# Олимпиада школьников по прикладной математике и информатике факультета ВМК МГУ имени М. В. Ломоносова

## Очный тур. Решения задач

## 30 марта 2013 года

## 8 класс

Решения задач по информатике представлены как комментированные программы на языке программирования Python.

### Упаковщик.

Вводится текст четной длины (не менее чем 8 символов), в котором встречаются только нули и единицы, и при этом число единиц не менее чем в 7 раз больше числа нулей. Напишите две программы: первая сжимает этот текст (без потерь информации) в более короткий текст из нулей и единиц, а вторая, получая на вход результат работы первой программы, выдает на выходе исходный текст. Предложенные алгоритмы должны быть обоснованы. Для хранения текстов можно пользоваться массивами.

**Решение.** Пример программы кодирования и декодирования (годится в качестве любой из программ, о которых идет речь в условии задачи).

```
#!/usr/bin/env python
# coding: utf
# Поскольку задача состоит только в каком-то уменьшении объёма,
# первый вопрос - как упаковать семь единиц и один ноль (где бы он ни стоял)
# или восемь единиц. Произведём замену:
# 11 -> 0
# 10 -> 10
# 01 -> 110
# 00 -> 1110

Text=raw_input("Введите текст: ")
Res=""                              # Строка-результат поначалу пустая
if Text.count('0')<=len(Text)/7:    # Введён незакодированный текст
    # Словарь кодировки.
    Coder={'11': '0', '10': '10', '01': '110', '00': '1110'}
    for i in xrange(0,len(Text),2): # Цикл по всем парам
        Pair=Text[i:i+2]            # Очередная пара
        Res+=Coder[Pair]            # Добавляем её код в результат
else:                               # Введён закодированный текст
    # Словарь декодировки
    Decoder={'0': '11', '10': '10', '110': '01', '1110': '00'}
    start,end=0,0                   # Начало и конец очередного кода пары
    while start<len(Text):          # Последний '0' встречается в конце
        end=Text.find('0',start)    # Найдём '0' - конец очередного кода пары
```

```
        Code=Text[start:end+1]        # Код пары
        Res+=Decoder[Code]            # Добавляем пару в результат
        start=end+1                   # Продолжим поиск после '0'
print Res
```

Поясним. Пусть в исходном тексте было $k_0$ нулей и $k_1$ единиц ($k_1 > 0$), а в кодированном — $k_0'$ нулей и $k_1'$ единиц. По условию задачи $7k_0 \leqslant k_1$, а по построению кодированного текста $3k_0' \geqslant k_1'$. Так как условия $0 \leqslant 7k_0 \leqslant k_1$ и $3k_0 \geqslant k_1 > 0$ не могут выполняться для целых чисел одновременно, то по виду текста ясно, исходный он или кодированный.

Пусть, далее, исходный текст разбивался на $a_1$ слов '11', $a_2$ слов '10', $a_3$ слов '01', $a_4$ слов '00'. Тогда длина исходного текста равна $L = 2(a_1 + a_2 + a_3 + a_4)$, число нулей в нем равно $k_0 = a_2 + a_3 + 2a_4$, а число единиц в нем равно $k_1 = 2a_1 + a_2 + a_3$, и по условию $7(a_2 + a_3 + 2a_4) \leqslant k_1 = 2a_1 + a_2 + a_3$, то есть

$$2a_1 - 6a_2 - 6a_3 - 14a_4 \geqslant 0. \qquad (*)$$

Тогда длина кодированного текста будет равна $L' = a_1 + 2a_2 3a_3 + 4a_4$. Требуется доказать, что $L' < L$, то есть, что $a_1 + 2a_2 + 3a_3 + 4a_4 < 2(a_1 + a_2 + a_3 + a_4)$, и, значит, что

$$a_1 - a_3 - 2a_4 > 0. \qquad (**)$$

Из неравенства $(*)$ следует:

$$a_1 - a_3 - 2a_4 \geqslant 3a_2 + 2a_3 + 5a_4.$$

Значит, если $\max(a_2,\ a_3,\ a_4) > 0$, то неравенство $(**)$ доказано, если же $\max(a_2,\ a_3,\ a_4) = 0$, то $a_1 > 0$, $a_2 = a_3 = a_4 = 0$, и неравенство $(**)$ очевидно.

