

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

Очный тур. Решения задач

26 апреля 2014 года

8 класс

Задачи по математике

Задача по математике считалась решенной, если правильный ответ был получен в результате строгих и обоснованных рассуждений.

1. Дважды

Двоичная запись десятичного числа оканчивается на это десятичное число, записанное дважды. Найти два таких десятичных числа.

Решение. Возможными ответами являются числа 10 и 100:

$$10_{10} = 1010_2, \quad 100_{10} = 1100100_2.$$

2. Далёкое и близкое

Найти минимальную и максимальную разность между двумя натуральными числами, суммы цифр которых делятся на 8, а между ними таких чисел нет.

Решение. Минимальная разность равна 1 (меньше быть не может, а пример двух подряд идущих натуральных чисел с кратными восьми суммами цифр даётся парой чисел 79, 80).

Обозначим через Δ максимальную разность между двумя натуральными числами, суммы цифр которых делятся на 8, а между ними таких чисел нет. Конечность Δ очевидна, так как среди 10 последовательных чисел каждого десятка имеется число с кратной восьми суммой цифр. Отсюда следует также, что если Δ достигается на числах a и b , $a < b$, то a и b лежат в последовательных десятках. Заметим: a не может оканчиваться на цифры 0 и 1, а b не может оканчиваться на цифры 8 и 9. Значит, $\Delta \leq 15$, причем равенство возможно лишь для таких a и b из двух последовательных десятков, что a оканчивается на 2, а b оканчивается на 7. Пример чисел 9999992 и 10000007 показывает, что $\Delta = 15$.

3. Неизвестный квадрат

Дано число, состоящее из 2014 единиц (других цифр нет). Доказать, что к этому числу можно приписать справа 2015 цифр так, что получится точный квадрат.

Решение. Для любого числа a в отрезке $[a; a + 1]$ найдется целое число, так как длина этого отрезка равна единице. Поэтому для того, чтобы доказать требуемое в

задаче, достаточно показать, что не меньше единицы длина отрезка, заключенного между квадратными корнями из наименьшего и наибольшего 4029-значных десятичных чисел, начинающихся с 2014 единиц. То есть, достаточно доказать, что

$$\underbrace{11 \dots 11}_{2014} \underbrace{99 \dots 99}_{2015}^{0,5} - \underbrace{11 \dots 11}_{2014} \underbrace{00 \dots 00}_{2015}^{0,5} \geq 1 \quad \Leftrightarrow$$

$$\sqrt{\left(\frac{10^{2014} - 1}{9} + 1\right) \cdot 10^{2015} - 1} - \sqrt{\frac{10^{2014} - 1}{9} \cdot 10^{2015}} \geq 1. \quad (1)$$

Положим

$$A = \left(\frac{10^{2014} - 1}{9} + 1\right) \cdot 10^{2015} - 1, \quad B = \frac{10^{2014} - 1}{9} \cdot 10^{2015} \quad (A > 0, \quad B > 0).$$

Так как при положительных A и B имеет место равенство $\sqrt{A} - \sqrt{B} = \frac{A - B}{\sqrt{A} + \sqrt{B}}$, а $\sqrt{A} + \sqrt{B} > 0$, то неравенство (1) равносильно неравенству

$$A - B \geq \sqrt{A} + \sqrt{B} \quad \Leftrightarrow$$

$$\left(\frac{10^{2014} - 1}{9} + 1\right) \cdot 10^{2015} - 1 - \frac{10^{2014} - 1}{9} \cdot 10^{2015} \geq$$

$$\geq \sqrt{\left(\frac{10^{2014} - 1}{9} + 1\right) \cdot 10^{2015} - 1} + \sqrt{\frac{10^{2014} - 1}{9} \cdot 10^{2015}} \quad \Leftrightarrow$$

$$10^{2015} - 1 \geq \sqrt{\left(\frac{10^{2014} - 1}{9} + 1\right) \cdot 10^{2015} - 1} + \sqrt{\frac{10^{2014} - 1}{9} \cdot 10^{2015}} \quad \Leftrightarrow$$

$$3 \cdot 10^{2015} - 3 \geq \sqrt{(10^{2014} + 8) \cdot 10^{2015} - 9} + \sqrt{(10^{2014} - 1) \cdot 10^{2015}} \quad \Leftrightarrow$$

