

«Мир информатики»

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 106, декабрь 2025 г.

История информатики

День информатики

4 декабря в России отмечают День информатики — в этот день в 1948 году началась электронная эра отечественных вычислительных машин.

4 декабря 1948 года в Государственном патентном бюро СССР было зарегистрировано авторское свидетельство на изобретение под названием «Автоматическая цифровая вычислительная машина». Это событие, упомянутое в архивных материалах Академии наук СССР, стало отправной точкой развития отечественной информатики и вошло в календарь как отраслевой праздник.

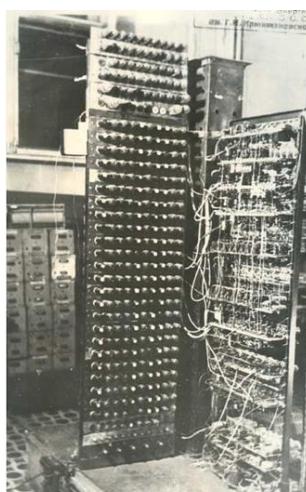


День информатики — это напоминание о моменте, когда страна сделала первые шаги в сторону цифровых технологий. Несмотря на послевоенные ограничения, советские ученые сумели сформировать новую научную область, которая позже дала стране десятки вычислительных школ, инженерных центров и собственную линию ЭВМ.

Регистрация авторского свидетельства 4 декабря 1948 года официально закрепила рождение советской цифровой вычислительной техники. До этого момента в СССР не существовало ни одной признанной разработки, связанной с цифровыми ЭВМ.

Первая советская ЭВМ «М-1» как точка отсчета

На основе указанного авторского свидетельства была разработана «Автоматическая цифровая вычислительная машина М-1» (АЦВМ М-1)



ЭВМ М-1

00000045	5	5177			
76000000	1	74040000	5	74000000	1
72000000	1	64440000	5	70000000	1
66000000	1	55440000	5	64000000	1
62000000	1	47040000	5	50000000	1
56000000	1	41040000	5	54000000	1
52000000	1	33440000	5	50000000	1
46000000	1	26440000	5	44000000	1
42000000	1	22040000	5	40000000	1
36000000	1	16040000	5	34000000	1
32000000	1	12440000	5	30000000	1
26000000	1	07440000	5	24000000	1
22000000	1	05040000	5	20000000	1
16000000	1	03040000	5	14000000	1
12000000	1	01440000	5	10000000	1
06000000	1	00440000	5	04000000	1
02000000	1	00040000	5	00000000	1
02000000	5	00040000	5	04000000	5
06000000	5	00440000	5	10000000	5
12000000	5	01440000	5	14000000	5
15000000	5	03040000	5	20000000	5
22000000	5	05040000	5	24000000	5
26000000	5	07440000	5	30000000	5
32000000	5	12440000	5	34000035	2
36000000	5	16040000	5	40000000	5
42000000	5	22040000	5	44000000	5
46000000	5	26440000	5	50000000	5
52000000	5	33440000	5	54000000	5
56000000	5	41040000	5	60000000	5
62000000	5	47040000	5	64000000	5
66000000	5	55440000	5	70000000	5
72000000	5	64440000	5	74000000	5
76000000	5	74040000	5	70200000	5

Первая программа, выполненная на М-1

Главной особенностью «М-1» стала электронная память на лучевых трубках, а также использование полупроводниковых элементов, которые заметно повысили надежность. Для своего времени это было смелое инженерное решение. Оно позволило отказаться от медленных механических систем, которые тогда еще доминировали в мире.

Первая серия расчетов прошла в начале 1950-х годов. «М-1» работала в исследовательских задачах, связанных с энергетикой, физическими расчетами и первыми моделями автоматизации.

Кто создал «М-1»: Исаак Брук и Башир Рамеев



Исаак Брук стал одним из первых ученых, увидевших потенциал электронных машин для сложных инженерных задач и научных вычислений.

Исаак Брук был специалистом в области электросистем и одним из первых ученых, увидевших потенциал электронных машин для инженерных расчетов. После войны он организовал лабораторию, в которой и зародилась идея советской ЭВМ.



Башир Рамеев определил архитектуру М-1 и позже создал ключевые серии ЭВМ, сформировавшие основу отечественной вычислительной школы.

Башир Рамеев — инженер, который пришел в вычислительную технику из радиосвязи. Его практические навыки в схемотехнике и электронике стали основой для разработки архитектуры «М-1». Позже он стал автором проектов «Стрела» и «Урал», определивших развитие вычислительной техники на десятилетия вперед.

Работа Брука и Рамеева стала примером того, как синтез академического знания и инженерной смелости может создать технологию нового поколения.

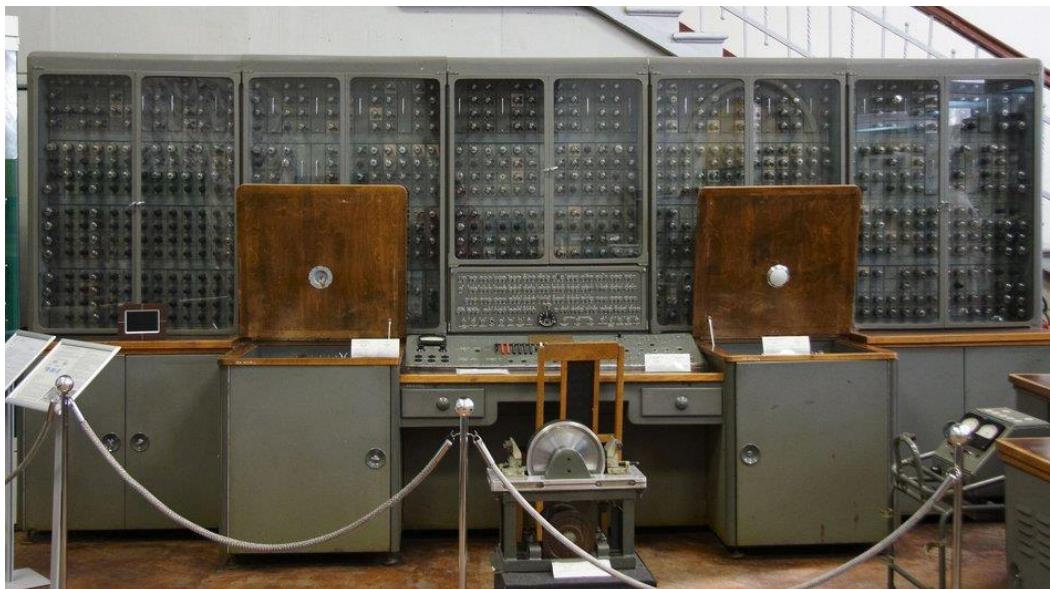
История ЭВМ в России: от «Стрелы» до современных процессоров

Создание «М-1» стало отправной точкой. Уже в 1950-х годах появились первые серийные машины — «Стрела», а затем целая линейка «Урал», широко применявшаяся в научных центрах, оборонных задачах и на космодроме Байконур. С конца 1960-х в СССР начали выпускать семейство «Минск», которое стало самой массовой серией советских ЭВМ.