## Дракон.

Полоску бумаги длиной 1 метр складывают пополам, затем ещё пополам и так далее $N$ раз. При этом всякий раз в левой руке складывающего зажат край полоски (хотя бы один), а складывание осуществляется правой рукой по часовой стрелке (между разными складываниями бумага не выпускается из левой руки и не переворачивается). Затем полоску расправили, придав каждому сгибу угол ровно $90^\circ$. Полученную фигуру положили на прямоугольный стол так, чтобы каждый отрезок согнутой полоски был бы параллелен какой-то стороне стола, и фигура целиком умещалась на столе. Напишите программу, которая по заданному $N$ ($1 \leqslant N \leqslant 20$) вычисляет минимальную площадь такого стола в квадратных метрах.

**Решение.**

```
#!/usr/bin/env python
# coding: utf
# Данная ломаная - так называемая «драконова кривая»
# (точнее, N-е приближение «дракона Хартера-Хейтуэя», http://oeis.org/A014577).
N=input()

# Решение, основанное на повороте описанного прямоугольника
# относительно конца ломаной
def rotate(x0,y0,x,y):
 "Поворот точки (x;y) относительно точки (x0;y0) на 90° против часовой стрелки"
 return x0+(y-y0),y0-(x-x0)
```

```
# Координаты начала и конца ломаной (на шаге 0 - отрезка)
x0,y0,x1,y1=0.,0.,1.,0.
# Размеры описанного прямоугольника (минимальные и максимальные координаты)
mx,my,Mx,My=x0,y0,x1,y1
for i in xrange(N):
   # Повернём прямоугольник относительно конца ломаной
   (px,py),(Px,Py)=rotate(x1,y1,mx,my),rotate(x1,y1,Mx,My)
   # Вычислим новые размеры описанного прямоугольника
   mx,my,Mx,My=min(mx,px,Px),min(my,py,Py),max(Mx,px,Px),max(My,py,Py)
   # Повернём точку начала (однократно разложим полоску)
   # Результат - точка нового конца ломаной
   x1,y1=rotate(x1,y1,x0,y0)
print float((Mx-mx)*(My-my))/2**(2*N)
```

## Пятна в норме.

Древние кританские учёные полагали, что пятна на Луне не соответствуют пятнам на Солнце, только когда оба светила видны одновременно: по их подсчётам — не больше десяти часов в сутки. Всё остальное время пятна строго соответствуют друг другу. Измерения проводились каждый час в течение трёхсот лет без перерыва. Результат каждого измерения — показатель соответствия (целое число) — заносили на таблички (одно измерение — одна табличка) и складывали в ящики. Во время Большой кританской катастрофы все ящики сгорели, а таблички перемешались. Измерение несоответствующих пятен могло давать различные результаты (возможно, все результаты различаются), а измерение соответствующих пятен — всегда одно и то же число. Напишите программу, которая поможет современным кританским ученым выяснить по данным из табличек, что это за число. Дополнительное условие: запрещается заводить структуры, сопоставимые по размеру с количеством входных данных (скажем, миллион чисел).

**Решение.**

```
#!/usr/bin/env python
# coding: utf

# Это задача Брудно в чистом виде.
# Решение базируется на том факте, что искомых одинаковых чисел больше,
# чем всех остальных, и сумма последовательности из 1 и -1, где 1
# соответствует искомому числу, а -1 - любому другому, положительна.
# Единственное затруднение - вычисление общего количества измерений
# (с учётом високосных лет).
# Однако для решения задачи достаточно меньшего количества измерений:
# число искомых чисел составляет не менее 7/12 общего числа измерений и после,
# скажем, 23/24 проверок всё должно быть уже ясно.

Target, Hits = 0, 0                     # Искомое число и число «попаданий»
Num=300*365*24*23/24                    # Количество измерений для проверки
for i in xrange(Num):
    N=input()
    if Hits==0:                         # Хорошей гипотезы нет
        Target=N                        # Выберем новое число
```

```
    if N==Target:                           # Гипотеза подтверждается
        Hits+=1
    else:                                   # Гипотеза не подтверждается
        Hits-=1
print Target
```

Пояснение. Рассмотрим ситуацию, в которой Hits впервые после начального присвоения становится равным 0. Это происходит после анализа некоторого фрагмента последовательности. В любом случае, была ли гипотеза хорошей (Target есть искомое число) или плохой (Target не есть искомое число), в отсмотренной последовательности содержится не более половины искомых чисел. Тоже верно и для любой ситуации, в которой Hits равна нулю. Следовательно, по условию, в оставшейся части последовательности содержится строго больше половины искомых чисел, и задача сводится к решению исходной на оставшейся последовательности. Значит, на последнем (возможно, единственном) фрагменте Hits уже не может принимать нулевого значения, и гипотеза на этом фрагменте бует верна, что и составляет ответ.

## Подряд идущие цифры.

Составьте девятизначное десятичное число, в котором каждая цифра от 1 до 9 была бы использована по одному разу так, чтобы при некоторой расстановке знаков '+' и '·' между цифрами этого числа (не обязательно между любыми соседними) получилось числовое выражение, равное 1755, а при другой расстановке знаков — числовое выражение, равное 2013.

