Олимпиада школьников по прикладной математике и информатике факультета вычислительной математики и кибернетики Московского государственного университета имени М. В. Ломоносова

Очный тур (4 апреля 2015 года). 8-9 классы

Задачи по математике

1. Разрезание прямоугольника. Какое наибольшее количество прямоугольников 1×6 можно вырезать из прямоугольника 8×9? Ответ обосновать.

Ответ: 11.

Решение.

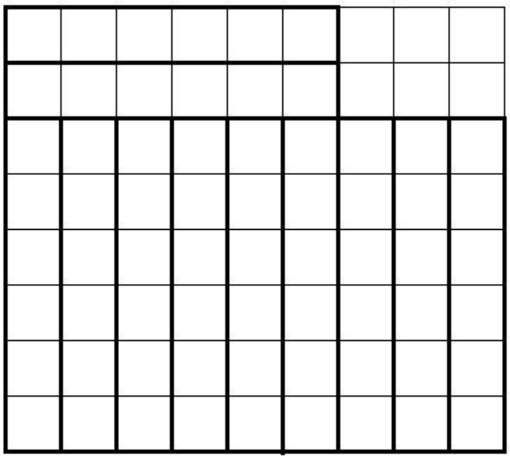


Рис. 1.

1	2	3	4	5	6	1	2	3
2	3	4	5	6	1	2	3	4
3	4	5	6	1	2	3	4	5
4	5	6	1	2	3	4	5	6
5	6	1	2	3	4	5	6	1
6	1	2	3	4	5	6	1	2
1	2	3	4	5	6	1	2	3
2	3	4	5	6	1	2	3	4

Рис. 2.

Поскольку $(8\cdot9):6=12$, из прямоугольника 8×9 вырезать более 12 прямоугольников 1×6 невозможно. На рис. 1 показано, как можно вырезать 11 прямоугольников.

Докажем, что 12 таких прямоугольников вырезать нельзя. Предположим, что можно. Тогда разрезы обязательно делались параллельно сторонам прямоугольника 8×9. Раскрасим 72 клетки этого прямоугольника в шесть цветов $(1, 2, \ldots, 6)$, как показано на рис. 2. Легко видеть, что любой прямоугольник 1×6 покрывает ровно одну «1», ровно одну «2», ... , ровно одну «6». Значит, должно быть ровно 12 клеток каждого из 6 цветов. Но это не так. Например, клеток цвета 5 имеется только 11. Противоречие.

2. Диаграмма. В кружочках на диаграмме (см. рис. 3) записаны целые числа. Числа в разных кружочках не обязательно различны. Известно, что сумма всех десяти чисел равна 2015. Суммы чисел, находящихся в вершинах каждого треугольника (с номерами 1, 2, 3, 4, 5, 6) и каждого квадрата (с номерами 7, 8) равны между собой (то есть все восемь указанных сумм попарно равны). Может ли число a быть равным а) 1970? б) 1?

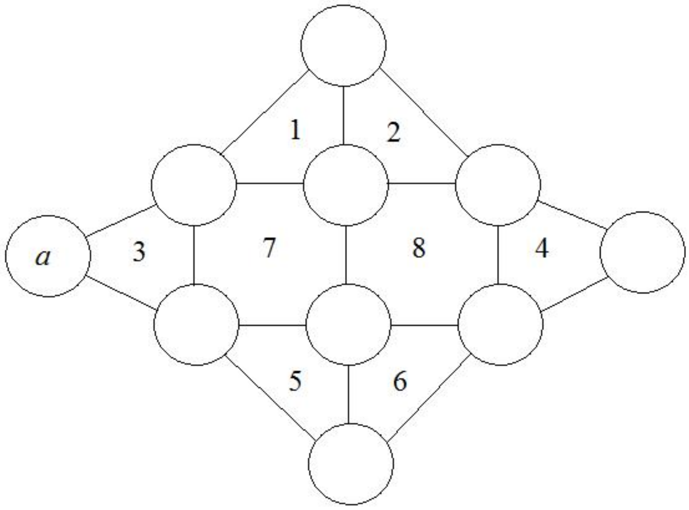


Рис. 3.

Ответ: а) да; б) нет.

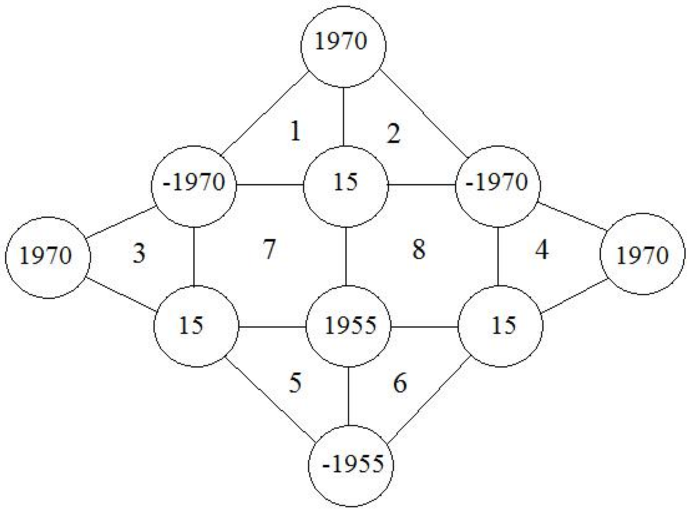


Рис. 4.

Решение.

а) пример расстановки 10 чисел см. на рис. 4;

б) обозначим сумму чисел, стоящих в вершинах треугольника, через S. Поскольку сумма всех 10 чисел равна сумме чисел, стоящих в вершинах треугольников с номерами 1, 4, 5 и числа a, имеем $2015 = 3 \cdot S + 1$, откуда $3 \cdot S = 2014$. Но 2014 на делится на 3. Противоречие.

3. Сон Буратино. Буратино снилось, будто бы у него три мешка денег: мешок с сотней золотых монет, мешок с двумя сотнями серебряных монет и мешок с тремя сотнями медных монет. Лиса Алиса предложила Буратино сыграть с Котом Базилио в игру по следующим правилам. За один ход разрешается взять себе любое количество монет из одного или двух мешков (но не из трех!). Ходы делаются по очереди, первым начинает Буратино. Выигрывает тот, кто возьмет последнюю монету. Сможет ли Буратино обыграть Кота Базилио? (Если “да”, то как именно? Если “нет”, то почему?)

