Всероссийская олимпиада школьников по информатике, 2019–2020 уч. г.
Первый (школьный) этап, г. Москва
Разбор задания для 7–8 классов

Задача 1. Палиндромы

Вам даны пять чисел:

54321
48987
112233
299995
999999

Для каждого из этих чисел найдите **минимальное** целое число, которое было бы **больше** данного, и запись этого числа была бы палиндромом, то есть читалась бы одинаково как слева направо, так и справа налево. Например, палиндромами являются такие числа, как 121, 9009, 734437.

В ответе нужно записать пять целых чисел, записанных в отдельных строках. Порядок записи чисел в ответе менять нельзя. Если Вы не можете найти ответ для какого-то из данных чисел, вместо этого ответа запишите любое целое число.

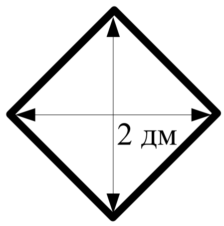
Решение

Ответом на это задание будут следующие числа.

54345
49094
113311
300003
1000001

Задача 2. Плитки

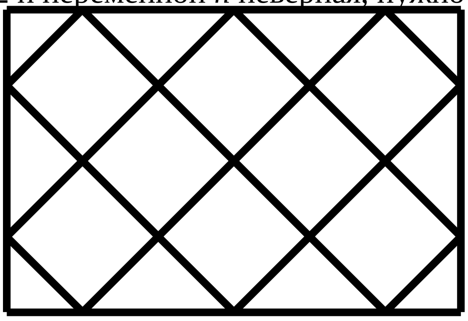
При проведении ремонта в квартире дизайнер предложил выложить стену узором из квадратных плиток, повернув их на 45°. Длина диагонали квадрата равна 2 дм.



Стена, которую необходимо покрыть плиткой, имеет размеры $n \times m$ дм, при этом числа n и m – целые чётные. Для покрытия такой стены необходимо какое-то количество целых квадратных плиток, а также несколько треугольных обрезков плитки. На рисунке на следующей странице приведён пример покрытия плиткой стены размером 4×6 дм, при этом было использовано 8 целых квадратных плиток.

По данным размерам стены n и m дм определите, какое число целых квадратных плиток будет содержать такой узор.

Ответом на эту задачу является некоторое выражение, которое может содержать целые числа, переменные n и m (записываемые английскими буквами), операции сложения (обозначаются «+»), вычитания (обозначаются «−»), умножения (обозначаются «*»), деления (обозначаются «/») и круглые скобки для изменения порядка действий. Запись вида «2n» для обозначения произведения числа 2 и переменной *n* неверная, нужно писать «2 * *n*».



Ваше выражение должно давать правильный ответ для любых чётных значений *n* и *m*, например, для *n* = 4 и *m* = 6 значение выражения должно быть равно 8.

Пример правильной формы записи ответа.

$$m / 2 + (m * n - m) * 2$$

Решение

В верхнем ряду будет *m* / 2 плиток, и таких рядов будет *n* / 2. А между ними находятся ещё *n* / 2 − 1 ряд по *m* / 2 − 1 плитке в каждом. Ответ.

$$(n / 2) * (m / 2) + (n / 2 - 1) * (m / 2 - 1)$$

Задача 3. Переливания

Есть три сосуда объёмами 6 л (обозначим буквой A), 10 л (B) и 15 л (C). С ними возможно выполнять следующие операции:

1. Наполнить какой-то сосуд водой из крана, пока он не заполнится целиком.
2. Вылить всю воду из какого-то сосуда.
3. Перелить воду из одного сосуда в другой, пока в первом сосуде не кончится вода или второй сосуд не заполнится целиком.

При помощи этих операций Вам необходимо отмерить 1 л воды, при этом нужно использовать как можно меньше воды (учитывается вся вода, которая была суммарно налита из крана). Составьте алгоритм переливаний, в результате исполнения которого в каком-то из сосудов окажется 1 л воды, а объём использованной воды будет как можно меньше.

Для записи алгоритма используются следующие команды.

>X	Наполнить сосуд X (вместо X должен быть один из символов A, B, C).
X>	Вылить воду из сосуда X (вместо X должен быть один из символов A, B, C).
X>Y	Перелить воду из X в Y (вместо X и Y должны быть два различных символа из A, B, C). Нельзя переливать воду из одного сосуда в тот же самый сосуд.

Команды записываются по одной в строке. Например, следующая последовательность команд

```
>B
B>C
C>
```

обозначает, что сначала наполняется сосуд B, потом вода из сосуда B переливается в сосуд C, потом из сосуда C выливается вся вода.

