



(от А до Н), которые являются обозначениями крайних томов в переворачиваемом фрагменте, например, выше был рассмотрен пример для операции «Н F».

Чем меньше операций будет в вашем алгоритме, тем больше баллов вы получите, при условии, что в результате применения вашего алгоритма тома будут расставлены по порядку от А до Н.

### Задача 3. Гирьки

Есть чашечные весы без делений. Для взвешивания груза также можно использовать гирьки, массы которых – целое число граммов. Вам необходимо предложить набор гирек, при помощи которого можно отмерить на весах любую массу, равную целому числу граммов от 1 до 40. Гирьки можно класть на каждую чашку весов, чашки весов должны находиться в равновесии, при этом на одной из чашек весов должен находиться взвешиваемый груз. Массы гирек в наборе могут повторяться.

Например, при помощи трёх гирек массами 1, 1 и 5 граммов можно отмерить любую целочисленную массу от 1 до 7 граммов по следующей схеме ( $x$  – взвешиваемая масса):

$$1 \text{ грамм: } x = 1,$$

$$2 \text{ грамма: } x = 1 + 1,$$

$$3 \text{ грамма: } x + 1 + 1 = 5,$$

$$4 \text{ грамма: } x + 1 = 5,$$

$$5 \text{ граммов: } x = 5,$$

$$6 \text{ граммов: } x = 5 + 1,$$

$$7 \text{ граммов: } x = 5 + 1 + 1.$$

Ответом на эту задачу являются несколько целых чисел, записанных через пробел, – массы гирек, при помощи которых можно отмерить любую целочисленную массу от 1 до 40. В наборе должно быть не более 8 чисел. Числа в наборе могут повторяться. Чем меньше гирек будет в предложенном наборе, тем больше баллов вы получите, при условии, что, используя гирьки из данного набора, можно отмерить каждую целочисленную массу от 1 до 40.

### Задача 4. Возведение в степень

Вы хотите возвести данное число  $a$  в некоторую целочисленную степень  $n$ , но ваш калькулятор умеет только перемножать числа. Например, вы можете вычислить  $a^2 = a \times a$ , затем вы можете вычислить  $a^3 = a^2 \times a$  или  $a^4 = a^2 \times a^2$ .

Вы можете по-разному организовать вычисление значения  $a^n$ . Например, вычислить  $a^5$  можно за 4 умножения:

$$1) a^2 = a \times a,$$

$$2) a^3 = a^2 \times a,$$

$$3) a^4 = a^3 \times a,$$

$$4) a^5 = a^4 \times a.$$

Но можно вычислить  $a^5$  всего лишь за 3 умножения:

$$1) a^2 = a \times a,$$

$$2) a^3 = a^2 \times a,$$

$$3) a^5 = a^3 \times a^2.$$

Вам необходимо определить, за какое минимальное число умножений можно вычислить следующие степени: 7, 15, 23, 63. Вычисление каждой из этих степеней должно быть независимо от остальных, то есть при вычислении 15-й степени нельзя использовать вычисления, сделанные ранее для вычисления 7-й степени. Вы решаете четыре независимые задачи – за какое минимальное число умножений можно вычислить 7-ю степень, 15-ю степень, 23-ю степень и 63-ю степень.

Ответ на это задание записывается в четырёх строках. Каждая строка должна содержать последовательность вычисления каждой из указанных степеней. Первая строка

должна содержать последовательность вычисления 7-й степени, вторая строка – 15-й степени, третья строка – 23-й степени, четвертая строка – 63-й степени.

Каждая строка содержит через пробел несколько целых чисел – значения степеней в том порядке, в котором они вычисляются. Например, для вычисления 5-й степени решение можно записать в виде строки

2 3 5

или

2 4 5,

что означает, что последовательно вычисляются степени  $a^2$ ,  $a^3$ ,  $a^5$  (одно возможное решение) или  $a^2$ ,  $a^4$ ,  $a^5$  (другое возможное решение). Таким образом, каждая строка должна начинаться числом 2, а заканчиваться тем значением степени, которое нужно вычислить (7, 15, 23, 63).

Чем меньше операций умножения вы будете использовать, тем больше баллов вы получите, при условии, что предложенные последовательности вычисления степеней являются корректными.

## 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 секунда. Ограничение по памяти – 512 МВ.

Решения оцениваются, только если они выдают правильный ответ на первом примере входных и выходных данных, приведённом 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.

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. Спиннеры

Спиннер – модная игрушка с подшипником в основании, к которому прикреплены лопасти. Афанасий открыл бизнес по производству спиннеров. Он выяснил, что за спиннер, у которого  $N$  лопастей, покупатели готовы платить  $A + B \times N$  рублей, но при этом покупатель не станет покупать спиннер, если его цена будет выше  $C$  рублей. Определите максимальное число лопастей спиннера, который согласится приобрести покупатель.