**Пример решения.** Искомым числом может являться число 172645389, так как

$$1726 + 4 + 5 + 3 + 8 + 9 = 1755, \qquad 1726 + 4 \cdot 5 + 3 \cdot 89 = 2013.$$

## Делимость и суммы.

Существует ли

а) число, кратное 2013 и имеющее в десятичной записи сумму цифр, равную 1755,

б) число, кратное 1755 и имеющее в десятичной записи сумму цифр, равную 2013?

**Решение.** Заметим, что $1755 = 3^3 \cdot 5 \cdot 13$, $2013 = 3 \cdot 11 \cdot 61$.

а) Такое число существует. Так как сумма цифр числа 8052 равна 15, а само число 8052 кратно 2013, то число

$$\underbrace{80528052\ldots8052}_{117 \text{ блоков } `8052'}$$

кратно 2013 и имеет сумму цифр, равную 1755.

б) Такого числа не существует. Допустим, некоторое число $A$ кратно 1755 и имеет в десятичной записи сумму цифр, равную 2013. Тогда $A$ кратно 9, и по свойству делимости на 9 сумма цифр числа $A$ кратна 9, что не так, ибо 2013 не делится на 9 нацело. Получили противоречие, числа $A$ не существует.

**Ответ.** а) Да. б) Нет.

## Неизвестное выражение.

Известно, что для всех (действительных) чисел $x$, $y$ некоторое выражение $f(t)$, зависящее от переменной $t$, удовлетворяет равенству

$$f(x + 2y) - f(y - x) = 3x + y.$$

Найти все возможные значения выражения

$$\frac{f(5\cdot 2013)-f(2013)}{f(4\cdot 2013)-f(2\cdot 2013)}.$$

**Решение.**

$$\frac{f(5\cdot 2013)-f(2013)}{f(4\cdot 2013)-f(2\cdot 2013)}=\frac{f(2013+2\cdot(2\cdot 2013))-f((2\cdot 2013)-2013)}{f(0+2\cdot(2\cdot 2013))-f(2\cdot 2013-0)}=$$

$$=\frac{3\cdot 2013+2\cdot 2013}{3\cdot 0+2\cdot 2013}=\frac{5}{2}.$$

(В числителе воспользовались приведенным в условии задачи свойством при $x=2013$, $y=2\cdot 2013$, а в знаменателе — тем же свойством при $x=0$, $y=2\cdot 2013$).

**Ответ.** $2,5$.

## Три в одном.

Разрежьте шахматную доску $10\times 10$ вдоль границ клеток так, чтобы из полученных частей можно было бы сложить одновременно две шахматные доски — доску $8\times 8$ и доску $6\times 6$, а общая длина линий разреза при этом была бы минимальной. Минимальность обоснуйте.

**Решение.** Поскольку никакие две точки, лежащие на противоположных сторонах доски $10\times 10$, не могут оказаться на одном куске доски размера $8\times 8$ или $6\times 6$, то на доске $10\times 10$ должен быть как разрез, отделяющий верхнюю сторону от нижней, так и разрез, отделяющий левую сторону от правой. Значит, общая длина линий разреза не может быть меньше 20. На рисунке 1 продемонстрировано, как можно разрезать доску так, чтобы общая длина линий разреза была равна 20 (при этом доска $6\times 6$ имеется как отдельный кусок, а из трех других кусков легко сложить доску $8\times 8$).

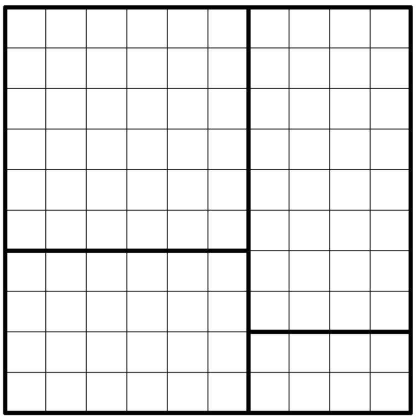

*Рис. 1.*

## Наибольшая диагональ.

Определим $n$-угольник как часть плоскости, ограниченную замкнутой несамопересекающейся $n$-звенной ломаной снаружи (звенья ломаной называются сторонами, а вершины

— вершинами $n$-угольника). Выпуклым назовем такой $n$-угольник, который вместе с любой парой точек целиком содержит и отрезок, соединяющий эти точки. Диагональю $n$-угольника называется отрезок, соединяющий вершины $n$-угольника и не являющийся стороной $n$-угольника. Каково наибольшее число сторон выпуклого 30-угольника, равных по длине его наибольшей диагонали?