Ответ: да, Буратино сможет обыграть Кота Базилио.

Решение. Первым ходом Буратино необходимо взять 100 серебряных и 200 медных монет. При этом количества монет в каждом мешке станут равными. И после каждого хода Кота Базилио он так же старательно уравнивает количества монет во всех трех мешках.

Ясно, что своим ходом Кот Базилио сравнять числа монет не сможет (поскольку не может брать монеты сразу из трех мешков). С другой стороны, Буратино своим ходом сравнять количества монет всегда сможет. Для этого надо брать (лишние) монеты из тех мешков (одного или двух), в которых в текущий момент находится НЕ минимальное количество монет.

Так как общее число монет конечно, игра обязательно закончится. Последний ход, после которого во всех мешках останется число монет, раное нулю, сделает обязательно Буратино.

Замечание. Данная стратегия единственно верная. Если после какого-либо хода Буратино число монет во всех мешках не будет одинаковым, коварный Кот Базилио тут же уравняет все мешки и воспользуется выигрышной стратегией, украденной у Буратино.

4. Ребус. Сколько решений имеет ребус (одинаковые буквы соответствуют одинаковым цифрам, разные — разным): ПО $\cdot$ (Л $+$ Е $+$ Т) $=$ ВМК, если известно, что ПО $= 45$?

Ответ: 18.

Решение.

Поскольку К $\neq$ О $= 5$, а ВМК делится на 5, то К $= 0$.

Кроме того, ВМК делится на 9, значит, сумма цифр (В $+$ М) делится на 9, причем цифры В и М разные. Поэтому В$+$М $= 9$.

С учетом того, что цифры 4, 5 и 0 уже заняты, имеем

$$\text{ВМК} \in \{180, 270, 360, 630, 720, 810\}.$$

Для каждого из этих случаев пытаемся составить сумму (Л $+$ Е $+$ Т), равную ВМК : 45, из цифр, отличных от цифр числа ВМК и числа ПО.

Легко убедиться, что это возможно только в двух последних случаях: Л $+$ Е $+$ Т $= 16 = 1 + 6 + 9$ или Л $+$ Е $+$ Т $= 18 = 2 + 7 + 9 = 3 + 6 + 9$.

В итоге имеем 18 решений. Все эти решения получаются из трех решений

$$45 \cdot (1 + 6 + 9) = 720;$$
$$45 \cdot (2 + 7 + 9) = 810;$$
$$45 \cdot (3 + 6 + 9) = 810$$

всевозможными перестановками слагаемых.

5. Дедуктивные выводы. Однажды мистер Шерлок Холмс сделал следующие наблюдения:

1) люди, каждое утро делающие зарядку, никогда не болеют;

2) не все добрые люди имеют дома домашних животных;

3) все спортсмены обязательно начинают свой день с зарядки;

4) у владельцев кошек часто развивается аллергия;

5) злых спортсменов не бывает.

Может ли знаменитый детектив на основе этих данных сделать дедуктивное заключение о том, что: а) хозяева кошек нередко не являются спортсменами, б) существуют спортсмены, у которых нет домашних животных (считая, что аллергия — это болезнь)?

Ответ: а) да; б) нет.

Решение.

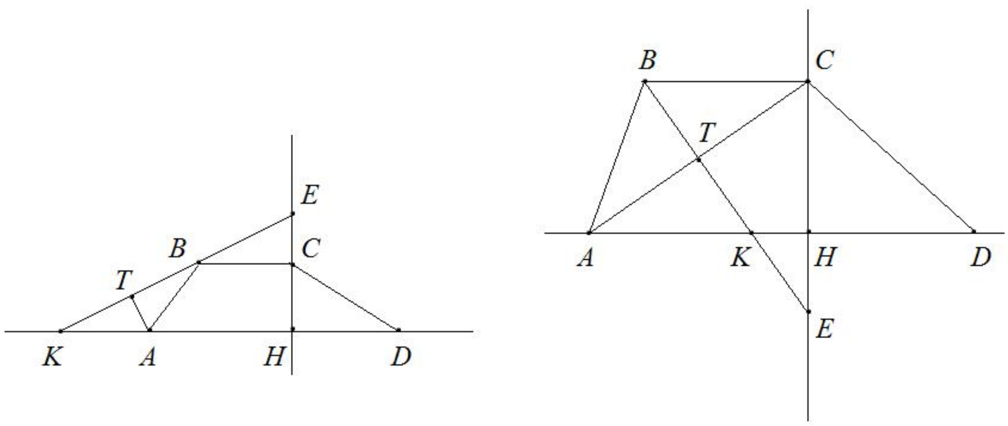
а) поскольку все спортсмены обязательно начинают свой день с зарядки (3), а люди, каждое утро делающие зарядку, никогда не болеют (1), можно сделать вывод, что все спортсмены никогда не болеют.

Далее, у владельцев кошек часто развивается аллергия (4), значит, они болеют (т. к. аллергия — это болезнь), а значит, те хозяева кошек, которые страдают аллергией, спортсменами не являются.

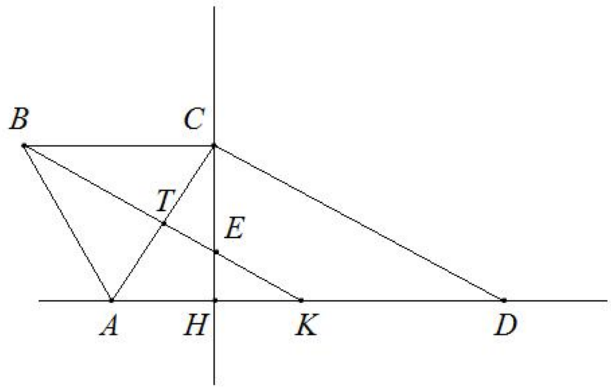
б) такой вывод сделать нельзя. Действительно, пусть все спортсмены являются добрыми людьми и имеют домашних животных, и пусть среди владельцев домашних животных число спортсменов невелико. Тогда если существуют добрые люди, которые не являются спортсменами и не имеют домашних животных, то все условия 1) – 5) могут быть выполнены.

