меледа[i] = 1 | Если есть хотя бы одно кольцо
                    то
                        знач := нет
                    выход
                все
            кц
        все
    кц
все
кц
кон

```

Осталось правильно оформить оператор цикла повторяющихся ходов играющего (этап 4). Когда головоломка считается решенной? — Когда все кольца с вилки сняты. В нашей программе такой ситуации соответствует массив *меледа*, состоящий из одних нулей. Значит, в программе можем использовать, например, оператор цикла с предусловием:

```

нц пока <сумма всех элементов массива меледа не равна нулю>3
    вывод нс, "Введите номер кольца, ... "
    ...
кц

```

При небольшом размере массива меледа можно записать в операторе цикла такое условие:

```

нц пока меледа[1] + меледа[2] + меледа[3] + меледа[4] <> 0
    вывод нс, "Введите номер кольца, ... "
    ...
кц

```

Прежде чем представлять основную часть программы, обсудим важный вопрос. Так как массив *меледа* будет меняться в этой части программы, а вывод элементов этого массива проводится отдельно с помощью процедуры *вывод_меледы*, то массив *меледа* должен быть описан как глобальный (в программе на школьном алгоритмическом языке — до основной части программы).

```

цел таб меледа[1:4] | Глобальный массив
алг Основная_часть_программы
нач цел номер_кольца, ход
    вывод "ГОЛОВОЛОМКА =МЕЛЕДА="
    ... (см. выше)

```

³ Можно также использовать условие:

<количество нулей в массиве меледа не равна четырем>

или

<количество единиц в массиве меледа не равна нулю>

```

ввод ход    | Условно
вывод нс
вывод_меледы
| Этап 4
нц пока меледа[1] + меледа[2] + меледа[3] + меледа[4] <> 0
| Ввод номера кольца
нц
    вывод нс, "Введите номер кольца, ... "
    ...
кц при номер_кольца >= 1 и номер_кольца <= 4
| Проверяем, можно ли снять или одеть кольцо с указанным номером
если можно(номер_кольца) | Если можно
то
    | Снимаем или надеваем кольцо с номером номер кольца
    | Для этого меняем значение значения элемента массива меледа
    | с индексом номер кольца
    | Уточняем, что именно делатся меледа
если меледа[номер_кольца] = 1 | Кольцо было надето
то | Снимаем его
    меледа[номер_кольца] := 0
иначе | Кольцо было снято
    меледа[номер_кольца] := 1
все
иначе | Нельзя
    | Выводим, что именно нельзя
если меледа[номер_кольца] = 1 | Кольцо в данный момент надето
то
    вывод нс, "Это кольцо нельзя снять!"
иначе | Кольцо снято
    вывод нс, "Это кольцо нельзя надеть!"
все
все
вывод нс
вывод_меледы
кц
вывод нс, "Поздравляем!"
кон

```

Усовершенствуем программу — будем подсчитывать количество операций (ходов), сделанных пользователем при решении головоломки. Для этого используем уже представленную в программе переменную *ход*, значение которой будем увеличивать на 1 при каждом вводе пользователем номера кольца:

```

...
| Этап 4
ход := 0 | Обязательное начальное значение!
нц
| Ввод номера кольца
нц пока сумма <> 0
    ход := ход + 1
    вывод нс, "Введите номер кольца, ... "
    ...

```

а количество ходов выведем в заключительной части программы:

```

| Этап 5
вывод нс, "Поздравляем!"
вывод нс, "Вы решили задачу за ", ход, " ходов"
кон

```

Предлагаем читателям разработать соответствующую программу (на языке программирования, который изучаете) и прислать ее в редакцию⁴. Все приславшие правильную программу будут награждены дипломами.

Приведем некоторые комментарии к программе на языке Python.

В программах на этом языке не предусмотрен оператор цикла с постусловием. Поэтому при вводе и проверке правильности введенного пользователем номера кольца можно применить следующий фрагмент:

```
while True: # Такой оператор будет выполняться бесконечно
    print()
    nomer = int(input("Введите номер кольца,
                     состояние которого Вы хотите изменить "))
    if nomer < 1 or nomer > n:
        print("Неправильный номер! Попробуйте еще")
    else: # и только в этом случае (номер правильный)
        break # выходим из тела оператора цикла
```

По-особому в этом языке описываются глобальные переменные. Для проверки правильности введенного номера кольца с точки зрения правил работы с головоломкой можно создать функцию `test` с глобальной переменной `mozno`.