**Решение.** Пусть наибольшая длина диагонали выпуклого 30-угольника равна $d$. Покажем сначала, что в этом выпуклом 30-угольнике не может быть более двух сторон длины $d$.

Допустим, это не так. Тогда в 30-угольнике найдутся две стороны длины $d$, не имеющие общей вершины (обозначим эти стороны через $AB$ и $CD$, см. рис. 2, и пусть при этом $ABCD$ — выпуклый чатырехугольник, и отрезки $AC$ и $BD$ пересекаются в точке $K$). По неравенству треугольника имеем:

$$AK + KB > AB, \qquad CK + KD > CD,$$

значит,

$$AK+KB+CK+KD > AB+CD, \quad (AK+KC)+(BK+KD) > AB+CD, \quad AC+BD > 2d,$$

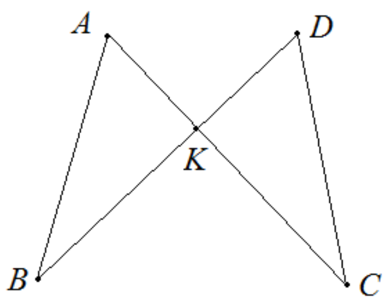


*Рис. 2.*

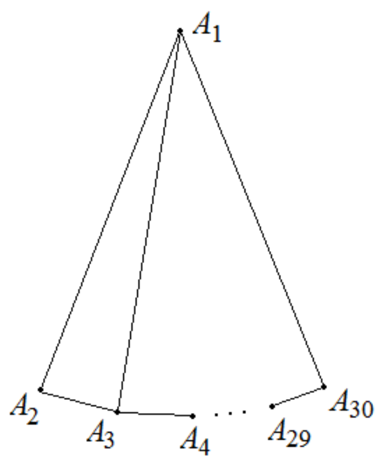


*Рис. 3.*

и $\max(AC,\ BD) > d$, то есть $d$ не может быть наибольшей длиной диагонали 30-угольника. Получаем противоречие, доказывающее, что в этом выпуклом 30-угольнике не может быть более двух сторон длины $d$. На рисунке 3 приведен пример построения выпуклого 30-угольника, у которого ровно две стороны по длине равны самой длинной диагонали (вершина $A_1$ — центр окружности, на которой лежат вершины $A_2$, $A_3$, $\ldots$, $A_{30}$, длина наибольшей диагонали $A_1A_3$ равна радиусу этой окружности, ровно две стороны $A_1A_2$ и $A_1A_{30}$ имеют такую длину).

**Ответ.** 2.

# Олимпиада школьников по прикладной математике и информатике факультета ВМК МГУ имени М. В. Ломоносова

## Очный тур. Решения задач

## 30 марта 2013 года

## 9-10 классы

Решения задач по информатике представлены как комментированные программы на языке программирования Python.

### Дракон.

Полоску бумаги длиной 1 метр складывают пополам, затем ещё пополам и так далее $N$ раз. При этом всякий раз в левой руке складывающего зажат край полоски (хотя бы один), а складывание осуществляется правой рукой по часовой стрелке (между разными складываниями бумага не выпускается из левой руки и не переворачивается). Затем полоску расправили, придав каждому сгибу угол ровно $90°$. Полученную фигуру положили на прямоугольный стол так, чтобы каждый отрезок согнутой полоски был бы параллелен какой-то стороне стола, и фигура целиком умещалась на столе. Напишите программу, которая по заданному $N$ ($1 \leqslant N \leqslant 20$) вычисляет минимальную площадь такого стола в квадратных метрах.

**Решение.**

```
#!/usr/bin/env python
# coding: utf
# Данная ломаная - так называемая «драконова кривая»
# (точнее, N-е приближение «дракона Хартера-Хейтуэя», http://oeis.org/A014577).
N=input()

# Решение, основанное на повороте описанного прямоугольника
# относительно конца ломаной
def rotate(x0,y0,x,y):
 "Поворот точки (x;y) относительно точки (x0;y0) на 90° против часовой стрелки"
 return x0+(y-y0),y0-(x-x0)

# Координаты начала и конца ломаной (на шаге 0 - отрезка)
x0,y0,x1,y1=0.,0.,1.,0.
# Размеры описанного прямоугольника (минимальные и максимальные координаты)
mx,my,Mx,My=x0,y0,x1,y1
for i in xrange(N):
   # Повернём прямоугольник относительно конца ломаной
   (px,py),(Px,Py)=rotate(x1,y1,mx,my),rotate(x1,y1,Mx,My)
   # Вычислим новые размеры описанного прямоугольника
   mx,my,Mx,My=min(mx,px,Px),min(my,py,Py),max(Mx,px,Px),max(My,py,Py)
   # Повернём точку начала (однократно разложим полоску)
```

```
   # Результат - точка нового конца ломаной
   x1,y1=rotate(x1,y1,x0,y0)
print float((Mx-mx)*(My-my))/2**(2*N)
```