6. Трапеция. В трапеции $ABCD$ основания BC и AD параллельны, длина основания BC равна длине боковой стороны AB. На прямой AD взяли точку K так, что $AK = AB$. Прямая, содержащая высоту CH трапеции (перпендикулярную основаниям), пересекается с прямой BK в точке E. Известно, что $EK = a$, $BE = b$. Найдите длину отрезка AB.

Ответ: $\sqrt{\frac{b\cdot|a\pm b|}{2}}$.



а) б)



в)

Рис. 5.

Решение. Три возможных принципиально различных конфигурации, соответствующих условию задачи, изображены на рис. 5 (а, б, в). Расположение точки D относительно отрезка AK на рисунках б), в) несущественно. Обозначим через x величину каждого из отрезков AB, BC, AK.

Рассмотрим рис. 5(а). Ясно, что треугольник KAB равнобедренный ($KA = AB$), а углы AKB и CBE равны. Опустим перпендикуляр AT на прямую KB. Он является медианой и высотой треугольника KAB, поэтому $KT = \frac{a-b}{2}$. Прямоугольные треугольники KTA и BCE подобны по двум углам, так что $\frac{KT}{KA} = \frac{BC}{BE}$, то есть $\frac{(a-b)/2}{x} = \frac{x}{b}$, откуда $x = \sqrt{\frac{b(a-b)}{2}}$.

Рассмотрим рис. 5(б). Пусть T — точка пересечения AC и BK. Треугольники TKA и TBC равны по стороне и двум прилежащим углам (т. к. $BC = AK$, $BC||AK$). Значит, точка T — основание медиан равнобедренных треугольников ABK и ABC, то есть прямые BK и AC перпендикулярны. Прямоугольные треугольники BCE и KTA подобны (так как углы BCE и KTA прямые, а $\angle AKT = \angle HKE = \angle CBE$). Поэтому $\frac{KT}{KA} = \frac{BC}{BE}$, то есть $\frac{(b-a)/2}{x} = \frac{x}{b}$, откуда $x = \sqrt{\frac{b(b-a)}{2}}$.

Рассмотрим рис. 5(в). Пусть T — точка пересечения AC и BK. Треугольники TKA и TBC равны по стороне и двум прилежащим углам (т. к. $BC = AK$, $BC||AK$). Значит, точка T — основание медиан равнобедренных треугольников ABK и ABC, то есть прямые BK и AC перпендикулярны. Прямоугольные треугольники BCE и KTA подобны (так как углы BCE и KTA прямые, а $\angle AKT = \angle HKE = \angle CBE$). Поэтому $\frac{KT}{KA} = \frac{BC}{BE}$, то есть $\frac{(a+b)/2}{x} = \frac{x}{b}$, откуда $x = \sqrt{\frac{b(a+b)}{2}}$.

7. Уравнение. Решите уравнение в целых числах (то есть найдите все пары целых чисел (x, y), обращающие уравнение в верное равенство):

$$2x^2 - 6xy + 5y^2 - 4x + 4y = 21.$$

Ответ: $\{(14;7),(-6;-3),(9;3),(-7;-5),(15;9),(-1;1),(6;1),(-6;-5),(14;9),(2;3),(-1;-3),(9;7)\}$.

Решение.

I способ.

Легко проверить, что уравнение можно переписать в виде

$$(x-2y)^2+(x-y-2)^2=25.$$

Учитывая, что каждый из квадратов неотрицателен и не превосходит 25, конечным перебором убеждаемся, что подходят только 4 случая:

$0+25=25$, $9+16=25$, $16+9=25$ и $25+0=25$.

Далее конечный перебор:

$$\begin{cases} x-2y &= 0 \\ x-y-2 &= \pm 5 \end{cases}$$

или

$$\begin{cases} x-2y &= \pm 3 \\ x-y-2 &= \pm 4 \end{cases}$$

или

$$\begin{cases} x-2y &= \pm 4 \\ x-y-2 &= \pm 3 \end{cases}$$

или

$$\begin{cases} x-2y &= \pm 5 \\ x-y-2 &= 0 \end{cases}$$

Решая системы, получаем ответ.

II способ.

Просто считаем, что y — параметр, тогда имеем квадратное уравнение относительно x:

$$2x^2-2(3y+2)x+(5y^2+4y-21)=0$$

При этом квадратом целого числа должно быть выражение

$D/4=(3y+2)^2-2(5y^2+4y-21)=50-(y-2)^2=n^2, n\geq 0$, т. е. $(y-2)^2+n^2=50$. Далее с помощью несложного (и конечного) перебора получаем ответ.

Задачи на программирование

8. Псевдослучайное квадратное уравнение. Вывести квадратное уравнение со случайными коэффициентами в виде: `ax^2+bx+c=0`. Необходимо соблюдать следующие требования:

1. Уравнение должно иметь ровно два различных действительных корня.

2. Коэффициенты a, b, c должны быть целочисленными, a и c по модулю не должны превосходить 99, а b — 400.

3. Алгоритм может выдать любой допустимый набор a, b и c, а не только некоторые из них (в предположении, что стандартный датчик псевдослучайных чисел может выдать число из интервала (0; 1) с пятью произвольно заданными знаками после запятой).

4. Датчик псевдослучайных чисел разрешается использовать до вывода квадратного уравнения не более шести раз.

Пример вывода: `40x^2-139x+34=0`.

Пояснения. Придумаем совершенно случайные значения a, b и c из соответствующих диапазонов. Полученные значения могут оказаться подходящими (дискриминант положителен); в этом случае ответ готов.

Допустим, значения не подходят. Затрачено только три из разрешённых шести вызовов случайной функции, но это разрешение в задаче играет отвлекающую роль: неподходящие значения могут получиться и со второго раза.

Следовательно, два из трёх коэффицианетов надо выбирать случайно (например, a и c), а диапазон, из которого выбирать третий (в данном случае — b), — вычислять.