```
def test():
    global mozno
    if index == 0:
        mozno = True
    elif index == 1:
        if meleda[0] == 1:
            mozno = True
        else:
            mozno = False
    else:
        if meleda[index - 1] == 0:
            mozno = False
        else: # Справа надето
            mozno = True
            # Проверим остальные
            for i in range(index - 2, -1, -1):
                if meleda[i] == 1:
                    mozno = False
```

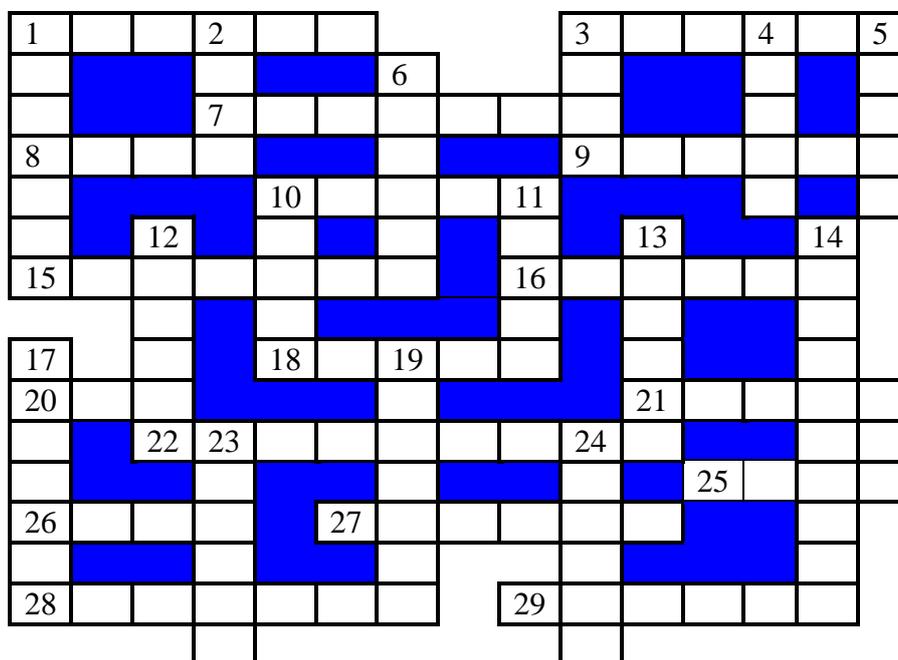
При этом, учитывая в языке Python элементы массивов (списков) нумеруются с нуля, при проверке соблюдения правил следует вместо переменной `nomer` использовать переменную `index`, которая и будет применяться в функции `test`:

```
# Этап 4
...
# Проверяем, можно ли снять или одеть кольцо с указанным номером
index = nomer - 1 # Уменьшаем номер
test()           # Вызываем функцию test
if mozno:      # Если можно
    ...
```

⁴ И, конечно, предложить ее для решения головоломки своим товарищам.

Кроссворд

Решите, пожалуйста, кроссворд.

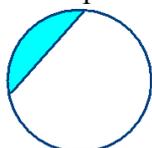


По горизонтали

1. Физическая величина, имеющая только значение и не имеющая направления.
3. Английский математик, в честь которого названа схема перевода чисел из одной системы счисления в другую.
7. Так называли гибкий магнитный диск.
8. Приставка в единице измерения количества информации, равной 1024 байта.
9. Совокупность правил записи данных, записанных в компьютерном файле.
10. Один из первых языков программирования высокого уровня.
15. Жительница российской республики, столица которой — город Кызыл.
16. Один из необходимых элементов блок-схемы алгоритма.
18. Знак, обозначающий число.
20. Цифра восьмеричной системы счисления.
21. Минимальное количество, на которое может изменяться дискретная величина.
22. Круговая или столбчатая ...
25. Почти синоним слова *единица*.
26. Единица измерения количества информации.
27. Принадлежность для печати на принтере.
28. Передвижная опора для оптических приборов, осветительного оборудования и т.п. с тремя «ножками».
29. Условный знак или физический процесс, передающие некоторую информацию.

По вертикали

1. Закрашенная часть рисунка.

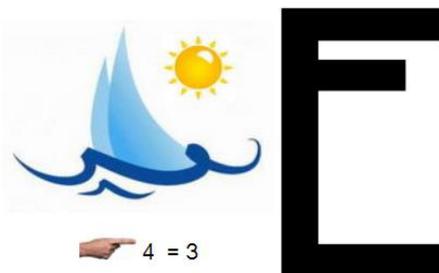


2. Марка отечественных автомобилей.
3. Конечное число точек на плоскости, соединенных отрезками кривых линий.
4. Термин для обозначения некоторой характеристики — допустимого значения, усреднённой или среднестатистической величины.
5. Совокупность точек графического изображения.
6. Знак препинания, используемый в формулах.
10. Один из необходимых элементов блок-схемы алгоритма (как бы антоним к слову 16 по горизонтали).
11. Элемент матричного принтера, а также перфо... .
12. Дневнегреческий математик, автор алгоритма нахождения наибольшего общего делителя двух натуральных чисел.
13. Элемент электронной таблицы.
14. Текст, повторяющийся на каждой странице документа в его верхней или нижней части.
17. Характеристика файла, синоним слова *поле* (в базах данных), а также необходимое, существенное, неотъемлемое свойство предмета или явления.
19. В электронной таблице — запись, начинающаяся символом “=”.
23. Значение переменной величины или константы логического типа (русский вариант).
24. Предмет, который создает вокруг себя поле, в котором другие предметы подвергаются действию определенной силы.

Семь ребусов

Решите, пожалуйста, ребусы. Ответы и комментарии к найденным словам присылайте в редакцию. Можно решать не все ребусы.

Ребус № 1



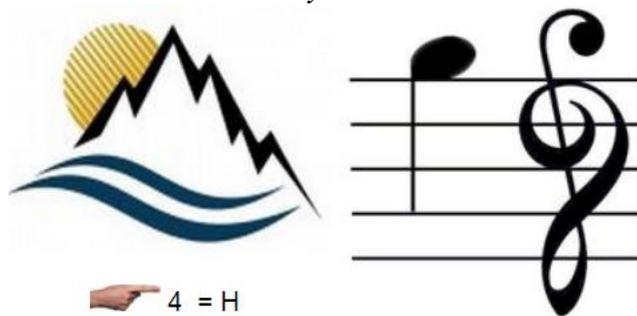
Ребус № 2



Ребус № 3



Ребус № 4



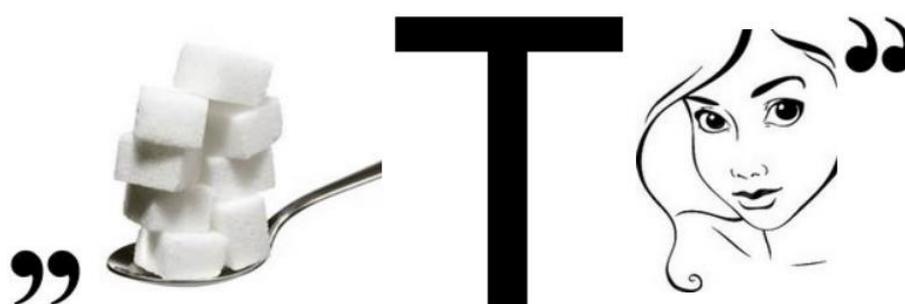