В 1970-х появился проект ЕС ЭВМ — серия машин, совместимая с международными стандартами. Он обеспечил переход к комплексным системам обработки данных и стал основой вычислительных центров страны.



ЭВМ «Стрела» стала первой советской серийной ЭВМ и открыла путь к промышленному применению электронных вычислений.



Серия ЭВМ «Урал» превратила вычислительную технику в массовый инструмент промышленности, науки и космических программ



ЭВМ БЭСМ-6 (1968 г.) была самой быстродействующей в Европе (1 млн операций в секунду)

Сегодня развитие продолжают российские компании, создающие процессоры и вычислительные системы — среди них «Байкал Электроникс» и производители серверного оборудования. Активно развивается архитектура RISC-V, которая дает возможность формировать независимую элементную базу для будущих вычислительных платформ.

От «М-1» к современным технологиям

День информатики — это напоминание о том, что современная цифровая среда выросла из инженерных проектов конца 1940-х годов. От первых электронных схем до современных процессоров путь занял почти восемь десятилетий. Но ключевая идея осталась той же: вычислительная машина должна помогать человеку решать сложные задачи быстрее и точнее.

И именно 4 декабря стало символом того, как в СССР началась электронная эпоха, продолжение которой мы наблюдаем сегодня — от исследовательских вычислений до искусственного интеллекта.

По материалам сайта Наук Mail. Автор Владимир Барышев

Основы программирования на языке Python¹

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

5.2. Один, но не обязательный вариант действий

Пример. Даны три целых числа, среди которых есть отрицательные. Вывести на экран отрицательные числа на одной строке.

Используем в программе три переменные – a, b и c.

Начальная часть программы решения задачи:

```
print('Введите 3 целых числа (как минимум одно - отрицательное)')
a = int(input())
b = int(input())
c = int(input())
print('Среди них отрицательные:')
```

Далее – в каком случае выводить значение первого числа a? Только если соблюдается условие $a < 0$. Аналогично и для двух других чисел. В таких случаях (когда какие-то действия в программе выполняются не всегда, а только при определенном условии) используется инструкция if в следующей форме:

```
if <условие>:
    <действия (серия инструкций)>
```

Такой вариант инструкции if называют «неполный вариант».

Применительно к рассмотренному примеру завершающая часть программы решения задачи оформляется так:

```
if a < 0:
    print(a, ' ', , end = '')
if b < 0:
    print(b, ' ', , end = '')
if c < 0:
    print(c)
```

Схема работы неполного варианта инструкции if показана на рис. 5.5²:

¹ См. также предыдущие выпуски интернет-журнала «Мир информатики».

² Приведенную схему называют «обход».

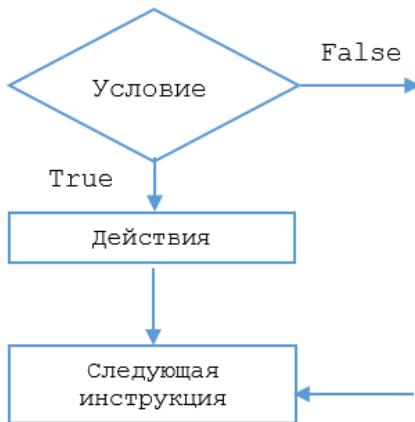


Рис. 5.5

Еще раз обратим внимание на то, что нет альтернативных вариантов действий, как в случае, рассмотренном в пункте 5.1, а есть единственно возможный, но не обязательный, вариант.

Контрольные вопросы

В каких случаях в программе используется неполный вариант инструкции if? Как он оформляется? Как он «работает»? (Что происходит при его выполнении?) Нарисуйте графическую схему выполнения.

Задачи для разработки программ

1. Даны три целых числа. Вывести на экран те из них, которые являются четными.
2. Дано вещественное число. Вывести на экран его абсолютную величину (условно принимая, что соответствующей стандартной функции Python нет). Полный вариант инструкции if не использовать.
3. Даны два различных вещественных числа. Определить наибольшее и наименьшее из них, не используя полный вариант инструкции if, а применив:
 - а) два неполных варианта инструкции;
 - б) один неполный вариант инструкции.
4. Даны три вещественных числа. Вывести на экран те из них, которые принадлежат интервалу [1.6, 3.8].
5. Даны три числа a, b, c. Определить, сколько из них положительных.

Решение

Можно после ввода исходных данных

```

print('Введите 3 числа')
a = float(input())
b = float(input())
c = float(input())

```

рассмотреть все возможные варианты, используя неполные варианты инструкции if:

```

if a <=0 and b <= 0 and c <= 0:
    k = 0
if (a > 0 and b <= 0 and c <= 0 or b > 0 and a <= 0 and c <= 0
       or c > 0 and a <= 0 and b <= 0):
    k = 1
if (a > 0 and b > 0 and c <= 0 or a > 0 and c > 0 and b <= 0
       or b > 0 and c > 0 and a <= 0):
    k = 2
if a > 0 and b > 0 and c > 0:
    k = 3
#Вывод ответа

```

...

Обратим внимание на то, что, поскольку логическая операция с оператором `and` выполняется раньше, чем операция с оператором `or`, то скобки в сложном условии записывать не надо (имеющиеся в программе скобки позволяют записать длинное сложное условие на двух строках).

6. Даны шесть целых чисел a, b, c, d, e, f . Определить, сколько из них положительных.

Решение

Если решать задачу так, как предыдущую, то программа получится достаточно громоздкой (а если бы заданных чисел было 10?). Можно сократить размер программы, учитывая следующие рассуждения.

Во-первых, будем вводить заданные числа по одному и сразу будем проверять, является ли очередное число положительным (в предыдущей программе проверка проводилась после ввода всех трех чисел).

Во-вторых, используем в программе переменную k , которая будет хранить³ количество положительных чисел среди уже рассмотренных заданных чисел.

Составим таблицу:

Число	Значение	k
a	7	1
b	-2	1
...		k
n	6	?
...		

Как определить количество положительных чисел после ввода некоторого очередного положительного числа n , если информацию о таком количестве среди уже рассмотренных чисел хранит величина k ? Ответ – надо к «старому» значению количества k прибавить 1:

$$k = k + 1$$

Напомним, что в программировании такая запись возможна.