## Пятна в норме.

Древние critанские учёные полагали, что пятна на Луне не соответствуют пятнам на Солнце, только когда оба светила видны одновременно: по их подсчётам — не больше десяти часов в сутки. Всё остальное время пятна строго соответствуют друг другу. Измерения проводились каждый час в течение трёхсот лет без перерыва. Результат каждого измерения — показатель соответствия (целое число) — заносили на таблички (одно измерение — одна табличка) и складывали в ящики. Во время Большой кританской катастрофы все ящики сгорели, а таблички перемешались. Измерение несоответствующих пятен могло давать различные результаты (возможно, все результаты различаются), а измерение соответствующих пятен — всегда одно и то же число. Напишите программу, которая поможет современным кританским ученым выяснить по данным из табличек, что это за число. Дополнительное условие: запрещается заводить структуры, сопоставимые по размеру с количеством входных данных (скажем, миллион чисел).

**Решение.**

```
#!/usr/bin/env python
# coding: utf

# Это задача Брудно в чистом виде.
# Решение базируется на том факте, что искомых одинаковых чисел больше,
# чем всех остальных, и сумма последовательности из 1 и -1, где 1
# соответствует искомому числу, а -1 - любому другому, положительна.
# Единственное затруднение - вычисление общего количества измерений
# (с учётом високосных лет).
# Однако для решения задачи достаточно меньшего количества измерений:
# число искомых чисел составляет не менее 7/12 общего числа измерений и после,
# скажем, 23/24 проверок всё должно быть уже ясно.

Target, Hits = 0, 0                     # Искомое число и число «попаданий»
Num=300*365*24*23/24                    # Количество измерений для проверки
for i in xrange(Num):
    N=input()
    if Hits==0:                         # Хорошей гипотезы нет
        Target=N                        # Выберем новое число
    if N==Target:                       # Гипотеза подтверждается
        Hits+=1
    else:                               # Гипотеза не подтверждается
        Hits-=1
print Target
```

Пояснение. Рассмотрим ситуацию, в которой Hits впервые после начального присвоения становится равным 0. Это происходит после анализа некоторого фрагмента последовательности. В любом случае, была ли гипотеза хорошей (Target есть искомое число) или плохой (Target не есть искомое число), в отсмотренной последовательности содержится не более половины искомых чисел. Тоже верно и для любой ситуации, в которой Hits равна нулю. Следовательно, по условию, в оставшейся части последовательности содержится строго больше половины искомых чисел, и задача сводится к решению исходной на оставшейся последовательности. Значит, на последнем (возможно, единственном) фрагменте Hits уже не может принимать нулевого значения, и гипотеза на этом фрагменте бует верна, что и составляет ответ.

## Живой лабиринт.

После игры в Королевский Крокет Алисе предложили пройти Королевский Лабиринт, в котором нет ни ежей, ни фламинго, а вот роль стенок, как и прежде, исполняют гвардейцы. Как и прежде, гвардейцы не стоят на месте: за каждый ход Алисы на одну клетку каждый гвардеец перемещается также на одну клетку по следующему пути: вниз, влево, вверх, вправо — и так до бесконечности. Если на пути человек (гвардеец или Алиса) вынужден пересечь границу лабиринта, по законам Страны Чудес он появляется с противоположной стороны (например, абсцисса $M+1$ превращается в 1, а ордината 0 — в $N$). Напишите программу, отвечающую на вопрос: может ли Алиса пройти заданный лабиринт. В начале пути Алиса стоит в левой верхней клетке с координатами (1; 1) (на момент старта эта клетка свободна), выход находится в клетке $(M;\ N)$ (размер лабиринта). Лабиринт задается массивом чисел $M \times N$, в котором 0 означает свободное (на момент старта) место, а 1 — место, занятое гвардейцем.