Заметим, что в этом случае вероятность появления наборов с бóльшим b будет выше вероятности появления наборов с b, близким к 0.

«Чистое» решение состоит в том, чтобы сгенерировать полный список наборов (недостаток: в списке более 28 миллионов таких троек) и выбирать из него случайно.

Наконец, «чистое» решение можно модифицировать: для всех допустимых значений a и c вычислить допустимые диапазоны b, таким образом перенумеровав все 28 с лишним миллионов допустимых троек (a, b, c), а затем случайно выбрать одну. Ниже приводятся тексты программ на языке Python.

```
import random
from math import *

A,B,C = 99,400,99

# "Неравномерное» решение"
a = random.randint(-A,A-1)
if a>=0: a+=1                        # случайное ненулевое a
c = random.randint(-C,C)             # случайное c
if a*c < 0:                          # b может быть любым
    b = random.randint(-B,B)
elif c == 0:                         # b может быть любым, кроме 0
    b = random.randint(-B,B-1)
    if b >= 0: b+=1
else:                                # b надо вычислять
    z = 1+int(sqrt(4*a*c))           # z - минимальное значение |b|
```

```
    b = random.randint(-B,B-2*z)
    if b>-z: b+=2*z-1     # b из диапазонов [-B,-z],[z,B]

print "{}x^2{:+}x{:+} = 0".format(a,b,c)

# "Равномерное" решение
def valid(a,c):
    '''Сколько допустимых коэффициентов b возможно
    при данном значении a и c'''
    if a == 0: return 0
    if a*c < 0: return B*2+1
    if a*c == 0: return B*2
    return B*2-2*int(sqrt(4*a*c))

def locate(q, Diap):
    '''Вычислить q-ю тройку a,b,c среди диапазонов'''
    z = 0
    for a,c,x in Diap:
        # Выбор подходящей пары a,c
        if z+x-q > 0:
            # Выбор b
            if q-z < x/2:
                b = q-z-B
            else:
                b = q-z+B-x+1
            return a,b,c
        z += x

# Список "диапазонов": пары a, c и количество допустимых b
Diap = [(a,c,valid(a,c))
for a in xrange(-A,A+1)
for c in xrange(-C,C+1)]
# Сколько допустимых троек всего
Q = sum(l[2] for l in Diap)
# Случайный номер тройки
q = random.randrange(Q)
a,b,c = locate(q, Diap)
print "{}x^2{:+}x{:+}=0".format(a,b,c)
```

9. Скрытая подстрока. Ввести строку S (длины l, не превосходящей 100) и на следующей строке — слово W (длины k, не превосходящей 10). Посчитать, сколькими способами можно найти W, «спрятанное» в S. Иными словами, указать, сколько существует различных наборов натуральных чисел $i_1, i_2, \ldots, i_k$ таких, что $1 \le i_1 < i_2 < \cdots < i_k \le l$ и слово $S[i_1]S[i_2]\ldots S[i_k]$ — это W.

Пример. Ввод:

`ahomiomoto`

`homo`

Вывод: 7

Пояснения. Раз в зачаче спрашивается «сколько», а не «какие», для её решения не нужно перечистять все $i_1, \ldots, i_k$, достаточно оценить количество способов их построения.

Рекурсивный спуск функции `wwount(слово,строка)`, которая выдаёт решение задачи, будет иметь следующий вид:

`wwount(слово,строка)` $= \sum$ `wwount(слово[1:],строка[i+1:])`

(сумма берется по всем таким i, что `строка[i]==слово[0]`

Например,

```
wwount("хрю","ахxрюхр") =
= wwount("рю","хрюхр") + wwount("рю","рюхр") + wwount("рю","р").
```

Глубина рекурсии подобной функции вполне приемлема: она не будет превышать длину слова.

Алгоритм можно модифицировать для работы с большой строкой (если при этом нежелательно её дублирование в рекурсивном спуске), передавая начальный индекс поиска.

Наконец, при действительно больших строках и большом количестве «скрытых» слов можно пожертвовать памятью и заранее вычислять результаты таких промежуточных поисков: это существенно понизит вычислительную сложность решения. Ниже приводятся тексты программ на языке Python.

```
S=raw_input()
W=raw_input()


# Рекурсивный вариант
def wwount(w,b=0):
    '''Сколько слов w «скрыто» в строке S, начиная с позиции b
      (по умолчанию b=0)'''
    # Слово из одной буквы: ответ есть количество
    # таких букв в конце строки
    if len(w) == 1: return S.count(w,b)
    ret = 0
    # Поиск по всем оставшимся буквам строки, кроме последних
    for i in xrange(b, len(S)-len(w)+1):
```

```
        if S[i] == w[0]:
            ret += wwount(w[1:], i+1)
    return ret

# Более эффективный вариант
# с хранением промежуточных результатов
def wruler(w,s):
    '''Сколько слов w «скрыто» в строке s'''
    # Количество вариантов для слова длиной ww;
    # начальное значение для длины 0 - 1
    H=[1]*(len(s))
    # По всем буквам слова
    for ww in xrange(len(w)):
        # Количество вариантов для слова длиной ww+1
        NH=[0]*(len(s))
        # По всем допустимым буквам строки
        for sc in xrange(ww, len(s)-len(w)+ww+1):
            # Найдено ещё одно продолжение слова
            if s[sc] == w[ww]:
                # Это продолжение для всех вариантов
                # слова длиной ww
                NH[sc] = NH[sc-1]+H[sc-1]
            else:
                # Количество вариантов не меняется
                NH[sc] = NH[sc-1]
        H = NH
    return NH[-1]

print wwount(W)
print wruler(W,S)
```

Олимпиада школьников по прикладной математике и информатике факультета вычислительной математики и кибернетики Московского государственного университета имени М. В. Ломоносова

Очный тур (4 апреля 2015 года). 10 класс

Задачи по математике

1. Разрезание прямоугольника. Какое наибольшее количество прямоугольников 1×6 можно вырезать из прямоугольника 8×9? Ответ обосновать.

Ответ: 11.

Решение.