Ребус № 5



Ребус № 6



Ребус № 7



Составить слово

Н.А. Владимирова,

учитель информатики гимназии № 2 г. Заозерный Красноярского края

Из букв данных слов, переставив их, необходимо составить термин⁵, связанный с информатикой и компьютерами. Например, из слов *морс* и *порт* можно получить слово *просмотр*.

А

1. Сад, ре.
2. Ре, пат, ад.
3. Хек, аут, тир, ар.

Б

1. Бес, кий.
2. Клон, бот.
3. Арбуз, ре.

В

1. Стек, вар.
2. Сиг, дело, вина.
3. Дело, ре, витамин.

Г

1. Пир, стек, тег.
2. Факир, га.
3. Ель, сто, три, порог, фа.

Д

1. Ад, кiset.
2. Стан, дело, ля, тир.
3. Лига, до.

Е/Ё

1. Сеть, ком.
2. Шов, ре (следует получить фамилию).
3. Лес, и.

Ж

1. Кий, жест.
2. Кость, жест.
3. Чиж, едок, кристалл, кий, си

З

1. Го, коза, вол.
2. Ваза, лик.
3. Ваза, скат.

И

1. Пир, том.
2. Мир, нота, икра.
3. Лот, пень, и, лис.

К

1. Пан, кок.
2. Кит, вал, аура.
3. Титул, он, кол.

Л

1. Тис, лак.
2. Иней, лак.
3. Лак, иго.

М

1. Сом, рак.
2. Комод, рана, мак.
3. Ель, дом.

Н

1. Овраг, тина.
2. Сойка, рант
3. Ель, тон, си.

О

1. Робот, абак.
2. Рот, опера.
3. Точка, си.

П

1. Ар, плита.
2. Магма, тор, пик.
3. Кол, топор.

Р

1. Ор, бот.
2. Сет, рация, риза.
3. Катер, род.

С

1. Крен, ас.
2. Вор, рак, сито.
3. Стан, иск, си.

Т

1. Ре, тайм.
2. Тура, стек.
3. Мел, тиран.

⁵ Или фамилию ученого — Прим. ред.

У

1. Ковка, па, у.
2. Ре, лупа, иена.
3. Сукно, вата

Ф

1. Фарт, ом.
2. Мул, фора.
3. Франт, ор.

Х

1. Ар, хек.
2. Хаки, актер, тир, ас.
3. Тире, холл (следует получить фамилию).

Ц

1. Сон, тост, цель.
2. Лось, цент, сто.
3. Рант, цельный.

Ч

1. Каша, череп.
2. Енот, час.
3. Сота, чат.

Ш

1. Рота, шифр.
2. Док, штрих.
3. Ре, штык.

Щ

1. Акт, щель.
2. Кнут, щель.
3. Лещ, чок.

Э

1. Рак, эн.
2. Ля, эму, рот.
3. Эхо, ноль, корт.

Ю

1. Ре, юз.
2. ВИКА, НЮ (название ЭВМ).
3. ИКС, НЮ.

Я

1. Ор, яд.
2. Кость, яр.
3. Рык, ля.

От редакции

1. Прокомментируйте найденные термины и фамилии ученых.
2. Ответы присылайте в редакцию (можно выполнять не все задания).

Три задачи со спичками⁶**Задача 1 “Правильные ли примеры?”**

Ученик изобразил при помощи спичек такие непонятные примеры:

$$5 + 2 + 5 = 21$$

$$2 \times 8 = 91$$

Но, как ни странно, учитель похвалил ученика и сказал, что все примеры правильные. Как такое могло произойти?

⁶ Задачи, предназначены для учащихся начальной школы и учеников 5–7-х классов.

Задача 2 “Тройки и пятерки”

Переставьте в каждом из двух примеров по две спички так, чтобы оба данных равенства оказались верными.

$$\begin{array}{l} \text{III} \times \text{III} \times \text{III} = 5 \\ \text{III} \times \text{III} \times \text{III} = 55 \end{array}$$

Задача 3 “Снова три тройки”

Переставьте две спички так, чтобы данное равенство оказалось верным.

$$\text{III} \times \text{III} \times \text{III} = 75$$

Найдите два различных способа.

Задачи с песочными часами. Часть 3