**Решение.**

```
#!/usr/bin/env python
# coding: utf

# Проще всего представить Королевский Лабиринт четырёхэтажным, где каждый этаж
# соответствует положению гвардейцев в четырёх последовательных случаях,
# а возможные перемещения Алисы ограничить постоянным подъёмом на один этаж
# (5-й превращается в 1-й). Начальное положение Алисы - (1;1;1), положение
# четырёх выходов - (M;N;любой этаж). Тем самым мы сводим задачу к обычному
# обходу трёхмерного лабиринта.

# Индексы в программе начинаются не с 1, а с 0

# Введём N - количество линий лабиринта, число столбцов определим по ширине
N=input()
Lab=[[],[],[],[]]                      # Четыре этажа
# Лабиринт вводится построчно, в виде списков 0 и 1 через запятую
for i in xrange(N):
    Lab[0].append(list(input()))       # Очередная линия первого этажа
M=len(Lab[0][0])                       # Все линии лабиринта, например,
                                       # нулевая, одинаковой ширины
# Теперь сформируем оставшиеся три этажа
from copy import deepcopy
# Сдвиг вниз: последняя линия, после неё - все остальные, кроме последней
Lab[1]=deepcopy(Lab[0][-1:]+Lab[0][:-1])
# Свдвиг влево: каждая линия начинается с первой колонки,
# нулевая колонка - в конце
Lab[2]=[l[1:]+l[:1] for l in Lab[1]]
# Сдвиг вверх: все линии, начиная с первой, нулевая - в конце
```

```
Lab[3]=deepcopy(Lab[2][1:]+Lab[2][:1])

# Разрешённые перемещения Алисы.
Next=((1,0,1),(0,1,1),(M+1,0,1),(0,N+1,1))
# Решим задачу методом Флойда
Todo=[(0,0,0)]                          # Список достижимых клеток,
                                        # в которых надо вести поиск
Step,Exit=1,False                       # Номер шага и индикатор выхода
while Todo and not Exit:                # Если список будет пуст,
                                        # больше искать негде
    Tonew=[]                            # Новый список поиска
    Step+=1
    for x,y,z in Todo:                  # Цикл по всем тройкам координат
                                        # в списке поиска
        for dx,dy,dz in Next:           # Цикл по всем возможных ходам
            nx,ny,nz=(x+dx)%M,(y+dy)%N,(z+dz)%4
            if Lab[nz][ny][nx]==0:      # Неисследованная клетка...
                Lab[nz][ny][nx]=Step    # ...теперь исследована!
                Tonew.append((nx,ny,nz))# Добавим её в список поиска
                if ny==N-1 and nx==M-1: # Да это же выход!
                    Exit=True           # Поиск окончен
    Todo=Tonew                          # Ищем дальше
if Exit:                                # Если список поиска не пуст,
                                        # в нём был найден выход
    print "Выход есть"
else:
    print "Выхода нет"
```

## По круглому дну.

Шарик (пренебрежимо малого размера) катается по имеющему радиус 1 круглому дну сосуда цилиндрической формы. Шарик совершает перемещения по хордам дна, имеющим равную длину (следующая хорда всегда не совпадает с предыдущей). Антип время от времени посматривает на шарик, но все время считает удары шарика о стенки. В какой-то момент Антип замечает, что через 12 ударов шарик попадает в одну и ту же точку стенки сосуда.

а) На каком минимальном расстоянии от центра дна сосуда мог оказываться шарик при этих условиях?

б) На каком минимальном расстоянии от центра дна сосуда гарантированно оказывался шарик при этих условиях?

**Решение.** Пусть дно имеет вид круга с центром в точке $O$ и радиусом 1 (см. рисунок 4), $AB$ — одна из хорд, по которым движется шарик, $H$ — середина $AB$ (и основание высоты $OH$ равнобедренного треугольника $OAB$). Заметим: длина окружности равна $2\pi$. Обозначим через $l$ длину меньшей из стягиваемых хордой $AB$ дуг окружности. По условию $l < \pi$ и существует натуральное число $n$ такое, что $12l = 2\pi n$. Значит, $l = \frac{\pi n}{6} < \pi$, $n \in \{1,\ 2,\ 3,\ 5,\ 5\}$ и чем больше $n$, тем больше длина $AB$ (центральный угол $\angle AOB$ при этом будет иметь меру $\frac{\pi n}{6}$ радиан). Поскольку любая хорда $A'B'$, по которой движется шарик, равна по длине хорде $AB$ (и отлична от диаметра окружности), треугольники $ABO$ и $A'B'O$ равны по третьему признаку, имеем: минимальное расстояние от шарика до центра дна есть длина отрезка $OH$. Из прямоугольного треугольника $OAH$ получаем:

$OH = OA\cos\angle AOH = \cos(0,5\angle AOB) = \cos\frac{\pi n}{12}$.

а) Это расстояние $OH$ минимально возможно при $n = 5$ и равно

$$OH = \cos\frac{5\pi}{12} = \cos\left(\frac{2\pi}{3} - \frac{\pi}{4}\right) = \cos\frac{2\pi}{3}\cos\frac{\pi}{4} + \sin\frac{2\pi}{3}\sin\frac{\pi}{4} = \frac{\sqrt{6}-\sqrt{2}}{4},$$

что является ответом в пункте (а) (шарик при этом движется по хордам, образующим 12-конечную звезду).