$$C \geq D + E, \quad (2)$$

где $C = 3 \cdot 10^{2015} - 3$, $D = \sqrt{(10^{2014} + 8) \cdot 10^{2015} - 9}$, $E = \sqrt{(10^{2014} - 1) \cdot 10^{2015}}$. Но, очевидно, $D + E < 2D < 2\sqrt{(10^{2014} + 8) \cdot 10^{2015}}$, поэтому для доказательства неравенства (2) достаточно доказать такое неравенство:

$$C \geq 2\sqrt{(10^{2014} + 8) \cdot 10^{2015}}.$$

А это неравенство равносильно неравенству

$$9 \cdot 10^{4030} - 18 \cdot 10^{2015} + 9 \geq 4 \cdot 10^{4029} + 32 \cdot 10^{2015} \quad \Leftrightarrow$$

$$86 \cdot 10^{4029} + 9 \geq 50 \cdot 10^{2015}.$$

Последнее неравенство очевидно, так что требуемое доказано.

4. Сдвиг?

На доске выписаны все натуральные числа от 1 до 50. За одну операцию разрешается выбрать два числа x и y и заменить их на числа $(3x - 7y)$ и $(5x - 6y)$. Можно ли в результате конечного числа операций получить на доске все целые числа от 0 до 49?

Ответ. Нельзя.

Решение. Поскольку в результате одной операции сумма всех чисел изменяется на $7(x - 2y) : 7$, то при преобразованиях сохраняется остаток суммы всех чисел от деления на 7. Если бы из набора чисел 1, 2, ..., 50 можно было бы получить набор чисел 0, 1, ..., 49, то сумма всех чисел изменилась бы на 50, а 50 не кратно семи. Следовательно, требуемое осуществить нельзя.

5. Такие разные, а всё-таки вместе!

Разрезать шахматную доску 8×8 по границам клеток на попарно неравные части (т. е. не совпадающие при наложении, — переворачивать части разрешается, цвет клеток не учитывается) так, чтобы получилось наибольшее число частей. Максимальность числа частей обосновать.

Ответ. Максимум числа частей равен 16.

Решение. Легко видеть, что может получиться не более одной одноклеточной части, не более одной двуклеточной, двух трёхклеточных и пяти четырёхклеточных. Остальные части — не менее чем из пяти клеток каждая. Значит, общее количество частей не превосходит величины

$$1 + 1 + 2 + 5 + \frac{64 - (1 \cdot 1 + 1 \cdot 2 + 2 \cdot 3 + 5 \cdot 4)}{5} = 16.$$

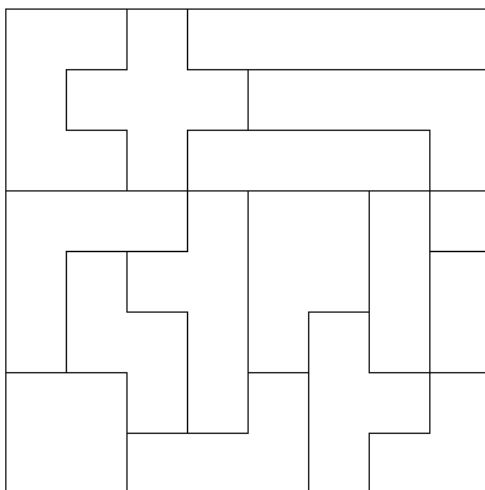


Рис. 1.

Пример разрезания на 16 частей представлен на рисунке 1.

Задачи по информатике

Во всех задачах по информатике хорошим решением считался комментированный алгоритм, позволяющий на обычном современном компьютере получить ответ при любых допустимых входных данных за относительно небольшое время (порядка минуты).

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

6. Сомножители

Вывести первые 1000 составных натуральных чисел, которые раскладываются ровно на N ($1 < N < 100$) простых сомножителей.

Входные данные: N .

Выходные данные: последовательность чисел по одному в строке.

Решение. Несмотря на «безобидные» с виду исходные данные, придётся иметь дело со сверхбольшими целыми числами: при $N = 99$ первое число ответа — 2^{99} . Теоретический

характер заданий позволяет допустить поддержку сверхбольших целых чисел в любом языке программирования так, как это сделано в языке Python. Разумеется, метод «рассмотрим все числа и выведем только те, чьё разложение на простые сомножители даёт N элементов» не подходит: слишком велик диапазон (для $N = 99$ — от 2^{99} до $2^{93} \cdot 3^5 \cdot 73$).