Чем меньше воды будет использовано для реализации Вашего алгоритма, тем больше баллов Вы получите.

Решение

Несложно придумать решение, использующее 16 л воды, если заметить, что 10 + 6 − 15 = 1, то есть нужно будет налить воду в A и B, перелить всё в C, тогда останется

1 л. Но можно придумать решение, использующее 15 литров. Для этого нужно налить воду в сосуд C.

```
 >C
C>B
B>A
 B>
A>C
C>B
```

Решения, использующих меньшее число литров, не существует. Для этого достаточно заметить, что т. к. НОД(A, B) = 2, НОД(A, C) = 3, НОД(B, C) = 5, то отмерить один л, используя только два каких-то сосуда, нельзя. Если же будет суммарно налито меньше 15 л, то третий сосуд фактически не используется (наливание воды в этот сосуд равноценно её выливанию, или наливанию в сосуд бесконечного объёма), поэтому 15 л — минимальный объём воды, необходимый для решения задачи.

Задача 4. Квест

Новый квест, в котором участники должны выбраться с территории проведения, представляет собой прямоугольник из 16 комнат в виде квадрата 4 × 4. Каждая комната имеет четыре двери, ведущие в соседние комнаты, из комнат на краю прямоугольника двери ведут наружу, через эти двери можно покинуть территорию проведения квеста.

В начале квеста в каждой комнате находится по человеку, а все двери заперты. После начала квеста организаторы дистанционно открывают в каждой комнате запирающий механизм одной из четырёх дверей. Теперь человек, находящийся в этой комнате, может открыть эту дверь и перейти в соседнюю комнату, через другие три двери выйти из этой комнаты нельзя. При этом может оказаться так, что дверь, соединяющая две комнаты, будет отпираться с одной стороны, тогда пройти через эту дверь можно только с той стороны, с которой она будет открываться, проходить через дверь в обратном направлении нельзя, если в соседней комнате будет отперта не эта дверь, а какая-то другая. Если комната находится на краю территории и из этой комнаты открыта дверь наружу, то, пройдя через эту дверь, участник навсегда покидает территорию квеста.

После начала квеста и отпирания дверей участники начинают перемещаться между комнатами. Каждый участник перемещается в соседнюю открытую комнату и продолжает перемещаться до тех пор, пока не покинет территорию квеста. Однако возможна ситуация, когда некоторые участники будут бесконечно перемещаться между комнатами и никогда не выйдут наружу.

Разработчки квеста попросили Вас составить такой план отпирания дверей, при котором ровно 7 человек из 16 смогут выбраться наружу с территории квеста. **При этом Вам необходимо минимизировать количество дверей, которые будут открыты из крайних комнат наружу (тех дверей, через которые участники будут покидать территорию квеста).**

Ответ на эту задачу нужно записать в виде плана территории квеста, состоящего из 4 строк, в каждой строке должно быть ровно 4 символа из следующего числа возможных.

U: дверь в верхнюю по данному плану комнату.

D: дверь в нижнюю комнату.

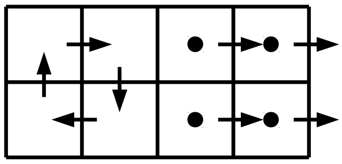
L: дверь в левую комнату.

R: дверь в правую комнату.

Например, рассмотрим следующий план квеста, в котором 8 комнат:

```
RDRR
ULRR
```

Этот план соответствует следующему рисунку. Стрелками обозначены открытые двери, точками помечены комнаты, обитатели которых выйдут из квеста.



В этом примере наружу выходят 4 человека из 8.

Вам необходимо составить план квеста из 4 рядов по 4 комнаты, в котором наружу выходят 7 человек из 16. При этом чем меньше будет число дверей, открытых из крайних комнат наружу, тем больше баллов Вы получите.

Решение

Несложно составить схему, в которой все 7 человек выходят наружу через одну дверь, а остальные 9 ходят по кругу. Например, так.

```
RRRR
DDDU
DDDU
RLLU
```

В этом примере обитатели верхней строки и правого столбца перемещаются в верхний правый угол и выходят из него. Остальные обитатели спускаются в нижний ряд и зацикливаются в двух левых клетках нижнего ряда.

Задания по программированию

Решением задач 5–7 является программа, написанная на одном из языков программирования. Ограничение по времени работы программы в задачах 5–7 – 1 с. Ограничение по памяти – 512 MB.

Решения оцениваются, только если они выдают правильный ответ на первом примере входных и выходных данных, приведёном в условии задачи. Проверка решений производится сразу же после отправки, по каждой задаче оценивается решение, набравшее наибольшее число баллов. На странице «Итог» Вы можете видеть окончательный балл по задачам 5–7.