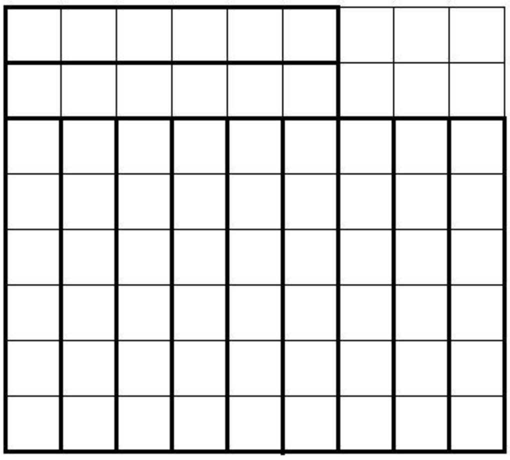


Рис. 1.

1	2	3	4	5	6	1	2	3
2	3	4	5	6	1	2	3	4
3	4	5	6	1	2	3	4	5
4	5	6	1	2	3	4	5	6
5	6	1	2	3	4	5	6	1
6	1	2	3	4	5	6	1	2
1	2	3	4	5	6	1	2	3
2	3	4	5	6	1	2	3	4

Рис. 2.

Поскольку $(8\cdot 9) : 6 = 12$, из прямоугольника 8×9 вырезать более 12 прямоугольников 1×6 невозможно. На рис. 1 показано, как можно вырезать 11 прямоугольников.

Докажем, что 12 таких прямоугольников вырезать нельзя. Предположим, что можно. Тогда разрезы обязательно делались параллельно сторонам прямоугольника 8×9. Раскрасим 72 клетки этого прямоугольника в шесть цветов $(1, 2, \ldots, 6)$, как показано на рис. 2. Легко видеть, что любой прямоугольник 1×6 покрывает ровно одну «1», ровно одну «2», ... , ровно одну «6». Значит, должно быть ровно 12 клеток каждого из 6 цветов. Но это не так. Например, клеток цвета 5 имеется только 11. Противоречие.

2. Диаграмма. В кружочках на диаграмме (см. рис. 3) записаны целые числа так, что суммы чисел, находящихся в вершинах каждого треугольника (с номерами 1, 2, 3, 4, 5, 6) и каждого квадрата (с номерами 7, 8) равны между собой (то есть все восемь указанных сумм попарно равны). Известно, что $a = b + c$. Доказать, что сумма всех десяти чисел в кружочках делится на 7 без остатка.

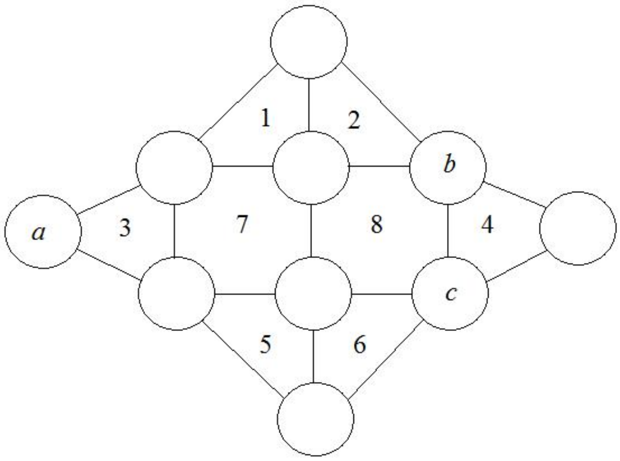


Рис. 3.

Решение. Обозначим сумму чисел, стоящих в вершинах треугольника с номером n через S_n, а сумму всех 10 чисел — через S. Заметим, что по условию все S_n равны. Пусть число, стоящее в самом правом кружочке (в оставшейся вершине треугольника 4) есть a'. Имеем

$$S = a + S_1 + S_4 + S_5 = S_2 + S_3 + S_6 + a',$$

значит $a' = a$ и $S_n = a + b + c = 2a$ при всех n.

Поэтому из написанного выше

$$S = a + S_1 + S_4 + S_5 = a + 3 \cdot 2a = 7a,$$

что и требовалось доказать.

3. Сон Буратино. Буратино снилось, будто бы у него три мешка денег: мешок с сотней золотых монет, мешок с двумя сотнями серебряных монет и мешок с тремя сотнями медных монет. Лиса Алиса предложила Буратино сыграть с Котом Базилио в игру по следующим правилам. За один ход разрешается взять себе любое количество монет из одного или двух мешков (но не из трех!). Ходы делаются по очереди, первым начинает Буратино. Выигрывает тот, кто возьмет последнюю монету. Сможет ли Буратино обыграть Кота Базилио? (Если "да", то как именно? Если "нет", то почему?)

Ответ: да, Буратино сможет обыграть Кота Базилио.

Решение. Первым ходом Буратино необходимо взять 100 серебряных и 200 медных монет. При этом количества монет в каждом мешке станут равными. И после каждого хода Кота Базилио он так же старательно уравнивает количества монет во всех трех мешках.

Ясно, что своим ходом Кот Базилио сравнять числа монет не сможет (поскольку не может брать монеты сразу из трех мешков). С другой стороны, Буратино своим ходом сравнять количества монет всегда сможет. Для этого надо брать (лишние) монеты из тех мешков (одного или двух), в которых в текущий момент находится НЕ минимальное количество монет.

Так как общее число монет конечно, игра обязательно закончится. Последний ход, после которого во всех мешках останется число монет, раное нулю, сделает обязательно Буратино.

Замечание. Данная стратегия единственно верная. Если после какого-либо хода Буратино число монет во всех мешках не будет одинаковым, коварный Кот Базилио тут же уравняет все мешки и воспользуется выигрышной стратегией, украденной у Буратино.

4. Углы треугольника. Пусть α, β, γ — выраженные в радианах величины трех углов треугольника (то есть α, β, γ положительны, а их сумма равна π). Доказать, что если

$$\frac{\alpha}{\sin\beta} = \frac{\beta}{\sin\gamma} = \frac{\gamma}{\sin\alpha},$$

то $\alpha^2+\beta^2+\gamma^2=\frac{\pi^2}{3}$.