Метод «составим всевозможные произведения из N простых чисел, отсортируем по возрастанию и отсечём первую тысячу» приводит к сортировке K^N произведений, где K — количество простых чисел, что ещё хуже.

Можно попробовать сгенерировать ответ, взяв за основу N двоек, и каждую последующую группу сомножителей изготавливать из предыдущих, заменяя один сомножитель на следующий в таблице простых чисел. Этот метод похож на задачу о разложении числа на всевозможные слагаемые, однако главным отличием (и трудностью) в нём будет необходимость выбирать наименьшее из произведений. Так, даже для $N = 2$ в ответе сначала появится $2 \cdot 7 = 14$ (с суммой 9), а только затем — $3 \cdot 5 = 15$ (с суммой 8).

Наконец, следующее соображение поможет нам решить задачу индукцией по N . Предположим, что у нас имеется ответ для чисел, разложимых на $N - 1$ сомножителей. Допустим, что среди ответа для N сомножителей имеется число P такое, что произведение Q каких-то $N - 1$ простых сомножителей P не входит в ответ для $N - 1$. Тогда Q больше любого числа R из ответа для $N - 1$. Выберем тот простой сомножитель S числа P , который равен $S = P/Q$. Тогда, так как $P = QS$, получим, что P превосходит каждое из 1000 чисел вида RS (где R — по-прежнему любое число из ответа для $N - 1$). Следовательно, P не может входить в ответ для N . Полученное противоречие обеспечивает шаг индукции.

В решение также включена простая реализация функции, которая возвращает (и запасает в списке) n -е простое число.

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

# Простые числа. Затравка
Pr=[2,3,5,7,11,13,17,19,23]

def Prime(n):
    '''Вычислить (или взять из списка) простое число номер n'''
    global Pr
    # Если вычисленных чисел не хватает, дополним список
    while len(Pr)<=n:
        p = Pr[-1]+2
        while True:
            for d in Pr:
                if p%d == 0:
                    break
            # Если выход из цикла не по break, то число простое
            else:
                Pr.append(p)
                break
        p+=2
    return Pr[n]

def Razl(MX,N):
    '''Первые MX чисел, состоящих из N простых сомножителей'''
```

```

Prime(MX)
# Составные числа. Для N==1 это - простые числа :)
Mul = Pr[:]
# Ответ для N==k получается из ответа для k-1
for somn in xrange(N-1):
    # Ind[i]: на какое простое число мы ещё не умножали Mul[i]?
    Ind = [0]*len(Pr)
    # Первое число в ответе очевидно: 2**(k-1)*2=2**k
    New = [Pr[0]*Mul[0]]
    # Запомним его
    Ind[0] = 1
    # Наберём MX наименьших составных чисел
    while len(New)<MX:
        # Наименьшее произведение (и номер числа)
        Nx,ci = min((Mul[i]*Prime(Ind[i]),i) for i in xrange(MX))
        # В следующий раз умножим на следующее простое число
        Ind[ci]+=1
        # Если такого числа ещё не было, добавим его в ответ
        if Nx!=New[-1]: New.append(Nx)
    Mul=New
return Mul