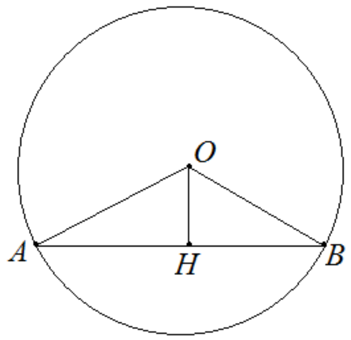


*Рис. 4.*

б) Расстояние $OH$ максимально возможно при $n = 1$ и равно

$$OH = \cos\frac{\pi}{12} = \sqrt{\frac{\cos\frac{\pi}{6}+1}{2}} = \sqrt{\frac{3+2\sqrt{3}+1}{8}} = \frac{\sqrt{6}+\sqrt{2}}{4},$$

что является ответом в пункте (б), так как при любом возможном движении шарик в какой-то момент оказывается на таком расстоянии от центра (при $n = 1$ шарик движется по сторонам правильного вписанного в окружность 12-угольника).

**Ответ.** а) $\frac{\sqrt{6}-\sqrt{2}}{4}$, б) $\frac{\sqrt{6}+\sqrt{2}}{4}$.

## Делимость и суммы.

Существует ли

а) число, кратное 2013 и имеющее в десятичной записи сумму цифр, равную 1755,

б) число, кратное 1755 и имеющее в десятичной записи сумму цифр, равную 2013?

**Решение.** Заметим, что $1755 = 3^3 \cdot 5 \cdot 13$, $2013 = 3 \cdot 11 \cdot 61$.

а) Такое число существует. Так как сумма цифр числа 8052 равна 15, а само число 8052 кратно 2013, то число

$$\underbrace{80528052\ldots8052}_{117 \text{ блоков } `8052'}$$

кратно 2013 и имеет сумму цифр, равную 1755.

б) Такого числа не существует. Допустим, некоторое число $A$ кратно 1755 и имеет в десятичной записи сумму цифр, равную 2013. Тогда $A$ кратно 9, и по свойству делимости на 9 сумма цифр числа $A$ кратна 9, что не так, ибо 2013 не делится на 9 нацело. Получили противоречие, числа $A$ не существует.

**Ответ.** а) Да. б) Нет.

## Дружественные квадратные трехчлены.

Найти все такие пары квадратных трехчленов $x^2+ax+b$ и $x^2+cx+d$, что $x^2+ax+b$ имеет два различных корня $c$ и $d$, а $x^2+cx+d$ имеет два различных корня $a$ и $b$.

**Решение.** По теореме Виета получаем:

$$\begin{cases} a+b=-c, \\ ab=d, \\ c+d=-a, \\ cd=b, \end{cases} \quad \Leftrightarrow \quad \begin{cases} b=-a-c, \\ a(-a-c)=d, \\ d=-a-c, \\ c(-a-c)=b. \end{cases}$$

Из первого и третьего уравнений последней системы выводим:

$$\begin{cases} b=d=-a-c, \\ ab=d=b, \end{cases} \quad \Rightarrow \quad \left[ \begin{array}{l} b=0, \\ a=c=1. \end{array} \right.$$

Если $b=0$, то $d=0$, $c=-a$, и вид получаемых в этом случае дружественных квадратных трехчленов таков: трехчлен $x^2+qx$ с корнями $\{0;\ -q\}$ и трехчлен $x^2-qx$ с корнями $\{0;\ q\}$ при любом действительном не равном нулю значении $q$ (при $q=0$ каждый из этих трехчленов имеет только один корень).

Если же $a=c=1$, то $b=d=-2$, и вид получаемых в этом случае дружественных квадратных трехчленов таков: трехчлен $x^2+x-2$ с корнями $\{1;\ -2\}$ и трехчлен $x^2+x-2$ с корнями $\{1;\ -2\}$.

**Ответ.** $(x^2+ax+b;\ x^2+cx+d) \in \{(x^2+qx;\ x^2-qx);\ (x^2+x-2;\ x^2+x-2) \mid q \neq 0\}$.

## Функции.

Найти все функции $f(x)$, определенные на всей числовой прямой, такие, что при всех действительных числах $x$, $y$, $z$ справедливо равенство

$$f(x+y+z)=f^2(x)+f(y)f(z).$$

**Решение.** Обозначим $a=f(0)$. Имеем: $f(0+0+0)=f^2(0)+f(0)\cdot f(0)$, $a=a^2+a^2$, $a(2a-1)=0$, $a\in\{0;\ 0,5\}$.

Если $a=f(0)=0$, то для любого значения $x$ справедливо равенство:

$$f(0+x+0)=f^2(0)+f(x)\cdot f(0), \qquad f(x)=0.$$

Значит, в этом случае $f(x)=0$.