Решение. Пусть a, b, c — стороны треугольника, а α, β, γ — противолежащие им углы соответственно. Тогда из условия и теоремы синусов

$$\frac{b}{\sin\beta} = \frac{c}{\sin\gamma} = \frac{a}{\sin\alpha}$$

вытекают равенства

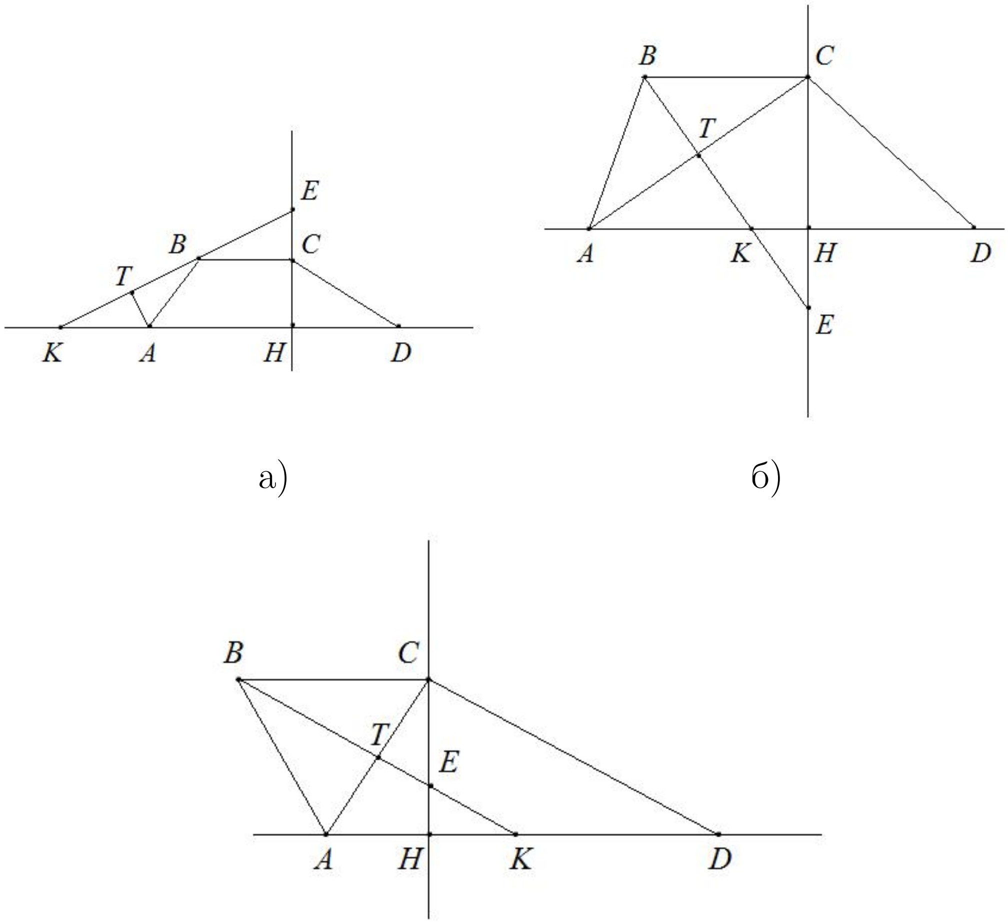
$$\frac{\sin\alpha}{\sin\beta} = \frac{\gamma}{\alpha} = \frac{a}{b}, \tag{1}$$

$$\frac{\sin\alpha}{\sin\gamma} = \frac{\gamma}{\beta} = \frac{a}{c}. \tag{2}$$

Пусть $\alpha \geq \beta$. Тогда $a \geq b$ (против большего угла лежит бóльшая сторона), следовательно, в силу (1), $\gamma \geq \alpha$, из чего вытекает, что $c \geq a$ (против большего угла лежит бóльшая сторона). Значит, $\beta \geq \gamma$ (в силу (2)), и, наконец, $b \geq c$. Получили цепочку неравенств $a \geq b \geq c \geq a$, то есть треугольник равносторонний, а все углы в нем равны друг другу и $\frac{\pi}{3}$. Отсюда сразу получается требуемое. Рассуждения в предположении $\alpha \leq \beta$ аналогичны.

5. Трапеция. В трапеции $ABCD$ основания BC и AD параллельны, длина основания BC равна длине боковой стороны AB. На прямой AD взяли точку K так, что $AK = AB$. Прямая, содержащая высоту CH трапеции (перпендикулярную основаниям), пересекается с прямой BK в точке E. Известно, что $EK = a$, $BE = b$. Найдите длину отрезка AB.

Ответ: $\sqrt{\frac{b\cdot|a\pm b|}{2}}$.



а) б)

в)

Рис. 4.

Решение. Три возможных принципиально различных конфигурации, соответствующих условию задачи, изображены на рис. 4 (а, б, в). Расположение точки D относительно отрезка AK на рисунках б), в) несущественно. Обозначим через x величину каждого из отрезков AB, BC, AK.

Рассмотрим рис. 4(а). Ясно, что треугольник KAB равнобедренный ($KA = AB$), а углы AKB и CBE равны. Опустим перпендикуляр AT на прямую KB. Он является медианой и высотой треугольника KAB, поэтому $KT = \frac{a-b}{2}$. Прямоугольные треугольники KTA и BCE подобны по двум углам, так что $\frac{KT}{KA} = \frac{BC}{BE}$, то есть $\frac{(a-b)/2}{x} = \frac{x}{b}$, откуда $x = \sqrt{\frac{b(a-b)}{2}}$.

Рассмотрим рис. 4(б). Пусть T — точка пересечения AC и BK. Треугольники TKA и TBC равны по стороне и двум прилежащим углам (т. к. $BC = AK$, $BC||AK$). Значит, точка T — основание медиан равнобедренных треугольников ABK и ABC, то есть прямые BK и AC перпендикулярны. Прямоугольные треугольники BCE и KTA подобны (так как углы BCE и KTA прямые, а $\angle AKT = \angle HKE = \angle CBE$). Поэтому $\frac{KT}{KA} = \frac{BC}{BE}$, то есть $\frac{(b-a)/2}{x} = \frac{x}{b}$, откуда $x = \sqrt{\frac{b(b-a)}{2}}$.