MX = 1000
N = input()
Mul = Razl(MX,N)
for p in Mul:
    print p,

```

7. Задача о шкатулке

Ослик Иа зашел в магазин подарков и обнаружил, что в нём нет ничего, кроме ста китайских шкатулок, имеющих вид параллелепипедов. В магазине имелся список всех шкатулок с указанием точных их размеров. Все шкатулки одинаково красивы, но ручек они не имеют и унести можно только одну. Зато внутрь можно положить ещё одну (но не две сразу — поцарапаются), внутрь второй — третью и так далее. Помогите Иа унести как можно больше шкатулок. Толщиной стенок пренебречь.

Входные данные: размеры шкатулок — 100 строк по три числа через пробел в строке.

Выходные данные: номера шкатулок, которые можно вложить одна в другую.

Решение. Главная «западня» этой задачи — она маскируется под задачу о рюкзаке, но ею не является. На самом деле требуется найти глубину максимальной цепочки частично упорядоченного множества. Для этого для каждой введённой шкатулки найдём те, что в неё помещаются и те, в которую помещается она. По самым грубым оценкам такая структура данных будет содержать не больше $100 \cdot 99 \cdot 4 = 49500$ элементов. Полученный ориентированный граф не будет иметь циклов; наша задача сведётся к поиску самого длинного пути в этом графе. Для этого в полученном списке введём дополнительное поле «вложенность». На первом шаге рассмотрим шкатулки, которые нельзя ни во что положить: их вложенность будет равна 1. На следующем шаге рассмотрим все шкатулки, которые можно положить в те, что мы нашли на предыдущем шаге. Вложенность этих шкатулок будет равна номеру шага. Так как граф не имеет циклов, процесс закончится

на каком-то шаге N . Это и есть наибольшее количество вложенных шкатулок. Осталось привести ответ. Для этого возьмём любую шкатулку вложенности N . Она обязательно входит в какую-нибудь шкатулку вложенности $N - 1$, та, в свою очередь, — в шкатулку вложенности $N - 2$ и так далее до 1.

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

# Список коробок; каждый элемент имеет следующий вид:
# Ширина, высота, длина, меньшие коробки, большие коробки, длина пути
Net=[]
# Перестановки 0,1,2
R=((0,1,2),(0,2,1),(1,0,2),(1,2,0),(2,0,1),(2,1,0))
# Добавляем размеры коробок
for l in xrange(100):
    x,y,z = map(int,raw_input().split())
    less, more = [], []
    # Находим коробки, которые больше или меньше очередной
    for m in xrange(l):
        for i,j,k in R:
            if x<Net[m][i] and y<Net[m][j] and z<Net[m][k]:
                Net[m][3].append(l)
                more.append(m)
            elif x>Net[m][i] and y>Net[m][j] and z>Net[m][k]:
                less.append(m)
                Net[m][4].append(l)
    # Вложенность пока не посчитана
    Net.append([x,y,z,less,more,0])

# Посчитаем вложенность: начнём с самых больших коробок
Front=set(i for i in xrange(100) if Net[i][4] == [])
n=0
while Front:
    n+=1
    Newfront=set()
    for b in Front:
        Newfront.update(Net[b][3])
        Net[b][5]=n
    Oldfront,Front=Front,Newfront
# Итак, начнём с самой внутренней коробки (любой)
Path=[Oldfront.pop()]
while n>0:
    n-=1
    for i in Net[Path[0]][4]:
        # ...которая лежит в другой коробке (вложенность на 1 меньше)
        if Net[i][5] == n:
            Path.insert(0,i)
            break
# Так можно было вывести ответ
# for p in Path: print p,
```

```
# Но мы для красоты выведем и размеры
for p in Path:
    print "{:2}: {}?{}?{}".format(p+1,Net[p][0],Net[p][1],Net[p][2])
```

8. Индейский узник

Все постройки индейцев племени польбаи имеют треугольное основание. Казематы во дворце Великого Вождя Куа-Тотля — тоже треугольник, разделённый на 100 одинаковых треугольных комнат стенами, параллельными стенам дворца. Узник находится в самой северной комнате казематов, выход на поверхность — в наиболее юго-восточной комнате. Между всеми соседними комнатами имеются проходы, и в каждой комнате подготовлена смертельная ловушка. К счастью, узник умеет избегать некоторые из этих ловушек: у него имеется список относительно безопасных комнат. Комнаты пронумерованы от 1 до 100 по рядам с запада на восток: северная комната имеет № 1, следующий ряд — № 2, 3 (комната с вершиной на юг) и 4; третий ряд — № 5, 6, 7, 8 и 9; и т. д. (пример плана подобных, но меньших, казематов приведен на рисунке 2). Может ли узник добраться до выхода, заходя только в относительно безопасные комнаты?

Входные данные: номера безопасных комнат через пробел.

Выходные данные: «да» или «нет».

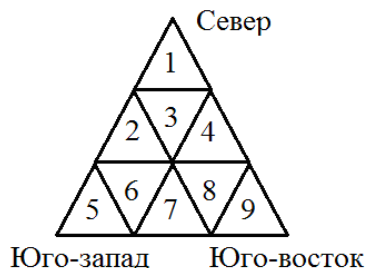


Рис. 2.