При сделанных рассуждениях вся программа решения задачи может быть оформлена так:

```
k = 0 #Начальное значение переменной k
a = int(input('Введите значение числа a '))
if a > 0:
    k = k + 1      #Применен неполный вариант инструкции if
b = int(input('Введите значение числа b '))
if b > 0:
    k = k + 1
c = int(input('Введите значение числа c '))
if c > 0:
    k = k + 1
d = int(input('Введите значение числа d '))
if d > 0:
    k = k + 1
e = int(input('Введите значение числа e '))
if e > 0:
    k = k + 1
f = int(input('Введите значение числа f '))
if f > 0:
    k = k + 1
#Вывод ответа
```

³ Напомним, что переменные используются в программах для хранения информации.

...

Согласитесь, что приведенный вариант программы не только имеет меньший размер, чем вариант с полным перебором значений, но и понятнее.

Переменные, аналогичные переменной k , которые предназначены для подсчета количества некоторых значений, называют «счетчиками».

7. Даны три целых числа a, b, c . Определить сумму положительных из них.

8. Даны шесть целых чисел a, b, c, d, e, f . Определить сумму положительных из них.

Комментарий к решению

Если предыдущую задачу можно решить полным перебором, как это делалось при решении задачи 5, то в данном случае это нерационально (программа получится достаточно громоздкой).

Лучше использовать методику, аналогичную использованной при решении задачи 6.

Используем в программе переменную sum , которая будет хранить сумму положительных чисел среди уже рассмотренных заданных чисел.

Составим таблицу:

Число	Значение	sum
a	7	7
b	2	9
...		sum
n	6	?
...		

Как определить сумму положительных чисел после ввода некоторого очередного положительного числа n , если информацию о такой сумме среди уже рассмотренных чисел хранит величина sum ? Ответ – надо к «старому» значению суммы sum прибавить n :

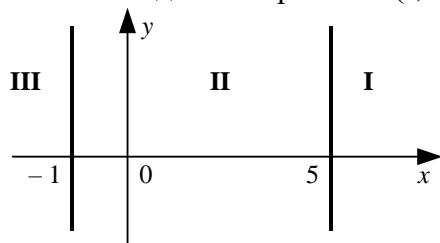
$$sum = sum + n$$

Переменную sum , накапливающую в себе сумму нескольких значений, будем называть «сумматором».

Предлагаем читателям обратить внимание на переменную-счетчик и переменную-сумматор, так как они неоднократно будут использоваться в программах далее.

5.3. Три и более вариантов действий

Рассмотрим задачу: «На плоскости выделены три зоны (I, II, III).



Дана координата x точки. Определить, в какую зону попала эта точка. Принять, что x не равно границам зон (-1 и 5)».

Решение

Можно в программе использовать 3 неполных варианта инструкции if (без ветви else):

```
if x < -1:
```

```

        print('В зону III')
if x > 5:
    print('В зону I')
if x > -1 and x < 5:
    print('В зону II')

```

Можно ли «сэкономить» одно слово if и использовать полный вариант инструкции?⁴

```

if x < -1:
    print('В зону III')
if x > 5:
    print('В зону I')
else:
    print('В зону II')

```

Проверим.

При $x = 10$ ответ будет правильным (см. схему на рис. 5.1).

При $x = 0$ – то же.

При $x = -8$ на экран будет выведено (почему – установите самостоятельно):

```

В зону III
В зону II

```

то есть ответ будет ошибочным.

Правильный вариант:

```

if x < -1:
    print('В зону III')
else: #Возможны 2 варианта действия, то есть надо использовать
      #полный вариант инструкции if
    if x > 5:
        print('В зону I')
    else:
        print('В зону II')

```

Получается, что одна инструкция if «вложена» в другую, поэтому такой вариант оформления называют «вложенной инструкцией if».

Можно «объединить» первое служебное слово else и следующее за ним слово if и записать слово elif:

```

if x < -1:
    print('В зону III')
elif x > 5:
    print('В зону I')
else:
    print('В зону II')

```

Использование служебного слова elif позволяет упростить запись и избежать чрезмерных отступов.

Ветвь elif может присутствовать несколько раз; наличие ветви else необязательно:

```

if ...:
    ...
elif ...:
    ...
elif ...:
    ...
elif ...:
    ...

```

⁴ Начинающие программисты, как правило, отвечают: «Да, можно». Так ли это – см. далее.

Если в цепочке if-elif-elif-... истинными являются несколько условий, то «срабатывает» первое из них. Например, программа:

```
if cost < 1000:  
    print ('Скидок нет')  
elif cost < 2000:  
    print('Скидка 2%')  
elif cost < 5000:  
    print('Скидка 5%')  
else:  
    print('Скидка 10%')
```

при `cost == 1500` выводит на экран:

Скидка 2%

хотя условие `cost < 5000` тоже выполняется.

Это означает, что в общем случае:

```
if <условие 1>:  
    <действия 1>  
elif <условие 2>:  
    <действия 2>  
elif <условие 3>:  
    <действия 3>  
elif ...:  
    ...  
elif <условие n>:  
    <действия n>  
else:  
    <действия (n + 1)>
```

схема работы вложенной инструкции if будет такой, как на рис. 5.6.

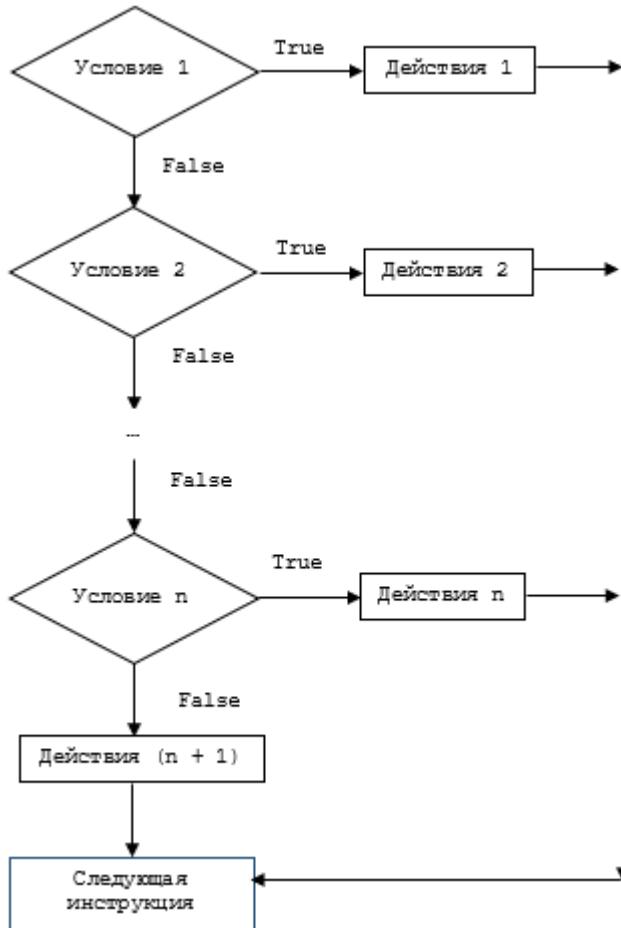


Рис. 5.6

Контрольные вопросы

В каких случаях в программе используется вложенная инструкция if? Как она оформляется? Как она «работает»? (Что происходит при ее выполнении?)

