

Всероссийская олимпиада школьников по информатике, 2017/18 уч. год
Первый (школьный) этап, г. Москва
Задания для 9–11 классов

Ограничение по времени работы программы в каждой задаче – 1 секунда.

Каждая задача оценивается в 100 баллов. Вы можете отправить на проверку не более 50 решений суммарно по всем задачам. Решения оцениваются, только если они выдают правильный ответ на всех примерах входных и выходных данных, приведённых 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 всем задачам.

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 (то есть недопустимо использование записи 1000000.0 или 1e6 вместо числа 1000000). Каждое число во входных данных записано в отдельной строке.

Задача 1. 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 2×10^9 , 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 60 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.

Язык программирования Python	Язык программирования Pascal
Пример для версий 2 и 3 языка Python	Компиляторы Free Pascal, Borland Delphi, Pascal ABC.NET
<pre>A = int(input()) B = int(input()) C = int(input()) ... N = ... print(N)</pre>	<pre>var A, B, C, N: longint; begin readln(A); readln(B); readln(C); ... N := ... writeln(N); end.</pre>

Язык программирования Basic Компилятор Free Basic (аналог qbasic)	Язык программирования Basic Компилятор Mono Visual Basic
<pre> DIM A AS LONG DIM B AS LONG DIM C AS LONG DIM N AS LONG INPUT A INPUT B INPUT C ... N = ... print N </pre>	<pre> Module ProgramA Sub Main() DIM A, B, C, N AS INTEGER A = CInt(Console.ReadLine()) B = CInt(Console.ReadLine()) C = CInt(Console.ReadLine()) ... N = ... Console.WriteLine(CStr(N)) End Sub End Module </pre>
Язык программирования C Компилятор GNU C	Язык программирования C++ Компилятор GNU C++
<pre> #include<stdio.h> int main() { int A, B, C, N; scanf("%d%d%d", &A, &B, &C); ... N = ... printf("%d\n", N); return 0; } </pre>	<pre> #include<iostream> using namespace std; int main() { int A, B, C, N; cin >> A >> B >> C; ... N = ... cout << N << endl; return 0; } </pre>
Алгоритмический язык Интерпретатор Кумир	Язык программирования C# Компилятор Mono C#
<pre> алг Задача1 нач цел A, B, C, N ввод A ввод B ввод C ... N := ... вывод N кон </pre>	<pre> 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); } } </pre>
Язык программирования PHP Работает в режиме CLI (без web-сервера)	Язык программирования Java
<pre> <?php \$A = fgets(STDIN); \$B = fgets(STDIN); \$C = fgets(STDIN); ... \$N = ... print \$N; ?> </pre>	<pre> 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); } } </pre>

Задача 2. 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 2×10^9 , – 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 60 баллов.

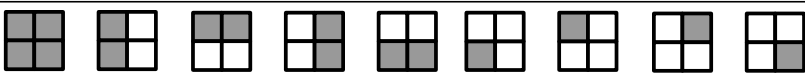

Задача 3. Не про спиннеры

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 40 баллов.

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 200, будет оцениваться в 70 баллов.

Задача 4. Плацкартный вагон

В плацкартном вагоне 54 места, пронумерованных числами от 1 до 54. Вагон разбит на 9 купе. Первые 36 мест расположены по левую сторону от прохода, места 1–4 находятся в первом купе, места 5–8 – во втором и т. д. В девятом купе находятся места с номерами 33–36. По правую сторону от прохода находятся боковые места, их номера от 37 до 54, причём они нумеруются в противоположном направлении: места 37 и 38 находятся напротив девятого купе, а места 53 и 54 – напротив первого. Ниже приведена схема всех мест в вагоне.

1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36
53	54	51	52	49	50	47	48	45	46	43	44	41	42	39	40	37	38

Группа школьников едет на олимпиаду и будет всю дорогу крутить спиннеры. Поэтому им нужно купить места в нескольких подряд идущих купе вместе с прилегающими боковыми местами. Даны номера свободных мест в поезде. Определите, какое наибольшее число подряд идущих купе полностью свободны.

Программа получает на вход число N – количество свободных мест в вагоне ($0 \leq N \leq 54$). Следующие N строк содержат номера свободных мест – различные числа от 1 до 54 в произвольном порядке, по одному числу в строке.

Программа должна вывести одно целое число – максимальное число подряд идущих свободных купе (купе – 4 места слева от прохода и 2 боковых места) в этом вагоне.

Примеры входных и выходных данных

Ввод	Вывод	Примечание
12 5 6 3 4 8 7 51 9 10 54 49 52	1	Свободно одно купе с местами 5, 6, 7, 8, 51, 52.
1 1	0	В вагоне только одно свободное место, поэтому свободных купе нет совсем.

Система оценивания

Решение, правильно работающее только для случаев, когда ответом является число 0 или 1, будет оцениваться в 40 баллов.

Задача 5. Кинотеатр

В первом ряду кинотеатра $N + 2$ мест, крайние места заняты персоналом кинотеатра, но N мест посередине свободно. K школьников входят в зрительный зал по очереди, и, конечно же, каждый школьник достаёт спиннер и начинает его крутить до начала сеанса. Поэтому каждый школьник выбирает себе место как можно дальше от уже занятых мест. А именно, школьник находит самый большой свободный участок в ряду (любой, если таких несколько) и садится посередине него. Если число свободных мест на этом участке было нечётно, то школьник садится точно посередине участка, тогда слева и справа от него остаётся поровну свободных мест. Если же это число чётно, то школьник выбирает одно из двух свободных мест посередине, тогда с одной стороны от школьника будет на одно свободное место больше, чем с другой стороны.

По данным числам N и K определите, сколько мест осталось свободными с двух сторон от школьника, который занял место последним (K -м по счёту).

Программа получает на вход два целых числа N и K , $1 \leq K \leq N \leq 10^{18}$, и должна вывести два целых числа **в порядке неубывания** – количество свободных мест с двух сторон от школьника, который последним занял место в ряду.

Примеры входных и выходных данных

Ввод	Вывод	Примечание
10 1	4 5	В зале 10 свободных мест, первый школьник сел посередине, с одной стороны от него 4 места, с другой стороны – 5.
10 2	2 2	Второй вошедший в зал школьник садится посередине группы из 5 свободных мест, с каждой стороны от него остаётся по 2 свободных места.
10 3	1 2	После того, как два школьника сели на места, в зале остались группы свободных мест из 4, 2, 2 мест. Третий школьник садится посередине группы из 4 мест, поэтому с одной стороны от него 1 место, с другой стороны – 2 места.

Система оценивания

Решение, правильно работающее только для случаев, когда N не превосходит 100, будет оцениваться в 30 баллов.

Решение, правильно работающее только для случаев, когда N не превосходит 10^5 , будет оцениваться в 60 баллов.