Решение. Это — простая задача обхода лабиринта. Вся премудрость в том, чтобы «справиться с управлением» в случае нестандартной связности. Несмотря на то, что число комнат в треугольном лабиринте — всегда точный квадрат, располагать комнаты в матрице 10×10 очень неудобно.

Номера комнат, доступных из данной можно просто вычислять: в каждом ряду на две комнаты больше предыдущего (разность квадратов!), если комната чётная в ряду, из неё доступна комната предыдущего ряда (с разностью в ширину ряда минус один), а если нечётная — следующего ряда (с разностью в ширину ряда плюс один). Кроме того, если комната в ряду не первая, из неё доступна предыдущая в этом ряду, а если не последняя, то следующая. Наконец, комнаты последнего ряда не имеют соседей с юга.

Построим таким образом карту, в которой каждой комнате будет соответствовать список из не более чем трёх доступных из неё комнат. Для всех опасных комнат сделаем эти списки пустыми. Проходимость полученного лабиринта легко проверить методом «заливки».

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

```
lab={}
```

```

# Сформируем лабиринт по рядам
for r in xrange(1,11):
    w, n = r*2-1, (r-1)**2
    for i in xrange(1,w+1):
        lab[n+i]=[]
        if i>1: lab[n+i].append(n+i-1)
        if i<w: lab[n+i].append(n+i+1)
        if r>1 and i%2 == 0: lab[n+i].append(n+i-w+1)
        if r<10 and i%2 == 1: lab[n+i].append(n+i+w+1)
# Множество безопасных комнат
Safe=set(map(int,raw_input().split()))
# Множество опасных комнат
Unsafe=set(xrange(1,101))-Safe
# Из опасных комнат нет выхода
for i in Unsafe: lab[i]=[]
# Можно проверять доступность
Plan=[1] # План разведки
while Plan: # Пока есть что разведывать
    Next=Plan.pop(0) # Очередная комната для разведки
    if Next == 100: # Это выход!
        print "Да"
        break
    Plan.extend(lab[Next]) # Добавим безопасные комнаты в план
    lab[Next]=[] # В текущую комнату заходить больше нет смысла
else: # Так и не нашли
    print "Нет"

```

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

Очный тур. Решения задач

26 апреля 2014 года

9–10 классы

Задачи по математике

Задача по математике считалась решенной, если правильный ответ был получен в результате строгих и обоснованных рассуждений.

1. Далёкое и близкое

Найти минимальную и максимальную разность между двумя натуральными числами, суммы цифр которых делятся на 8, а между ними таких чисел нет.

Решение: см. задачу 2 для 8-го класса.

2. Неизвестный квадрат

Дано число, состоящее из 2014 единиц (других цифр нет). Доказать, что к этому числу можно приписать справа 2015 цифр так, что получится точный квадрат.

Решение: см. задачу 3 для 8-го класса.

3. Сдвиг?

На доске выписаны все натуральные числа от 1 до 50. За одну операцию разрешается выбрать два числа x и y и заменить их на числа $(3x - 7y)$ и $(5x - 6y)$. Можно ли в результате конечного числа операций получить на доске все целые числа от 0 до 49?

Ответ. Нельзя.

Решение: см. задачу 4 для 8-го класса.

4. Пять раз отмерь...

В треугольнике ABC точка D — середина стороны AB , точка E — середина стороны BC , точка F — середина стороны AC , точка Q лежит внутри треугольника ABC . По изображениям пяти отрезков, равных отрезкам QA, QB, QD, QE, QF , с помощью циркуля и линейки постройте отрезок, равный QC .

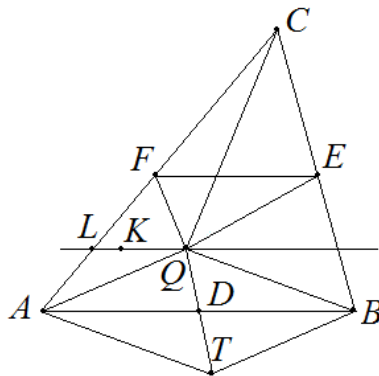


Рис. 3.