Есть двое песочных часов. Для приготовления эликсира бессмертия его надо *непрерывно* готовить ровно 4 минут. Как это сделать за минимально возможное число операций (установок часов и их переворачиваний)? Время, затрачиваемое на переворачивание часов, не учитывать. Задачу решите для трех случаев, приведенных в таблице.

№ пп	Первые часы	Вторые часы	Время приготовления эликсира 4
1	3 мин	5 мин	4 мин
2	3 мин	8 мин	1 мин
3	3 мин	8 мин	4 мин

Решение, пожалуйста, оформите в виде схемы:

Первые часы (... мин)	1	3	4	5	
Вторые часы (... мин)	2			Н	К

в которой числами обозначены номера установок часов и их переворачиваний, а буквами Н и К — соответственно, моменты начала и конца приготовления эликсира. По возможности, соблюдайте пропорции в изображении отрезков времени. Можно также описать алгоритм решения задач словесно.

Два “географических” числовых ребуса

Решите, пожалуйста, два числовых ребуса. Как обычно в таких головоломках, одинаковыми буквами зашифрованы одинаковые цифры, разными буквами — разные цифры.

1. ГАВ = АН^А.

2. СССР = Р^Ф.

Ответы с обоснованием присылайте в редакцию.



Классические sudoku

1) простая:

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

2) сложная:

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

Судoku с суммами

Здесь надо решить классический вариант sudoku плюс дополнительное условие: цифры в углах пунктирных блоков равны сумме чисел в блоке.

9			14			4			9			10
9	4		13			16			8			
			16	8			7					18
14				5	13		11					16
3	14						9					10
	7			11	14							
20	6				14		11	15				7
	17											11
	12					13						7

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

Четыре вопроса

1. Какую настойку Плиний Старший считал эликсиром молодости и средством от плохого дыхания?
2. Какая швея, согласно легенде, сшила первый американский флаг?
3. В каком месяце Александром Сергеевичем Пушкиным овладевала "ужасная тоска"?
4. Кто, когда и в каком городе провел первую хирургическую операцию под наркозом?

Модель головоломки «меледа» с цифрами

В данной статье мы опишем методику оформления листа программы Microsoft Excel, моделирующего головоломку «Меледа», о которой рассказано в рубрике «Games.exe». Вместо колец мы будем использовать цифры 1 (соответствует надетому кольцу) и 0 (соответствует снятому кольцу). Примем, что количество колец равно 4. Это значит, что задачу можно сформулировать так: «Дано число 1111. Каждую его единицу можно заменить на 0 и наоборот. Но при этом обязательно соблюдаться следующие правила.

- последняя цифра может быть изменена всегда
 - предпоследняя — только если последняя цифра — 1
 - вторая слева — если справа от нее стоит 1 и последняя цифра 0.
 - первая слева — если справа от нее стоит 1, а потом 00.
- Следует получить число 0000».

Итак, начнем.

Верхнюю левую часть листа оформим в виде

	A	B	C	D	E
1	Заменить все единицы на нули				