Задачи для разработки программ⁵

1. Известны возрасты Мити и Васи. Определить, кто из них старше или они одного возраста.

⁵ Во всех программах использовать одну инструкцию if.

2. Известен вес боксера-любителя (в кг, в виде вещественного числа). Известно, что вес таков, что боксер может быть отнесен к одной из трех весовых категорий:

- 1) легкий вес – до 60 кг;
- 2) первый полусредний вес – до 64 кг;
- 3) полусредний вес – до 69 кг.

Определить, в какой категории будет выступать данный боксер.

3. Даны два различных целых числа a и b ($a \neq 0, b \neq 0$). Определить, является ли a делителем b , или b является ли делителем a , или ни одно из чисел не является делителем другого.

4. Дан порядковый номер дня недели (1, 2, ..., 7). Вывести на экран его название (понедельник, вторник, ..., воскресенье).

5. Дан порядковый номер месяца (1, 2, ..., 12). Вывести на экран его название (январь, февраль, ..., декабрь).

6. Дан порядковый номер месяца (1, 2, ..., 12). Вывести на экран количество дней в этом месяце. Принять, что год не является високосным. В инструкции if использовать не более трех ветвей.

Задачник

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

Джордж Пойя

Соревнования по бегу

На соревнованиях по бегу первые четыре места заняли Борис, Армен, Максим и Эльдар.

На вопрос, какое они заняли места, мальчики ответили честно:

– Эльдар не занял ни первое и не третье место;

– Борис занял второе место;

– Армен занял более «высокое» место, чем Максим (естественно, что на пьедестале для победителей третье место выше четвертого, второе – выше третьего, первое – выше второго).

Какое место занял каждый?

Задача предназначена для учеников 1–7-х классов.

Три пачки тетрадей

На столе лежат три пачки тетрадей.

На первой пачке написано: «10 класс».

На второй пачке написано: «5 или 7 класс».

На третьей пачке написано: «7 класс».

Можно ли с учетом этого определить, в какой пачке какие тетради?

Может ли такое быть?

Одна девочка написала подруге: «У меня есть братья. Всего их — 100. Младшему 1000 лет, а старшему 1111 лет. Старший учится в 1001 классе». Может ли такое быть?

Шесть преподавателей

В педагогическом институте Аркадьева, Бабанова, Корсакова, Дашков, Ильин и Флеров преподают экономическую географию, английский язык, немецкий язык, историю, французский язык и математику.

Известно, что:

- 1) преподаватель немецкого языка и преподаватель математики в студенческие годы занимались художественной гимнастикой;
- 2) Ильин старше Флерова, но стаж работы у него меньше, чем у преподавателя экономической географии;
- 3) будучи студентками, Аркадьева и Бабанова учились вместе в одном университете. Все остальные окончили педагогический институт;
- 4) Флеров — сын преподавателя французского языка, но студентом у него не был;
- 5) преподаватель французского языка — самый старший из всех по возрасту и по стажу работы. Он работает в педагогическом институте с тех пор, как окончил его. Преподаватели математики и истории — его бывшие студенты;
- 6) Аркадьева старше преподавателя немецкого языка.

Кто какой предмет преподает?

Три задачи из ЕГЭ

Задача 1

На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1. Строится двоичная запись числа N .
2. К этой записи дописываются справа ещё два разряда по следующему правилу:
 - а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;
 - б) над этой записью производятся те же действия — справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R .

Укажите минимальное число R , которое превышает 43 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

Задача 2

Шифр кодового замка представляет собой последовательность из пяти символов, каждый из которых является цифрой от 1 до 4. Сколько различных вариантов шифра можно задать, если известно, что цифра 1 встречается ровно два раза, а каждая из других допустимых цифр может встречаться в шифре любое количество раз или не встречаться совсем?

Задача 3

Сколько существует различных трёхзначных чисел, записанных в четверичной системе счисления, в записи которых сумма первой и последней цифры строго больше цифры стоящей в середине?

Гостиница в Лимонии

В некой стране Лимонии в гостиницах установлен необычный порядок оплаты за проживание – нужно платить каждый день по одному лиму (лим – денежная единица этой страны). В стране имели хождение монеты по 1, 2, 3, 4 и 5 лимов. Какое минимальное количество монет должен иметь проживающий в гостинице, чтобы прожить в ней 7 дней? Какого достоинства должны быть эти монеты? Хозяин гостиницы может давать сдачу монетами, полученными им ранее.

От перестановки цифр...

1. Сколько имеется двузначных чисел, при перестановке цифр которых получается число, на 75 процентов большее?
2. Найдите хотя бы одно двузначное число, при перестановке цифр которого получается число, на 75 процентов большее?

Семинар

Как определить, простое ли число, не используя компьютер

В течение многих лет французский математик искал доказательство того, что гигантское число является простым. Его метод используется и по сей день, спустя 150 лет.

Люди тысячелетиями были очарованы простыми числами. Эти числа делятся только на 1 и на самих себя, в то время как любое другое целое число можно однозначно выразить как произведение нескольких простых чисел; например, $15 = 3 \times 5$. Простые числа по сути образуют периодическую таблицу математики. Они также хранят в себе много секретов. Они появляются на числовой прямой с определённой регулярностью, но их появление характеризуется колебаниями, которые пока не получается оценить количественно. Эта непредсказуемость вызывает у экспертов что-то вроде беспокойства.

А любители математики постоянно ищут новые простые числа. Текущий рекорд (по состоянию на октябрь 2025 года) для самого большого простого числа составляет $2^{136\,279\,841} - 1$, число с 41 024 320 цифрами. Простое произнесение этого числа вслух заняло бы примерно 240 дней!

Является ли число 170 141 183 460 469 231 731 687 303 715 884 105 727 простым? Прежде чем искать ответ в Интернете, подумайте, как вы могли бы ответить на этот вопрос без компьютера или даже цифрового калькулятора?

В 1800-х годах французский математик Эдуард Люка потратил годы на то, чтобы доказать, что это 39-значное число действительно простое. Как он это сделал? Люка, который, кстати, также придумал известную головоломку «Ханойская башня», разработал метод, который по-прежнему полезен и сегодня, более чем столетие спустя.