Решение. Предположим, что построение осуществлено (см. рис. 3). Пусть точка T лежит на луче QD , причем $QD = DT$, точка L лежит на стороне AC треугольника ABC , причем прямая QL параллельна прямой AB . Заметим: четырёхугольник $AQBT$ — параллелограмм, и в треугольнике QBT длины всех сторон известны или легко находятся (длина QB указана в условии, $|BT| = |QA|$, $|QT| = 2|QD|$).

Приступим к построению. По третьему признаку равенства треугольников можно единственным (с точностью до движений) образом построить треугольник QBT . В этом треугольнике следует найти точку D как середину стороны QT , на луче BD найти такую точку A , что $|BD| = |DA|$. Далее построим проходящую через точку Q прямую, параллельную прямой AB , и отметим на этой прямой точку K , лежащую в разных полуплоскостях с точкой B относительно прямой AQ . FE — средняя линия треугольника ABC , так что отрезок длины, равной $|FE|$, может быть построен как равный отрезку BD . Построим треугольник $Q'E'F'$, равный треугольнику QEF (в нём известны все стороны).

Отложим от луча QK в полуплоскость, не содержащую точки D , угол, равный углу $Q'F'E'$ (и равный углу QFE). На стороне этого угла, отличной от QK , на расстоянии $|Q'F'|$ отметим точку F . Отметим на луче AF точку C так, что $|AF| = |FC|$. Отрезок QC — искомый. Единственность длины этого отрезка обеспечивается единственностью (с точностью до движений) треугольников QBT и $Q'E'F'$.

5. Такие разные, а всё-таки вместе!

Разрезать шахматную доску 8×8 по границам клеток на попарно неравные части (т. е. не совпадающие при наложении, — переворачивать части разрешается, цвет клеток не учитывается) так, чтобы получилось наибольшее число частей. Максимальность числа частей обосновать.

Ответ. Максимум числа частей равен 16.

Решение: см. задачу 5 для 8-го класса.

Задачи по информатике

Во всех задачах по информатике хорошим решением считался комментированный алгоритм, позволяющий на обычном современном компьютере получить ответ при любых допустимых входных данных за относительно небольшое время (порядка минуты).

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

6. Сомножители

Вывести первые 1000 составных натуральных чисел, которые раскладываются ровно на N ($1 < N < 100$) простых сомножителей.

Входные данные: N .

Выходные данные: последовательность чисел по одному в строке.

Решение: см. задачу 6 для 8-го класса.

7. Задача о шкатулке

Ослик Иа зашел в магазин подарков и обнаружил, что в нём нет ничего, кроме ста китайских шкатулок, имеющих вид параллелепипедов. В магазине имелся список всех шкатулок с указанием точных их размеров. Все шкатулки одинаково красивы, но ручек они не имеют и унести можно только одну. Зато внутрь можно положить ещё одну (но не две сразу — поцарапаются), внутрь второй — третью и так далее. Помогите Иа унести как можно больше шкатулок. Толщиной стенок пренебречь.

Входные данные: размеры шкатулок — 100 строк по три числа через пробел в строке.

Выходные данные: номера шкатулок, которые можно вложить одна в другую.

Решение: см. задачу 7 для 8-го класса.

8. Индейский узник

Все постройки индейцев племени польбаи имеют треугольное основание. Казематы во дворце Великого Вождя Куа-Тотля — тоже треугольник, разделённый на 100 одинаковых треугольных комнат стенами, параллельными стенам дворца. Узник находится в самой

северной комнате казематов, выход на поверхность — в наиболее юго-восточной комнате. Между всеми соседними комнатами имеются проходы, и в каждой комнате подготовлена смертельная ловушка. К счастью, узник умеет избегать некоторые из этих ловушек: у него имеется список относительно безопасных комнат. Комнаты пронумерованы от 1 до 100 по рядам с запада на восток: северная комната имеет № 1, следующий ряд — № 2, 3 (комната с вершиной на юг) и 4; третий ряд — № 5, 6, 7, 8 и 9; и т. д. (пример плана подобных, но меньших, казематов приведен на рисунке 2). Может ли узник добраться до выхода, заходя только в относительно безопасные комнаты?

Входные данные: номера безопасных комнат через пробел.

Выходные данные: «да» или «нет».

Решение: см. задачу 8 для 8-го класса.