Если $a=f(0)=0,5$, то для любого значения $x$ справедливо равенство:

$$f(0+x+0)=f^2(0)+f(x)\cdot f(0), \qquad f(x)=0,25+0,5f(x), \qquad f(x)=0,5.$$

Значит, в этом случае $f(x)=0,5$.

**Ответ.** Искомыми функциями являются константы $0$ и $0,5$ и только они.

## Середины и перпендикуляры.

Площадь остроугольного треугольника равна 1. Из середины каждого отрезка, соединяющего середину стороны треугольника с точкой пересечения всех серединных перпендикуляров треугольника, опускаются перпендикуляры на те стороны треугольника, которые не имеют общих точек с отрезком. Построенные шесть перпендикуляров ограничивают некоторый выпуклый шестиугольник. Найти площадь этого шестиугольника.

**Решение.** Пусть $ABC$ — исходный остроугольный треугольник, $D$, $E$, $F$ — середины его сторон $BC$, $AC$, $AB$ соответственно, $O$ — точка пересечения серединных перпендикуляров $\triangle ABC$ (см. рис. 5). Точка $O$ находится внутри $\triangle ABC$, так как этот треугольник — остроугольный. Пусть $D'$, $E'$, $F'$ — середины отрезков $OD$, $OE$, $OF$ соответственно. Проведя через точку $D'$ прямую, параллельную прямой $BC$, через точку $E'$ — прямую, параллельную прямой $AC$, а через точку $F'$ — прямую, параллельную прямой $AB$, получим ограниченный этими прямыми треугольник $A'B'C'$, подобный треугольнику $ABC$ по двум углам (см. рис. 5). Так как коэффициент подобия этих треугольников равен 0,5 (что легко заметить), то $S(\triangle A'B'C') = 0,25 \cdot S(\triangle ABC) = 0,25$.

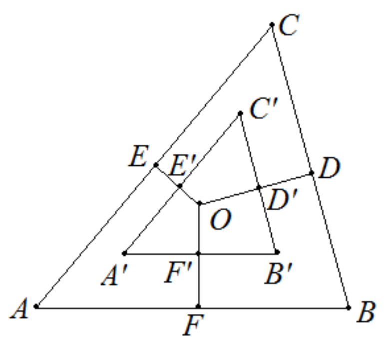


*Рис. 5.*

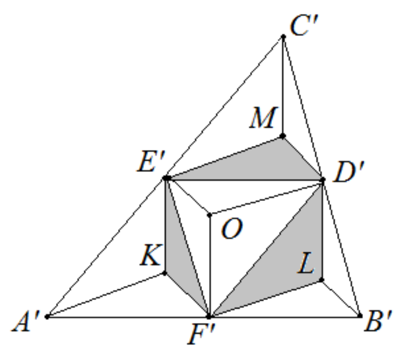


*Рис. 6.*

Рассмотрим теперь $\triangle A'B'C'$ (см. рис. 6). В этом треугольнике $D'$, $E'$, $F'$ — середины сторон $B'C'$, $A'C'$, $A'B'$ соответственно, $O$ — точка пересечения серединных перпендикуляров, а отрезки $D'E'$, $D'F'$, $E'F'$ являются его средними линиями и, следовательно, разбивают треугольник $\triangle A'B'C'$ на равные (по третьему признаку, например) треугольники $A'E'F'$, $B'D'F'$, $C'D'E'$, $D'E'F'$, имеющие площади $0,0625$ каждый и подобные треугольнику $A'B'C'$. Пусть точки $K$, $L$, $M$ — точки пересечения высот треугольников $A'E'F'$, $B'D'F'$, $C'D'E'$ соответственно (см. рис. 6). Тогда выпуклый шестиугольник $D'ME'KF'L$ и является шестиугольником, о котором говорится в условии задачи. По второму признаку равенства треугольников $\triangle E'MD' = \triangle A'KF'$, $\triangle D'LF' = \triangle E'KA'$. Значит,

$$S(D'ME'KF'L) = S(\triangle D'E'F') + (S(\triangle E'KF') + S(\triangle D'LF') + S(\triangle E'MD')) =$$

$$= S(\triangle D'E'F') + S(\triangle A'E'F') = 2 \cdot 0,0625 = 0,125.$$

**Ответ.** 0,125.