Программа не должна выводить никаких иных сообщений, кроме того, что требуется найти в задаче. Во всех задачах целые числа во входных и выходных данных записываются только цифрами (т.е. недопустимо использование записи 1000000.0 или 1e6 вместо числа 1000000). Каждое число во входных данных записано в отдельной строке.

Задача 5. Часовые пояса

Таня решила позвонить своей подруге, но вспомнила, что та живёт очень далеко, поэтому в часовом поясе подруги может быть слишком поздно или рано. Часы у Тани показывают ровно H часов, Таня живёт в часовом поясе UTC+A, а её подруга – в часовом поясе UTC+B. Помогите Тане определить время в часовом поясе подруги в этот момент.

Программа получает на вход три целых числа H, A и B, $0 \le H \le 23$, $-11 \le A \le 12$, $-11 \le B \le 12$.

В часовом поясе UTC+A местное время больше, чем время в часовом поясе UTC+0 на A часов (если же $A < 0$, то меньше на $|A|$ часов). Например, если в часовом поясе UTC+0 сейчас 12 часов, то в часовом поясе UTC+1 – 13 часов, а в часовом поясе UTC−1 – 11 часов.

Программа должна вывести одно число – время (количество часов) в часовом поясе подруги.

Под временем в этой задаче подразумевается количество часов, которое может принимать значения от 0 до 23. При решении задачи обратите внимание, что в часовом поясе подруги может быть уже следующая дата или предыдущая дата, программа должна вывести количество часов на часах подруги в этот момент, то есть число от 0 до 23.

Пример входных и выходных данных

Ввод	**Вывод**	**Примечание**

`15`
`3`
`-5`
	`7`	У Тани – 15 часов, она живёт в часовом поясе UTC+3. В часовом поясе UTC+0 сейчас 12 часов. Подруга живёт в часовом поясе UTC−5, и у неё сейчас 7 часов.

Решение

Если в часовом поясе UTC+A время H часов, то в поясе UTC+0 время $H - A$, а в часовом поясе UTC+B время будет равно $H - A + B$. Но при этом не учитываются возможные переходы между сутками — время может оказаться в следующих сутках или в предыдущих сутках. Это можно учесть при помощи двух условий: если получили время, больше или равное 24 часов, то нужно вычесть 24 часа, а если получили отрицательное время, то нужно прибавить 24 часа. Получается следующее решение:

```
H = int(input())
A = int(input())
B = int(input())
answer = H – A + B
if answer >= 24:
    answer -= 24
```

```
if answer < 0:
    answer += 24
print(answer)
```

Но вместо условий проще взять остаток от деления на 24:

```
H = int(input())
A = int(input())
B = int(input())
answer = (H – A + B) % 24
print(answer)
```

Заметим, что в большинстве языков программирования (Pascal, C++, Java, C#) при взятии остатка от деления отрицательного числа на положительное получится отрицательный результат, то есть решение задачи будет неверным. В этом случае нужно действовать так (пример для Pascal).

```
answer := (H – A + B + 24) mod 24
```

Задача 6. Чётные – нечётные

Маша любит чётные числа, а Миша – нечётные. Поэтому они всегда радуются, если встречают числа, которые им нравятся.

Сегодня им встретились все целые числа от *A* до *B* включительно. Маша решила посчитать сумму всех чётных чисел от *A* до *B*, а Миша – сумму всех нечётных, после чего они начали спорить, у кого получилась сумма больше. Помогите им – найдите разность между суммой Маши и суммой Миши.

Программа получает на вход два целых положительных числа *A* и *B*, не превосходящие 2×10^9. Программа должна вывести одно число – разность между суммой чётных чисел и суммой нечётных чисел от *A* до *B*.

Примеры входных и выходных данных

Ввод	**Вывод**	**Примечание**
3 6	2	Сумма чётных чисел равна 4 + 6 = 10, сумма нечётных чисел равна 3 + 5 = 8, разность равна 2.
3 7	-5	Сумма чётных чисел равна 4 + 6 = 10, сумма нечётных чисел равна 3 + 5 + 7 = 15, разность равна −5.

Система оценивания

Решение, правильно работающее только для случаев, когда числа *A* и *B* не превосходят 100, будет оцениваться в 6 баллов.