Программа получает на вход три числа  $A$ ,  $B$ ,  $C$  (стоимость основания спиннера, стоимость одной лопасти и максимальная стоимость всего спиннера). Все числа – целые положительные, не превосходящие  $2 \times 10^9$ , при этом  $A \leq C$ .

Программа должна вывести одно число – максимальное число лопастей спиннера.

### Пример входных и выходных данных

Ввод	Вывод	Примечание
20 10 55	3	Спиннер с 3 лопастями будет стоить 50 рублей, а с 4 лопастями – 60 рублей. Максимальная возможная стоимость спиннера – 55 рублей, поэтому максимальное число лопастей равно 3.

### Система оценивания

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 100, будет оцениваться в 6 баллов.

## Примеры оформления решения задачи

Ниже даны примеры ввода и вывода данных к этой задаче на нескольких языках программирования. Выберите один из языков программирования, допишите соответствующую программу и отправьте её на проверку с использованием одного из допустимых компиляторов.

<b>Python</b> A = int(input()) B = int(input()) C = int(input()) ... N = ... print(N)	<b>Pascal</b> var A, B, C, N: longint; begin readln(A); readln(B); readln(C); ... N := ... writeln(N); end.
<b>C++</b> #include<iostream> using namespace std; int main() { int A, B, C, N; cin >> A >> B >> C; ... N = ... cout << N << endl; return 0; }	<b>Java</b> import java.io.*; public class Main { public static void main(String[] args) throws Exception { DataInputStream in = new DataInputStream(System.in); int A, B, C, N; A = Integer.parseInt(in.readLine()); B = Integer.parseInt(in.readLine()); C = Integer.parseInt(in.readLine()); ... N = ... System.out.println(N); } }
<b>Алгоритмический язык (Кумир)</b> алг Задача1 нач цел A, B, C, N ввод A ввод B ввод C ... N := ... вывод N кон	<b>C#</b> using System; using System.IO; class Program { static void Main() { int A, B, C, N; A = int.Parse(Console.ReadLine()); B = int.Parse(Console.ReadLine()); C = int.Parse(Console.ReadLine()); ... N = ... Console.WriteLine("{0}", N); } }

## Задача 6. Снова спиннеры

Денис тоже решил заняться производством и продажей спиннеров, но он считает, что у спиннера может быть только три или четыре лопасти. У него есть ровно  $M$  лопастей, которые он может прикреплять к основаниям, и неограниченный запас оснований. Он хочет изготовить несколько трёхлопастных и несколько четырёхлопастных спиннеров так, чтобы использовать все  $M$  лопастей. Определите, сколько спиннеров каждого вида он должен произвести.

Программа получает на вход одно целое положительное число  $M$ , не превосходящее  $2 \times 10^9$ , – количество лопастей, которое есть у Дениса.

Программа должна вывести два целых числа – количество спиннеров с 3 лопастями и количество спиннеров с 4 лопастями, которые должен произвести Денис. Если у задачи есть

несколько решений, нужно вывести любое из них. Если Денис не может использовать ровно  $M$  лопастей для производства спиннеров, программа должна вывести два числа 0.

#### Примеры входных и выходных данных

Ввод	Вывод	Примечание
10	2 1	$10 = 3 \times 2 + 4 \times 1$
1	0 0	Невозможно произвести спиннеры так, чтобы суммарное число лопастей было равно 1.

#### Система оценивания

Решение, правильно работающее только для случаев, когда  $M$  не превосходит 100, будет оцениваться в 6 баллов.

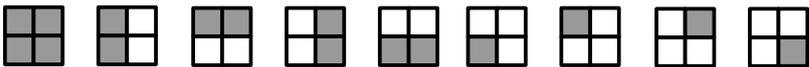
### Задача 7. Не про спиннеры

Саша совсем не любит спиннеры, поэтому он рисует в тетрадке. Он взял тетрадный лист из  $N \times M$  клеточек и пронумеровал все клетки различными числами. Теперь ему стало интересно, сколько различных прямоугольников он может вырезать из этого листа бумаги по границам клеточек.

Программа получает на вход два числа  $N$  и  $M$  – размеры исходного листа. Все числа – целые положительные, не превосходящие 75000.

Программа должна вывести одно число – количество прямоугольников, которые можно вырезать из данного листа бумаги (весь лист целиком также считается одним из возможных прямоугольников).

#### Примеры входных и выходных данных

Ввод	Вывод	Примеры вырезанных прямоугольников
2 2	9	
3 1	6	

#### Система оценивания

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 10, будет оцениваться в 4 балла.

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 200, будет оцениваться в 7 баллов.