Рассмотрим рис. 4(в). Пусть T — точка пересечения AC и BK. Треугольники TKA и TBC равны по стороне и двум прилежащим углам (т. к. $BC = AK$, $BC||AK$). Значит, точка T — основание медиан равнобедренных треугольников ABK и ABC, то есть прямые BK и AC перпендикулярны. Прямоугольные треугольники BCE и KTA подобны (так как углы BCE и KTA прямые, а $\angle AKT = \angle HKE = \angle CBE$). Поэтому $\frac{KT}{KA} = \frac{BC}{BE}$, то есть $\frac{(a+b)/2}{x} = \frac{x}{b}$, откуда $x = \sqrt{\frac{b(a+b)}{2}}$.

6. Уравнение. Решите уравнение в целых числах (то есть найдите все пары целых чисел (x, y), обращающие уравнение в верное равенство):

$$2x^2 - 6xy + 5y^2 - 4x + 4y = 21.$$

Ответ: $\{(14;7), (-6;-3), (9;3), (-7;-5), (15;9), (-1;1), (6;1), (-6;-5), (14;9), (2;3), (-1;-3), (9;7)\}$.

Решение.

I способ.

Легко проверить, что уравнение можно переписать в виде

$$(x-2y)^2 + (x-y-2)^2 = 25.$$

Учитывая, что каждый из квадратов неотрицателен и не превосходит 25, конечным перебором убеждаемся, что подходят только 4 случая:

$0 + 25 = 25$, $9 + 16 = 25$, $16 + 9 = 25$ и $25 + 0 = 25$.

Далее конечный перебор:

$$\left\{\begin{array}{rcl} x-2y &=& 0 \\ x-y-2 &=& \pm 5 \end{array}\right.$$

или

$$\left\{\begin{array}{rcl} x-2y &=& \pm 3 \\ x-y-2 &=& \pm 4 \end{array}\right.$$

или

$$\left\{\begin{array}{rcl} x-2y &=& \pm 4 \\ x-y-2 &=& \pm 3 \end{array}\right.$$

или

$$\left\{\begin{array}{rcl} x-2y &=& \pm 5 \\ x-y-2 &=& 0 \end{array}\right.$$

Решая системы, получаем ответ.

II способ.

Просто считаем, что y — параметр, тогда имеем квадратное уравнение относительно x:

$$2x^2 - 2(3y+2)x + (5y^2 + 4y - 21) = 0$$

При этом квадратом целого числа должно быть выражение

$D/4 = (3y+2)^2 - 2(5y^2+4y-21) = 50 - (y-2)^2 = n^2, n \geq 0$, т. е. $(y-2)^2 + n^2 = 50$. Далее с помощью несложного (и конечного) перебора получаем ответ.

7. Многочлен с целыми коэффициентами. Существует ли такой многочлен с целыми коэффициентами, у которого не менее 2015 попарно различных целых корней, а сумма всех коэффициентов равна а) 0, б) 2^{2015}, в) 6^{31744}?

Ответ. а) Да. б) Нет. в) Да.

Решение. Пусть имеется многочлен $P(x)$ степени n с целыми коэффициентами:

$$P(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \cdots + a_{n-1}x + a_n,$$

где $a_0 \neq 0$. Поскольку при любом натуральном m имеет место равенство

$$x^m - x_0^m = (x - x_0)(x^{m-1} + x^{m-2}x_0 + x^{m-3}x_0^2 + \cdots + x_0^{m-2}x + x_0^{m-1}),$$

то для любого целого корня x_0 многочлена $P(x)$ (то есть такого числа, что $P(x_0) = 0$) справедливо равенство

$$P(x) = P(x) - P(x_0) = (x - x_0) \cdot Q_0(x),$$

где $Q_0(x)$ — многочлен степени $n-1$ с целыми коэффициентами. Следовательно, всякий многочлен $P(x)$ с целыми коэффициентами, у которого имеется 2015 попарно различных целых корней x_1, x_2, ..., x_{2015}, можно представить в виде

$$P(x) = (x - x_1)(x - x_2) \times \cdots \times (x - x_{2015}) \cdot Q(x),$$

где $Q(x)$ — многочлен с целыми коэффициентами. Заметим, что сумма S всех коэффициентов $P(x)$ равна $P(1)$, поэтому для выполнения равенства $S = 0$ достаточно, чтобы среди различных целых корней $P(x)$ была бы единица, что обеспечивает положительный ответ в пункте а).

Для исследования задачи пункта б) рассмотрим получающиеся из приведенных рассуждений равенства:

$$P(1) = (1 - x_1)(1 - x_2) \times \cdots \times (1 - x_{2015}) \cdot Q(1) = 2^{2015}.$$

В этом равенстве каждый множитель — отличное от нуля целое число, абсолютная величина которого есть степень двойки, при этом первые 2015 множителей попарно различны. Ясно, что абсолютная величина произведения 2015 таких попарно различных целых чисел не меньше, чем

$$A = 2^0 \cdot 2^0 \cdot 2^1 \cdot 2^1 \cdot 2^2 \cdot 2^2 \times \cdots \times 2^{1007} \cdot 2^{1007} \cdot 2^{1008} = 2^{1007 \cdot 1008 + 1008} = 2^{1016064} > 2^{2015},$$

(повторы последовательных множителей связаны с возможностью выбирать в качестве различных множителей целые число и им противоположные), так что ответ в пункте б) отрицательный.

Для исследования задачи пункта в) рассмотрим многочлен $P(x)$, корнями которого являются 2048 чисел: $1 \pm 2^i \cdot 3^j$, где i и j независимо пробегают множество числел $\{0, 1, 2, \dots, 31\}$. Тогда

$$P(1) = (2^0 \cdot 3^0)^2 \cdot (2^0 \cdot 3^1)^2 \times \dots \times (2^0 \cdot 3^{31})^2 \cdot (2^1 \cdot 3^0)^2 \times \dots \times (2^{31} \cdot 3^{31})^2 = 6^{31 \cdot 32^2} = 6^{31744},$$

и ответ в пункте в) положительный.