Простые числа с особой структурой

Обнаружено, что рекордные простые числа последних лет имеют схожую структуру: $2^p - 1$ (где p — простое число). Простые числа подобного вида называются простыми числами Мерсенна. И число, которому Люка посвятил почти два десятилетия своей жизни, также является простым числом Мерсенна, а именно $2^{127} - 1$. Но в этих простых числах Мерсенна есть одна хитрость: не каждое число вида $2^p - 1$ является простым числом для каждого простого значения p . Например, число Мерсенна $2^{11} - 1$ равно 2047, но оно является произведением чисел 23 и 89.

Поэтому в середине XIX века Люка задался вопросом, является ли $2^{127} - 1$ простым числом. Он столкнулся с большой проблемой. Это число огромно, состоит из 39 цифр, а в то

время учёный, понятное дело, не имел доступа к компьютеру. Ему пришлось вручную убедиться, что $2^{127} - 1$ не имеет делителей (кроме 1 и самого себя).

Один из способов выполнить эту задачу — пройти только все простые числа до $2^{127} - 1$ и убедиться, что оно не делится ни на одно из них. Но и это чрезвычайно трудоёмко и просто невыполнимо, если у вас нет списка всех простых чисел меньше этого.

Тест простых чисел Люка

Люка не сдался. Он разработал новый метод, основанный на открытиях своего коллеги Эвариста Галуа, который требовал значительно меньше вычислений.

Чтобы проверить, является ли $2^p - 1$ простым числом, Люка разработал следующий алгоритм:

1. Создать последовательность чисел, первым членом которой является $s_0 = 4$, а каждый последующий член s_n вычисляется в зависимости от предыдущего как $(s_{n-1})^2 - 2$. Таким образом, последовательность будет следующей: 4, 14 ($4^2 - 2$), 194 ($14^2 - 2$), 37634 и так далее.
2. Тогда $2^p - 1$ является простым числом тогда и только тогда, когда $(p - 2)$ -й член последовательности (то есть s_{p-2}) делится на $2^p - 1$ без остатка. То есть каждое простое число Мерсенна имеет это свойство, и, наоборот, каждое s_{p-2} является простым числом Мерсенна $2^p - 1$.

Таким образом, вместо того, чтобы делить число Мерсенна на *все* простые числа, меньшие $2^{127} - 1$, достаточно выполнить вычисления, чтобы найти s_{125} , а затем разделить *только* на $2^{127} - 1$. Это гораздо проще, верно?

На практике есть только одна маленькая — или, скорее, очень большая — проблема. Члены последовательности s_n растут чрезвычайно быстро (ведь там есть возведение к квадрату) — настолько быстро, что работать с ними не особенно практично. Поэтому эксперты прибегают к хитрости: они делят члены последовательности s_n на число Мерсенна и продолжают работать с остатком деления, если деление не даёт целого числа. Это не меняет вторую часть алгоритма, поэтому условие для простых чисел Мерсенна остаётся прежним: они должны делиться на s_{p-2} без остатка. Однако эта хитрость значительно уменьшает s_{p-2} .

Чтобы лучше понять тест на простоту, мы можем рассмотреть его на простом примере: возьмём число Мерсенна $2^5 - 1$, которое равно 31. Используя алгоритм, разработанный Люка, нам нужно вычислить s_3 , которое равно 37634. Деля это число на 31, мы получаем точный результат 1214. Это означает, что s_3 делится на $2^5 - 1$, и, следовательно, последнее является простым числом.

После многих лет кропотливой работы Эдуард Люка разработал этот тест на простоту и применил его к числу $2^{127} - 1$. Таким образом, он смог показать, что это действительно простое число. На сегодняшний день оно остаётся самым большим простым числом, найденным без помощи компьютера...

Статья подготовлена по материалам сайта habr.com, автор Вячеслав Голованов

Проверяем числа Мерсенна на «простоту» методом Люка

Разработаем программу, которая проверяет, является ли некоторое число Мерсенна простым согласно методу Люка (см. рубрику «Семинар»).

Прежде чем описывать методику разработки, обсудим, как определять числа $4, 14 (4^2 - 2), 194 (14^2 - 2), 37634$ и т.д.

В этой последовательности чисел каждое очередное число (его имя $s_{\text{очер}}$) рассчитывается в зависимости от предыдущего числа ($s_{\text{пред}}$) так:

$$s_{\text{очер}} = (s_{\text{пред}})^2 - 2$$

Как известно, такие зависимости между числами называются «рекуррентными».

Допустим, надо вычислить 4-й член такой последовательности, принимая, что число 4 имеет номер 0.

Сначала $s_{\text{пред}} = 4$

Считаем первое число:

$$s_{\text{очер}} = (s_{\text{пред}})^2 - 2 = 14$$

После этого число 14 ($s_{\text{очер}}$) может рассматриваться как предыдущий для нахождения второго числа:

$$s_{\text{пред}} = s_{\text{очер}}$$

В свою очередь, оно стало предыдущим для нахождения следующего числа и т.д.

Можем сказать, что для определения 4-го числа надо 4 раза рассчитать очередное число и принять его в качестве предыдущего числа.

Соответствующий фрагмент программы на школьном алгоритмическом языке (система программирования КУМир):

```

s_пред := 4 | Число с номером 0
нц для k от 1 до 4
    s_очер := s_пред ** 2 - 2
    s_пред := s_очер
кц
вывод s_очер

```

Допустим, что мы хотим проверить число Мерсенна $2^p - 1$ при $p = 11$, то есть 2047.

Так как $p = 7$, то нам надо определить 5-й член рекуррентной последовательности (в общем случае – $(p - 2)$ -й).

Вся программа решения этой задачи:

```

алг
нач цел p, число_Мерсенна, s_пред, s_очер, k
    вывод нс, "p := "
    ввод p
    | Соответствующее число Мерсенна
    число_Мерсенна := 2 ** p - 1
    | Рассчитываем  $(p - 2)$ -й член последовательности
    s_пред := 4
    нц для k от 1 до p - 2
        s_очер := s_пред ** 2 - 2
        s_пред := s_очер
    кц
    | Выводим ответ

```

```

вывод "Число_Мерсенна ", число_Мерсенна
| проверяя делимость s_очер числа на число_Мерсенна
если mod(s_очер, число_Мерсенна) = 0
    то
        вывод " это простое число"
    иначе
        вывод " простым не является"
    все
кон

```

где *mod* – функция школьного алгоритмического языка, определяющая остаток от деления своего первого аргумента на второй (в других языках программирования для этого используется не функция, а специальная операция).

Задания для самостоятельной работы

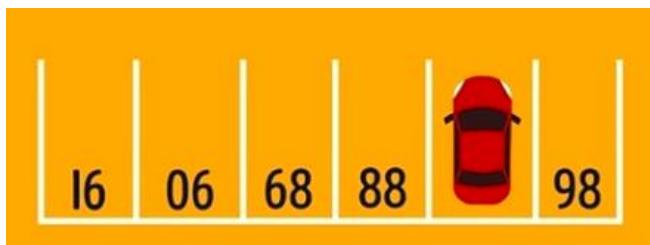
Разработав соответствующую программу (на языке программирования, который изучаете), проверьте на «простоту» как можно больше чисел Мерсенна вида $2^p - 1$ (то есть как можно больше значений p , где p – простое число).