Решение

На 6 баллов можно перебрать все числа от A до B и сложить их со знаком «+» или «−» в зависимости от чётности. Пример решения.

```
a = int(input())
b = int(input())
s = 0
for i in range(a, b + 1):
    if i % 2 == 0:
        s += i
    else:
        s -= i
print(s)
```

На полный балл можно посчитать сумму всех чётных и нечётных чисел на отрезке, используя сумму арифметической прогрессии. А именно, если числа A и B оба чётные (или, наоборот, оба нечётные), то сумма всех чётных (или нечётных) чисел на отрезке от A до B равна (A + B) / 2 * ((B − A) / 2 + 1). Но для того, чтобы использовать эту формулу, необходимо сделать границы отрезка одной чётности. Исправим границы отрезка – прибавим 1 к числу A, если A – нечётное, а также вычтем из B число 1, если B – нечётное. Получим два чётных значения a2 и b2, соответствующих границам чётного отрезка, содержащего чётные числа от A до B. Аналогично, получим два нечётных значения a1 и b1, соответсвующих границам нечётного отрезка. Затем посчитаем сумму чётных чисел на отрезке от a2 до b2 и сумму нечётных чисел на отрезке от a1 до b1.

```
a = int(input())
b = int(input())
a2 = a + a % 2
b2 = b - b % 2
even = (b2 + a2) // 2 * ((b2 - a2) // 2 + 1)
a1 = a + (1 - a % 2)
b1 = b - (1 - b % 2)
odd = (b1 + a1) // 2 * ((b1 - a1) // 2 + 1)
print(even – odd)
```

Заметим, что при реализации такого решения на языках Pascal, C++, Java, C# необходимо использовать 64-битные целочисленные переменные для хранения переменных even и odd во избежани переполнения. Можно реализовать решение и без переполнения 32-битных чисел если заметить, что разница между двумя соседними числами равна 1, далее подсчитать количество пар чисел. Но в таком решении придётся разбирать несколько случаев (4 возможных случая четности каждой из двух границ), поэтому такое решение более сложно в реализации.

Задача 7. Уточки

Как известно, при разработке и отладке программ большую помощь могут оказать игрушечные жёлтые уточки (см. статью «Метод утёнка» в википедии), поэтому Денис собрал большую коллекцию жёлтых уточек. Коллекция уже настолько большая, что Денис решил расставить уточек на полки шкафа. Сначала он начал ставить на каждую полку по A уточек, но одна уточка оказалась лишней. Тогда он заново начал расставлять уточек на полки, ставя на каждую полку по B уточек, но в этом случае ему не хватило одной уточки, чтобы на каждой полке оказалось ровно B уточек. Определите минимальное число уточек, которое могло быть в коллекции Дениса.

Программа получает на вход два целых положительных числа A и B, $2 \le A \le 2\times10^9$, $2 \le B \le 2\times10^9$ – количество уточек при расстановке на полке в первом и во втором случаях. Программа должна вывести одно число – минимально возможное количество уточек в коллекции Дениса. Гарантируется, что ответ существует и не превосходит 2×10^9.

Пример входных и выходных данных

Ввод	**Вывод**	**Примечание**
5 3	11	11 = 5 × 2 + 1 11 = 3 × 4 − 1

Система оценивания

Решение, правильно работающее только для случаев, когда числа A и B не превосходят 100, будет оцениваться в 4 балла.

Решение

На 4 балла можно перебрать все числа, пока не будет найдено число, дающее остаток 1 при делении на A и дающее остаток B − 1 при делении на B. Пример решения.

```
a = int(input())
b = int(input())
ans = 1
while ans % a != 1 or ans % b != (b – 1):
    ans += 1
print(ans)
```

Для ускорения этого решения можно перебирать не все числа, а только числа, дающие остаток 1 при делении на A. Поскольку мы начали с числа 1, которое даёт остаток 1 при делении на A, то дальше нужно прибавлять к числу значение A вместо 1. Достаточно в этом примере заменить ans += 1 на ans += a, и решение будет набирать 7 баллов. Его можно ещё упростить, если убрать из цикла проверку того, что ans даёт остаток 1 при делении на a. Пример решения на 7 баллов.

```
a = int(input())
b = int(input())
ans = 1
while ans % b != (b – 1):
    ans += a
print(ans)
```

Аналогично перебирать числа, которы дают остаток b – 1 при делении на b, если начать с числа b − 1 и дальше идти с шагом b. Такое решение будет также набирать 7 баллов.

Для того, чтобы набрать 10 баллов, необходимо выбрать из чисел a и b наибольшее и выбрать шаг, равный большему из двух чисел.

```
a = int(input())
b = int(input())
if a >= b:
    ans = 1
    while ans % b != b - 1:
        ans += a
else:
    ans = b - 1
    while ans % a != 1:
        ans += b
print(ans)
```

Это решение корректно, поскольку в условии задачи говорится, что ответ существует и не превосходит 2×10^9, то есть при больших числах A и B количество шагов цикла не превосходит 2×10^9 / max(A, B).