2	1	1	1	1	
3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
4	Включите радиокнопку для изменения цифры				

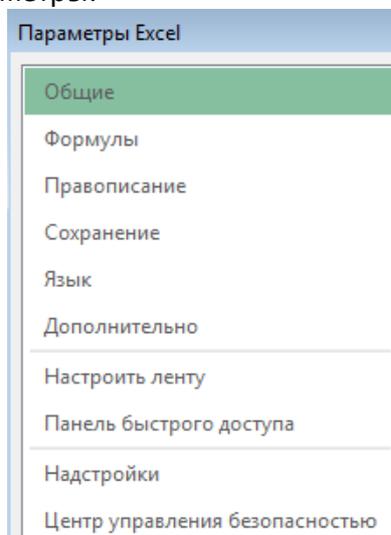
Рис. 1

В строке 3 размещены так называемые «радиокнопки» — элементы управления, которые, конечно, встречались вам в различных программах. Их особенность в том, что при включении той или иной кнопки остальные кнопки выключаются. Именно с помощью радиокнопок будут меняться значения цифр.

Для размещения элементов управления на листе нужно предусмотреть в главном меню программы Microsoft Excel пункт (вкладку) РАЗРАБОТЧИК:



Для этого нужно в пункте меню **Файл** выбрать подпункт **Параметры** — появится окно **Параметры**:



В нем надо выбрать строчку Настроить ленту, после чего в появившемся окне справа поставить «галочку» в строке Разработчик (см. рис. 2) и щелкнуть на ОК. В результате в главном меню появится пункт (вкладка) Разработчик.

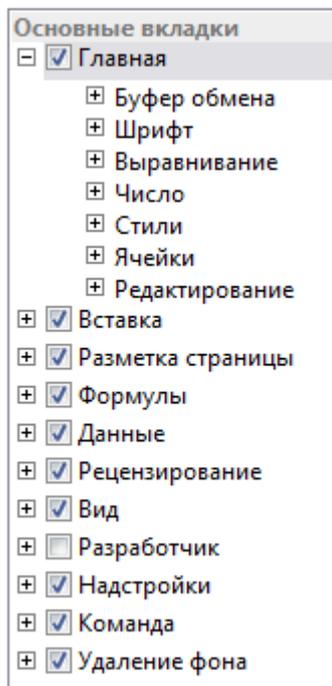
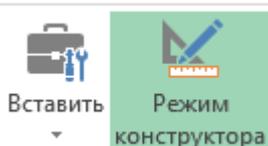


Рис. 2

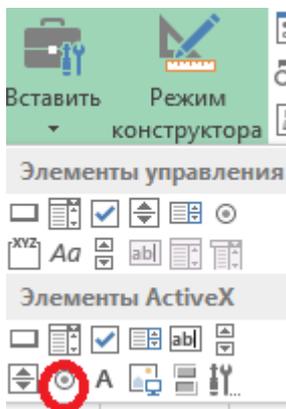
Как же разместить на листе радиокнопки? Прежде всего, следует включить так называемый «режим конструктора». Для этого на панели инструментов вкладки Разработчик существует специальный инструмент с таким именем:



После щелчка на нем следует щелкнуть на инструменте **Вставить**:



и выбрать элемент, обведенный красным цветом:



Указатель мыши примет форму знака +. Переместите его в ячейку A3 и, не отпуская левую кнопку мыши, нарисуйте в ячейке прямоугольник небольшого размера, после чего отпустите кнопку. В ячейке появится созданный элемент управления:

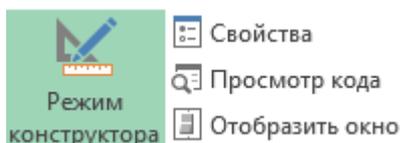
	A	B	C	D	E
1	Заменить все единицы на нули				
2	1	1	1	1	
3	<input type="radio"/> Opti				

Надпись справа от радиокнопки следует удалить. Для этого надо вызвать окно свойств этого элемента управления. Это делается так:

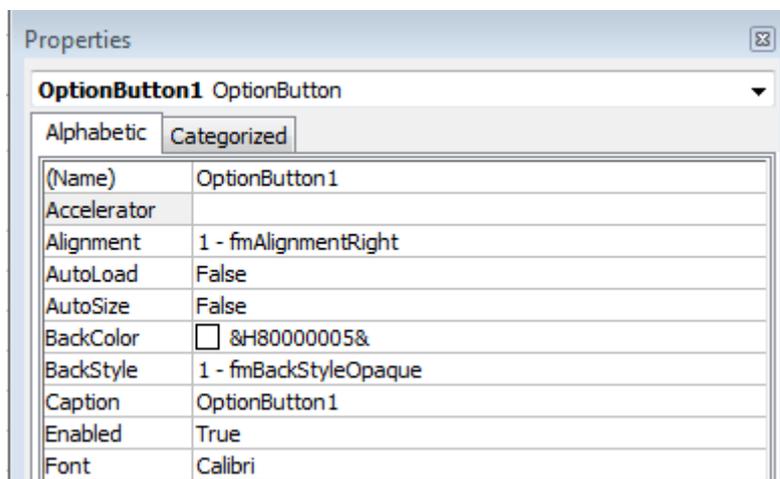
- 1) включить режим конструктора (если он не был включен);
- 2) выделить радиокнопку:

	A	B	C	D	E
1	Заменить все единицы на нули				
2	1	1	1	1	
3	<input checked="" type="radio"/> Opti				
4					

- 3) щелкнуть на инструменте Свойства справа от Режим конструктора:



В результате появится окно свойств:



В свойстве `Caption` нужно очистить (удалить) значение `OptionButton1`.

После полного оформления первой радиокнопки удобно создать еще 3 ее копии и разместить в соответствующих ячейках.

Имена созданных кнопок будут `OptionButton1`, `OptionButton2`, `OptionButton3` и `OptionButton4`.

К созданным кнопкам нужно «привязать» так называемые «макросы» — программы, который будут выполнять некоторые действия при включении той или иной кнопки. Эти программы пишутся на языке программирования VBA (Visual Basic for Application).

Как изменить значение в той или иной ячейке с цифрами с помощью программы-макроса? Например, в ячейке D2 это можно сделать так:

```

' Если текущее значение в ячейке Range("D2").Value равно 1
If Range("D2").Value = 1 Then
    ' Меняем его на 0
    Range("D2").Value = 0
Else 'иначе (текущее значение в ячейке Range("D2").Value равно 0)
    ' Меняем его на 1
    Range("D2").Value = 1
End If

```

Эту конструкцию (полный условный оператор) можно заменить на более компактную⁷:

```
Range("D2").Value = 1 - Range("D2").Value
```

После включения радиокнопок и изменения цифр будем выключать их. Для этого в макросе следует предусмотреть такую команду (например, для радиокнопки с именем OptionButton3):

```
OptionButton3.Value = False
```

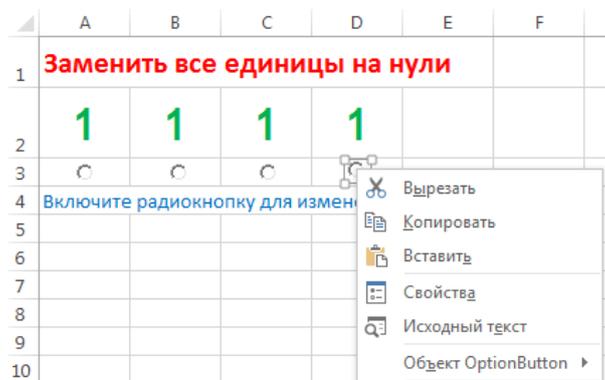
Это значит, что для крайней правой радиокнопки, которую, согласно правилам, можно включать всегда, макрос будет таким:

```

Private Sub OptionButton4_Click()
    Range("D2").Value = 1 - Range("D2").Value
    OptionButton4.Value = False
End Sub

```

Но как «привязать» этот макрос к кнопке? Для этого при включенном режиме конструктора (см. выше) нужно щелкнуть правой кнопкой мыши на выбранной радиокнопке и в появившемся контекстном меню:



выбрать пункт Исходный текст. Появится окно редактора программы с заголовком макроса и его завершающей конструкцией:

```

Private Sub OptionButton4_Click()

End Sub

```

Именно между этими двумя строками записывается текст макроса, который для последней радиокнопки мы уже описали:

```

Private Sub OptionButton4_Click()
    Range("D2").Value = 1 - Range("D2").Value
    OptionButton4.Value = False
End Sub

```

⁷ Убедитесь, что с ней будут происходить изменения значения так же, как и с полным условным оператором.

Для других радиокнопок макросы в соответствии с правилами решения головоломки будут такими:

```
Private Sub OptionButton3_Click()  
If Range("D2").Value = 1 Then  
    ' Меняем вторую справа цифру  
    Range("C2").Value = 1 - Range("C2").Value  
Else ' По правилам эту цифру изменять нельзя - выводим сообщение  
    MsgBox ("Ошибка! Эту цифру изменять нельзя!")  
End If  
OptionButton3.Value = False  
End Sub  
  
Private Sub OptionButton2_Click()  
If Range("C2").Value = 1 And Range("D2").Value = 0 Then  
    Range("B2").Value = 1 - Range("B2").Value  
Else  
    MsgBox ("Ошибка! Эту цифру изменять нельзя!")  
End If  
OptionButton2.Value = False  
End Sub  
  
Private Sub OptionButton1_Click()  
If Range("B2").Value = 1 And Range("C2").Value = 0 And Range("D2").Value = 0  
Then  
    Range("A2").Value = 1 - Range("A2").Value  
Else  
    MsgBox ("Ошибка! Эту цифру изменять нельзя!")  
End If  
OptionButton1.Value = False  
End Sub
```

Сделайте все, что описано выше, и проверьте, правильно ли происходит изменение значений в ячейках с цифрами и выводятся ли сообщения об ошибке только в случае нарушения правил работы с головоломкой. Для проверки (и для работы с «программой») следует выключать режим конструктора (с помощью путем повторного щелчка на этом инструменте).

Если все правильно, то осталось научиться проверять, не решена ли пользователем задача. Тот факт, что задача решена, можно зафиксировать, проверив такое условие:

```
Range("A2").Value + Range("B2").Value + Range("C2").Value +  
Range("D2").Value = 0
```

Соответствующую проверку можно проводить в *каждом* макросе (в его конце):