Результаты исследования присылайте в редакцию. Авторы лучших результатов будут награждены дипломами.

«Ломаем» голову

Странная нумерация на парковке

На машиноместе с каким номером стоит автомобиль?



Три задачи со спичками

В всех задачах необходимо, переложив всего одну спичку, получить верные равенства.
Задача 1

$$76 - 8 = 72$$

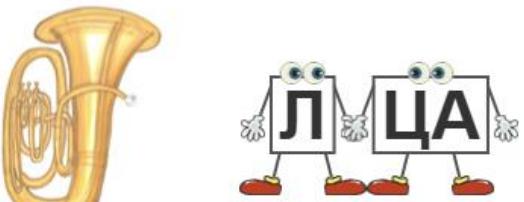
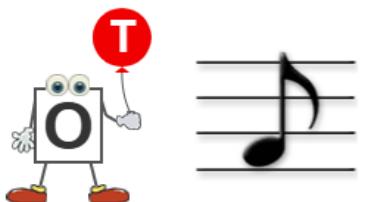
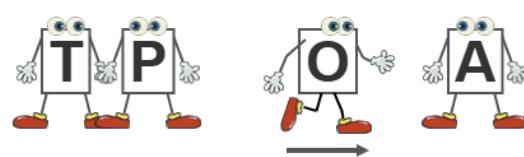
Задача 2

$$78 + 34 = 42$$

Задача 3

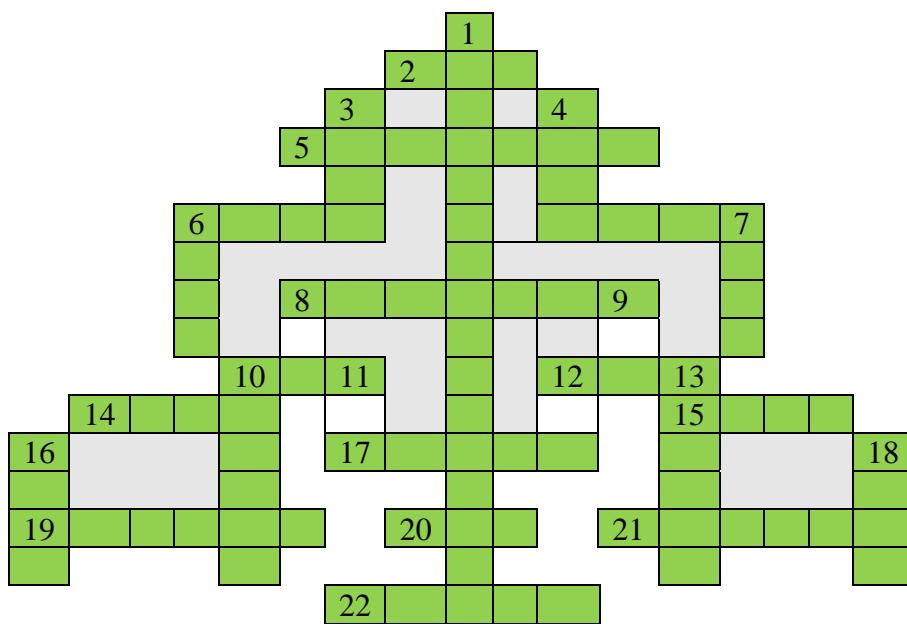
$$\boxed{6} \boxed{0} - \boxed{8} \boxed{0} = \boxed{6} \boxed{0}$$

Восемь ребусов на одну тему

<p>Ребус 1</p>  <p>1,4,3</p>	<p>Ребус 2</p>  <p>2 БЕЦ</p>
<p>Ребус 3</p> 	<p>Ребус 4</p>  <p>Л = ЯЧ</p>
<p>Ребус 5</p>  <p>Е РЕС</p>	<p>Ребус 6</p>  <p>Е = ФОР</p>
<p>Ребус 7</p>  <p>Г</p> <p>5,4,1,6,7</p>	<p>Ребус 8</p>  <p>+А</p> <p>2,3,4,4,3</p>

Новогодний кроссворд

Решите, пожалуйста, кроссворд:



По горизонтали:

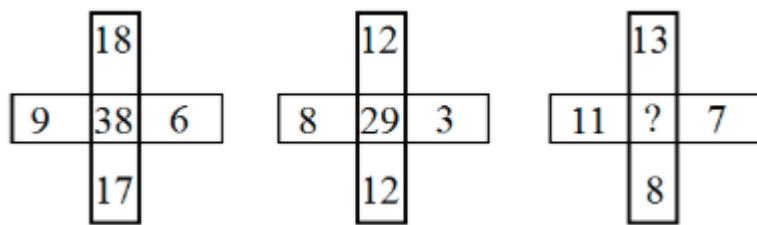
2. Цифра десятичной системы счисления.
5. Электронная схема, применяемая в регистрах компьютера для запоминания одного бита информации.
6. Выдумка с целью внушить другому преувеличенное представление о себе.
7. Цифра десятичной системы счисления (при записи справа налево — не ☺).
8. Часть массива (в программировании).
10. На жаргоне программистов — выявленная ошибка, неточность, требующая устранения.
12. Характеристика файла или переменной величины.
14. Знак препинания.
15. Цифра двоичной системы счисления.
17. Город в Швейцарии, в котором проходит ежегодный экономический форум, где обсуждают также вопросы, связанные с информационными технологиями.
19. Шрифт наклонного начертания.
20. Нижняя граница стека (противоположная его вершине).
21. Устройство ввода информации в компьютер.
22. Инструментальная программная оболочка, в которой происходит работа пользователя.

По вертикали:

1. Вид деятельности в сфере информационных технологий.
3. Конечное число точек на плоскости, соединенных отрезками кривых линий.
4. Несколько соединенных между собой компьютеров.
6. Единица измерения информации.
7. Структура данных, в которых применен принцип “ последним пришел — первым вышел”.
8. Буква греческого алфавита.
9. Носитель данных внешней памяти, обрабатываемый как единое целое.
10. Язык программирования.
11. Единица измерения времени.
12. Буква греческого алфавита
13. Популярная поисковая система (русский вариант названия)
16. Программная конструкция, обеспечивающая повторение одних и тех же операций.
18. Точка подключения внешних устройств к внутренней шине микропроцессора.

Два вопроса о знаке вопроса

Вопрос 1: «Чему равен ?»



Вопрос 2: «Чему равен ?»

Если

$$13 \times 12 = 651, \text{ а } 41 \times 23 = 448$$

то чему равно

$$24 \times 22 = ?$$

Ответы, пожалуйста, обоснуйте.