Задачи на программирование

8. Скрытая подстрока. Ввести строку S (длины l, не превосходящей 100) и на следующей строке — слово W (длины k, не превосходящей 10). Посчитать, сколькими способами можно найти W, «спрятанное» в S. Иными словами, указать, сколько существует различных наборов натуральных чисел $i_1, i_2, \dots, i_k$ таких, что $1 \le i_1 < i_2 < \dots < i_k \le l$ и слово $S[i_1]S[i_2]\dots S[i_k]$ — это W.

Пример. Ввод:

`ahomiomoto`

`homo`

Вывод: `7`

Пояснения. Раз в зачаче спрашивается «сколько», а не «какие», для её решения не нужно перечистять все $i_1, \dots, i_k$, достаточно оценить количество способов их построения.

Рекурсивный спуск функции `wwount(слово,строка)`, которая выдаёт решение задачи, будет иметь следующий вид:

`wwount(слово,строка)` $= \sum$ `wwount(слово[1:],строка[i+1:])`

(сумма берется по всем таким i, что `строка[i]==слово[0]`

Например,

```
wwount("хрю","ахxрюxр") =
= wwount("рю","хрюхр") + wwount("рю","рюхр") + wwount("рю","р").
```

Глубина рекурсии подобной функции вполне приемлема: она не будет превышать длину слова.

Алгоритм можно модифицировать для работы с большой строкой (если при этом нежелательно её дублирование в рекурсивном спуске), передавая начальный индекс поиска.

Наконец, при действительно больших строках и большом количестве «скрытых» слов можно пожертвовать памятью и заранее вычислять результаты таких промежуточных поисков: это существенно понизит вычислительную сложность решения. Ниже приводятся тексты программ на языке Python.

```
S=raw_input()
W=raw_input()

# Рекурсивный вариант
def wwount(w,b=0):
    '''Сколько слов w «скрыто» в строке S, начиная с позиции b
      (по умолчанию b=0)'''
    # Слово из одной буквы: ответ есть количество
    # таких букв в конце строки
    if len(w) == 1: return S.count(w,b)
    ret = 0
    # Поиск по всем оставшимся буквам строки, кроме последних
    for i in xrange(b, len(S)-len(w)+1):
        if S[i] == w[0]:
            ret += wwount(w[1:], i+1)
    return ret

# Более эффективный вариант
# с хранением промежуточных результатов
def wruler(w,s):
    '''Сколько слов w «скрыто» в строке s'''
    # Количество вариантов для слова длиной ww;
    # начальное значение для длины 0 - 1
    H=[1]*(len(s))
    # По всем буквам слова
    for ww in xrange(len(w)):
        # Количество вариантов для слова длиной ww+1
        NH=[0]*(len(s))
        # По всем допустимым буквам строки
        for sc in xrange(ww, len(s)-len(w)+ww+1):
            # Найдено ещё одно продолжение слова
            if s[sc] == w[ww]:
                # Это продолжение для всех вариантов
                # слова длиной ww
                NH[sc] = NH[sc-1]+H[sc-1]
            else:
                # Количество вариантов не меняется
                NH[sc] = NH[sc-1]
        H = NH
    return NH[-1]
```

```
print wwount(W)
print wruler(W,S)
```

9. Трисинусатая последовательность. Целое число A называется N-*трисинусатым по отношению к трёхзначному числу* B (лежащему в диапазоне от 000 до 999), если среди первых N цифр после запятой в числе $\sin A$ встречается число B (аргумент синуса можно трактовать как угол, измеряемый в радианах). Последовательность $A[1]$, $A[2]$, …, $A[k]$ называется N-*трисинусатой*, если для любого натурального i от 1 до $k-1$ число $A[i]$ является N-трисинусатым по отношению к числу $A[i+1]$. Ввести трёхзначное число B и вычислить такое минимальное N, при котором существует N-трисинусатая последовательность с первым членом B и последним членом 000.

Пояснения. N-трисинусатая последовательность — это не что иное, как путь в лабиринте, «комнаты» которого имеют имена от 000 до 999, а доступность одной вершины из другой пределяется свойством N-трисинусатости (при $N = 3$ у каждой комнаты, имеется только один сосед, возможно — та же самая комната, при $N = 4$ — два и т. д.). Один из вариантов решения задачи — сформировать такой лабиринт для $N = 3$ и попытаться пройти его, если не удастся — проделать то же самое для для $N = 4$, и так до тех пор, пока при очередном N лабиринт не окажется проходимым. Это N и будет ответом.

В простроении лабиринта немалую роль играет выделение троек цифр — имён комнат. Во многих языках прогрмамирования это удобно делать путём преобразования чисел в строки. В комментариях к решению строки вида “123” также называются числами.

Ниже приводbтся текст программы на языке Python.

```
from math import *
import sys

B = "{:03}".format(input())

def CloseMap(Map,st,fin):
    '''Алгоритм «заливки» для выяснения проходимости от st к fin'''
    # Множество чисел, доступных из st; числа,
    # доступные на очередном шаге
    S, A = set(), set([st])
    # Пока на очередном шаге доступны новые числа
    while A:
        # Добавляем их в общее множество
        S |= A
```

```
        # Новые числа - все, доступные из текущих, но не те,
        # что уже в множестве
        A = set().union(*[Map[a] for a in A])-S
        # Добрались до fin - путь есть!
        if fin in A:
            return True
    # Куда ни иди, новых чисел не прибавляется, пути нет
    return False

def TriSin(i,f,N):
    '''Подсчитать множество чисел, N-трисинусатых числу i'''
    return {f.format(sin(i)).split('.')[1][j:j+3]
    for j in xrange(0,N-2)}

def MinTriSin(st,en):
    N=3
    while True:
        # Строка форматирования вида "{:0N.0Nf}"
        f = "{:0"+str(N)+".0"+str(N)+"f}"
        # Карта вида "число":{множество N-трисинусатых ему}
        Map = {"{:03}".format(i):TriSin(i,f,N)
        for i in xrange(1000)}
        if CloseMap(Map, st, en):
            return N
        N+=1

print MinTriSin(B, '000')
```