```
...  
If Range("A2").Value + Range("B2").Value +  
    Range("C2").Value + Range("D2").Value = 0 Then  
    MsgBox ("Поздравляем! Вы решили задачу!")  
End If  
End Sub
```

Но можно это сделать только в одном макросе. В каком? Для ответа на вопрос, подумайте, после изменения какой цифры задача станет решенной, и запишите приведенный фрагмент проверки именно в соответствующем макросе.

Чтобы записать его, следует перейти в окно редактора (где до этого были написаны 4 макроса).

В заключение заметим следующее:

- 1) «играть» можно только при отключенном режиме конструктора;
- 2) для повторного начала игры можно записать цифры 1 в ячейки первой строки

Внимание! Конкурс!

Конкурс № 36

В качестве задания этого конкурса предлагаем выполнить задания для самостоятельной работы, предложенные в статье «Составить слово» в рубрике «“Ломаем” голову».

Ответы направьте в редакцию по адресу mirinfo@infojournal.ru. Пожалуйста, укажите в письме свои фамилию и имя, населённый пункт и номер школы, а также фамилию, имя и отчество учителя информатики. Срок представления ответов — 28 февраля 2021 г.

Дипломами будут награждены также читатели:

- 1) представившие программу (на изучаемом языке программирования), рассмотренную в рубрике «Школа программирования»;
- 2) представившие документ программы Microsoft Excel с листом, моделирующим головоломку «Меледа» (см. соответствующую рубрику).

Уважаемые читатели! Присылайте ответы на задания наших конкурсов и решения задач и головоломок, если по каким-то причинам вы подготовите их позже установленного срока — мы обязательно рассмотрим и их.

Ответы и решения к задачам, опубликованным в ноябрьском выпуске журнала «Мир информатики», и списки читателей, приславших правильные ответы, будут опубликованы в следующем выпуске.

Книжная полка

Книга для подготовки к ЕГЭ по информатике

Издательство ДМК Пресс предлагает книгу Д.М. Златопольского «Подготовка к ЕГЭ по информатике в компьютерной форме».

<p>Златопольский Д. М.</p> <p>Подготовка к ЕГЭ по информатике в компьютерной форме</p>  <p>ДМК ПРесс</p>	<p>Книга предназначена для самостоятельной подготовки учащихся к единому государственному экзамену по информатике и ИКТ, который начиная с 2021 года будет проходить в компьютерной форме. Согласно демонстрационному варианту ЕГЭ 2021 года, в содержании экзамена будет существенно увеличено количество заданий, связанных с алгоритмизацией и программированием. Таким задачам в книге уделяется особое внимание. Обсуждаются и методики выполнения ряда других заданий, представленных в демонстрационном варианте.</p> <p>В приложениях приведены вспомогательные материалы, связанные с заданиями ЕГЭ.</p> <p>Кроме учащихся старших классов, готовящихся к сдаче экзамена, книгу могут использовать преподаватели информатики, а также студенты и школьники, желающие углубить общие знания по информатике и ИКТ вне связи с ЕГЭ.</p>
--	---

Подробнее — см. <https://dmkpress.com/catalog/computer/handbooks/978-5-97060-896-8/>

Книга «400 вопросов по информатике на логику и смекалку»

В издательстве ДМК Пресс вышла книга Д.М. Златопольского «400 вопросов по информатике на логику и смекалку» (<https://dmkpress.com/catalog/computer/handbooks/978-5-97060-898-2/>).



В книге приведены более 400 вопросов по информатике, которые можно предложить учащимся при проведении классных и школьных конкурсов «Что? Где? Когда?» и «Брейн-ринг» и других внеклассных мероприятий. Ряд вопросов может быть использован и на уроках. Вопросы охватывают многие темы школьного курса информатики, имеют разный уровень сложности. В приложениях представлены другие вопросы и задания, связанные с интеллектуальными соревнованиями. Ко всем вопросам даны ответы.

Книга предназначена педагогам, но может быть полезна также родителям учащихся и, конечно, самим учащимся, которые могут проверить свои знания, логику и смекалку.

По издательской цене (гораздо дешевле, чем в магазинах, даже с учетом стоимости доставки) её можно приобрести через сайт издательства <https://dmkpress.com/>

Сборник задач по программированию

Если вы хотите стать программистом, то должны знать, что одним из условий этого является большой опыт решение задач по программированию. 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, что уже решенная задача или аналогичная ей встретится вам в будущем. В приобретении такого опыта вам поможет книга «1400 задач по программированию», выпущенная издательством ДМК Пресс.



В книге приведено более 1400 задач по программированию. Задачи имеют разный уровень сложности и охватывают большой круг тем. Они не привязаны к конкретному языку программирования и могут быть решены в программе на любом языке (Паскаль, Python, Си, школьном алгоритмическом языке и т. д.).

Издание будет полезно как читателям, изучающим программирование самостоятельно, так и преподавателям школ, колледжей и вузов. В последнем случае задачи могут быть использованы при объяснении нового материала, для организации самостоятельной работы учащихся и проведения контрольных мероприятий. Ряд задач предназначен для внутришкольных олимпиад, для использования в кружках по программированию и т. п.