Три криптографические задачи

Криптография — это технология шифрования данных таким образом, чтобы их невозможно было посмотреть, прочитать или прослушать без расшифровки.

Решите три криптографические задачи.

1. Если в некотором слове заменить буквы на номера этих букв в алфавите, то получится следующее число:

2221222111121

Определите это слово.

2. Когда-то давно, когда еще существовала страна под названием СССР, был маршрут движения поезда, особенный тем, что если в названиях пунктов отправления и назначения поезда заменить буквы их номерами в алфавите, то они (названия) записываются с помощью всего лишь двух цифр:

211221 – 21221

Между какими городами курсировал поезд?

3. Один из самых древних методов шифрования текста – метод «атбаш» был изобретен еще в V веке до н. э. При его использовании русское слово

ЯБЕДА

было бы зашифровано как

АЮЬЫЯ

А как методом «атбаш» шифровалось бы слово БАЙТ?

Нахождение суммы значений средствами электронных таблиц

Самой распространенной задачей, решаемой с помощью электронных таблиц, является задача определения суммы ряда значений. Как известно, в программе Microsoft Excel такая задача решается с помощью функции СУММ⁶. Ее формат (общий вид) в простейшем случае следующий:

СУММ(диапазон)

где диапазон – диапазон ячеек, значения в которых должны быть просуммированы:

	A	B
1		12
2		2
...		...
33		33
34	Всего	=СУММ(B1:B33)
5		

Рис. 1

	A	B	C	D
1		Численность учащихся		
2	Параллель	«А»	«Б»	«В»
3	10-я	24	26	–
4	11-я	22	23	21
5				
6	Общая численность	=СУММ(B3:D4)		

Рис. 2

Конечно, если количество суммируемых значений невелико, то вместо использования функции СУММ можно записать формулу со знаками «+». При этом обратим внимание на то, что если при решении примера на рис. 2 в ячейке B6 записать (до ввода исходных данных) формулу:

$=B3+C3+D3+B4+C4+D4,$

то для приведенного случая после записи данных она вернет ошибочный результат типа #ЗНАЧ! (в одной из ячеек представлено нечисловое значение), в то время как функция СУММ учитывает только числовые значения в заданном диапазоне и даст правильный результат.

Просуммировать значения можно и не выписывая формулу с функцией СУММ «вручную». Например, в примере на рис. 1 для этого следует активировать ячейку B34, после чего на панели инструментов щелкнуть по инструменту «Сумма» Σ (или нажать одновременно клавиши **ALT** и **=**). В результате формула в указанной ячейке будет записана автоматически. При этом следует иметь в виду, что если в столбце выше ячейки с формулой есть пустые ячейки (как на рис. 3), то в качестве аргумента функции СУММ будет записан диапазон, так сказать, «после последней пустой ячейки»:

⁶ В электронной таблице OpenOffice Calc имя аналогичной функции – SUM.

	...	J
1		12
2		2
3		5
4		1
5		2
6		
7		5
8		6
9		=СУММ(J7:J8)

Рис. 3

Чтобы просуммировать и другие значения, следует предварительно выделить нужный диапазон, после чего использовать инструмент «Сумма» (или комбинацию клавиш **ALT + =**).

Описанные особенности использования инструмента «Сумма» (или комбинации клавиш **ALT + =**) касаются также определения суммы в строке:

	A	B	C	D	E	F	G	H
...								
13	3	12	6		1	2	4	=СУММ(E13:G13)
14								

Рис. 4

Приведенная в ячейке H13 формула будет «работать», если в ячейке H12 не записано число.

Если значения, которые надо просуммировать, находятся в несмежных диапазонах, то в качестве аргументов функции СУММ следует указать эти диапазоны, разделив их символом «;»:

=СУММ(B2:B34; E2:E34)

Одновременное суммирование в нескольких столбцах

Можно одновременно просуммировать значения в нескольких столбцах. Например, для нахождения трех сумм значений в столбцах B, C и D (см. рис. 5) следует:

- выделить диапазон B1:D5;
- щелкнуть по инструменту «Сумма» (или нажать **ALT + =**).

В результате под каждым столбцом автоматически добавится формула с функцией СУММ и в ячейках B6; C4 и D6 будет отображаться результат суммирования для каждого столбца. Это самый быстрый и удобный метод.

	A	B	C	D
1		12	5	4
2		4	6	5
3		6	66	9
4		1	13	12
5		5	2	6
6	<i>Всего</i>			

Рис. 5

Отметим особенности описанного метода.

Результат суммирования записывается в строку, которая является свободной в столбце с самым большим количеством значений:

	A	B	C
1	№	Заголовок 1	Заголовок 2
2	1	12	20
3	2	4	0,32
...			
24	23	15	4,3
25	24	22,4	
26	25	21,8	
27	Итого	128,6	64,98

Рис. 6

Интересно, что при этом будут просуммированы все числовые значения в выделенном диапазоне даже в случае наличия в нем пустых ячеек.

Значения в нескольких столбцах можно одновременно просуммировать и в случае несмежных диапазонов.

	A	B	C	D	E
1	№	Заголовок 1	Заголовок 2		Заголовок 3
2	1	12	20		23
3	2	4	0,32		4
...					
24	23	15	4,3		12
27	Итого	128,6	64,98		312,5

Рис. 7

Для этого перед щелчком по инструменту «Сумма» (или перед нажатием комбинации **ALT + =**) следует выделить все нужные диапазоны, используя клавишу **Ctrl** (см. рис. 7).

Можно также провести одновременное суммирование в нескольких строках (с учетом всех описанных особенностей).

Примечание. Функцию СУММ можно применить и для нахождения суммы нескольких конкретных значений:

=СУММ(42,5; 56,1; 12,8; 67,4)

хотя удобнее это сделать, предварительно введя значения в ячейки некоторого диапазона.

В заключение – еще об одной малоизвестной, но полезной, возможности. При решении рассматриваемой задачи часто возникает необходимость учета в сумме еще одного значения, записанного в новой строке (или в новом столбце). Например, если после оформления листа для расчета суммы значений в диапазоне B2:B6 (рис. 8), возникнет необходимость учета еще одного значения (рис. 9), то после вставки новой строки и ввода нового значения величина суммы не изменится⁷.

⁷ При вставке новой строки «внутрь» диапазона B2:B6 формула в ячейке B8 автоматически изменится, и новое значение будет учтено в сумме.

	A	B			A	B	
1	№	Стоимость			1	№	Стоимость
2	1	120			2	1	120
3	2	40			3	2	40
4	3	123			4	3	123
5	4	57			5	4	57
6	5	224			6	5	224
7		564			7	6	25
			Рис. 8		8		564

Рис. 9

Между тем, имеется возможность таким образом оформить формулу для суммирования, чтобы каждое (!) дополнительное значение при расчетах учитывалось. Для этого следует использовать функцию СМЕЩ. Эта функция возвращает ссылку на диапазон (в том числе – на отдельную ячейку), которые смещены относительно исходной ссылки на определенное число столбцов и строк. Её формат в общем случае:

СМЕЩ(базовый диапазон; смещение по строкам; смещение по столбцам; количество строк; количество столбцов)

где базовый диапазон – ячейка или диапазон, относительно которых происходит смещение (исходная ссылка); смещение по строкам – количество строк смещения; смещение по столбцам – количество столбцов смещения; количество строк и количество столбцов – соответственно, – количество строк и столбцов в возвращаемом функцией СМЕЩ диапазоне.

В рассматриваемой нами задаче суммирования понадобится ссылка на отдельную ячейку, которая смещена относительно другой отдельной ячейки по вертикали, поэтому значение третьего аргумента функции СМЕЩ можно указать равным нулю, а значения четвертого и пятого аргументов – равными 1 (или вообще опустить эти три значения – см. ниже).

Итак, приступим к оформлению формулы с функцией СУММ в ячейке В7 (см. рис. 8) для автоматического учета в ней новых строк.

Первая ячейка в диапазоне суммирования – В2, поэтому начальная часть формулы должна выглядеть так:

=СУММ(В2:

Последняя же ячейка диапазона суммирования (адрес которой будет меняться) всегда должна быть на одну ячейку выше ячейки с формулой (В7). Ссылка на эту ячейку может быть получена с помощью функции СМЕЩ:

СМЕЩ(В7;-1; 0; 1; 1)

или

СМЕЩ(В7;-1; ; ;)

Это значит, что формула для суммирования в ячейке В7 оформляется так:

=СУММ(В2 : СМЕЩ(В7;-1; ; ;))

Предлагаем читателям убедиться в том, что указанная формула будет учитывать *все* новые значения, а также в том, что описанный подход можно применить и к учету значений в новых столбцах.

Внимание! Конкурс!

Итоги конкурса № 75 «Пирамида из домино»

Напомним, что в качестве задания этого конкурса было предложено решить такую задачу «Расположите все 28 костяшек домина в виде пирамиды так, чтобы в каждой строке соседние костяшки соприкасались по правилам домино (0 с 0, 1 с 1 и т.д.) и при этом сумма всех чисел в строке была полным квадратом. Начиная сверху вниз это числа 0, 4, 16, 25, 25, 49, 49. Некоторые костяшки уже размещены».

Решение

Правильные ответы прислали и стали победителями конкурса:

— Баженова Олеся, Грэзин Михаил и Грызунова Дарья, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Белявский Давид и Протопопов Вячеслав, г. Пенза, школа № 73, учитель
Диков А.В.;

— Быстрова Софья, Ломакина Ульяна и Николаева Дарья, Волгоградская обл., г. Фролово, школа № 1 имени А.М. Горького, учитель **Фирсова М.Н.**;

— Земанек Антон, Москва, школа № 1530 «Школа Ломоносова», учитель Каменецкая Т.С.;

— Провоторов Степан, Барнаульский государственный педагогический колледж имени В.К. Штильке, преподаватель **Лукьянова Н.В.**;

— Садыкова Сирина и Мутаева Ксения, Республика Татарстан, Мамадышский р-н, совхоз “Мамадышский”, политехнический колледж, преподаватель **Порываева Н.С.**

Все перечисленные читатели будут награждены дипломами. Поздравляем!



Решение конкурса, представленное Олесей Баженовой, Михаилом Грэзиним и Дарьей Грызуновой, учениками школы № 11 г. Струнино Владимирская обл., учитель **Волков Ю.П.**

Конкурс № 77 «Графы». Тур 1

В качестве задания этого конкурса предлагается ответить на ряд вопросов, связанных с так называемыми «графами».

Как известно, в информатике *граф* – это конечное число точек на плоскости, некоторые из которых соединены линиями (точки называют *вершинами графа*, соединяющие их линии – *ребрами*).

Граф, в котором каждое ребро имеет направление от одной вершины к другой, называют ориентированным. Неориентированный граф – граф, в котором рёбра не имеют определённого направления.

Конкурс будет проходить в несколько туров, а его итоги будут подводиться с учетом всех туров в целом

Итак.

На плоскости имеются 9 точек



Вопрос 1

Три из любых точек можно соединить так, что получится неориентированный граф-треугольник. Сколько различных графов можно получить при этом? Какие? (Изобразите их в ответе).

Обратим внимание на то, что графы, приведенный ниже, не являются различными, так как любой из них получается смещением, поворотом или зеркальным отражением другого



Вопрос 2

Сколько различных неориентированных графов-четырехугольников (в том числе невыпуклых) можно получить, соединив четыре точки из девяти? Какие? (Изобразите их в ответе).

Ответы присылайте по адресу: mirinfo@infojournal.ru. Укажите в письме свои фамилию и имя, населенный пункт, номер школы и фамилию, имя и отчество учителя информатики. Срок представления ответов — 31 января 2026 г.

Ответы и решения задач и головоломок, опубликованных в октябрьском номере нашего журнала, представили:

— Андреева Тамара, Пономаренко Иван и Щукина Татьяна, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Атмажар Ирина, Борисова Дарья, Замилацкая София, Короткин Егор, Русаков Савелий, Садовой Артемий, Степанян Габриэль, Степанян Даниэль и Хамуратов Адам, ученики 4-го класса школа № 11 г. Струнино Владимирской обл., учитель **Волков Ю.П.**;

— Ашурков Максим, Зверева Мария, Шахрай Сергей и Шмарова Ирина, г. Пенза, школа № 73, учитель **Диков А.В.**

— Баженова Олеся, Горский Никита, Грэzin Михаил, Грызунова Дарья, Дауд Григорий, Денисова Дарья, Джафарова Мирослава, Ильин Владислав, Кулаков Егор, Лузянин Антон, Мурзина Евгения, Терещенко Артем, Фролов Егор и Шевченко Юлия, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Байбуза Василиса и Вертянов Юрий, филиал МОУ "СОШ "Созвездие" с. Тёпловка" в селе Ириновка, Новобурасский р-н Саратовской обл., учитель **Брунов А.С.**;

— Виктюк Иван и Зубрицкая Анна, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Гросс Тимофей, Кобилочный Герман, Сабаев Матвей, Фирсова Ирина и Чернышова К., Волгоградская обл., г. Фролово, школа № 1 имени А.М. Горького, учитель **Фирсова М.Н.**;

— Демченко Степан и Яковлева Светлана, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Карапетян Армен и Половинкина Ксения, Москва, школа № 1530 «Школа Ломоносова», учитель **Каменецкая Т.С.**

Спасибо всем!

