**Всероссийская олимпиада школьников 2019-2020**

**Школьный этап**

**Информатика**

**7–8 класс**

***Продолжительность – 45 минут***

***Максимальный балл – 40***

**Задание №1. Семизначное число (10 баллов)**

Найдите натуральное число, которое удовлетворяет следующим условиям:

1. запись числа состоит из семи цифр;
2. сумма всех цифр числа равна 39;
3. в записи числа есть хотя бы одна цифра 4;
4. в записи числа есть хотя бы одна цифра 7;
5. запись числа является палиндромом, то есть одинаково читается как слева направо, так и справа налево (например, такими числами-палиндромами являются числа 121 и 7007, но не является число 123);
6. число является максимальным из всех чисел, удовлетворяющих пунктам 1-5.

В ответе запишите найденное вами число.

***Система оценивания***

Баллы начисляются за правильное решение следующим образом:

**5 баллов** – записан только правильный ответ.

**7 баллов** – записан правильный ответ и есть частичное обоснование решения.

**10 баллов** – записан правильный ответ и есть полное обоснование решения.

**Задание №2. Спиральная дорожка (10 баллов)**

Лужайка в деревне Простоквашино имеет форму прямоугольника размером *a* × *b* метров, разбитого на квадраты со стороной 1 метр. Дядя Федор решил поставить внутри лужайки ограждения между некоторыми квадратами так, чтобы образовалась спиральная дорожка, закручивающаяся к центру лужайки. Определите длину такого ограждения.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

На рисунке изображена лужайка размером 5 × 4 и ограждение, которое необходимо поставить на ней. Длина ограждения для такой лужайки будет равна 12.

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

Пример правильного (по форме записи) выражения: 2\* (*b* −*a*).

***Система оценивания***

Баллы начисляются за правильное решение следующим образом:

**5 баллов** – записан только правильный ответ.

**8 баллов** – записан ответ с объяснением хода решения, но выражение не упрощено.

**10 баллов** – записан правильный ответ с объяснением хода решения.

**Задание №3. Возведение в степень (10 баллов)**

Дядя Федор решал математическую задачу, в которой необходимо было возвести данное число *a* в некоторую целочисленную степень *n*. Но его калькулятор умеет только перемножать числа. Например, можно вычислить *a*2= *a* × *a*, затем можно вычислить *a*3 = *a*2 × *a* или *a*4 = *a*2 × *a*2.

Можно по-разному организовать вычисление значения *an*. Например, вычислить *a*5 можно за 4 умножения:

1) *a*2 = *a* × *a*,

2) *a*3 = *a*2 × *a*,

3) *a*4 = *a*3 × *a*,

4) *a*5 = *a*4 × *a*.

Но можно вычислить *a*5 всего лишь за 3 умножения:

1) *a*2 = *a* × *a*,

2) *a*3 = *a*2 × *a*,

3) *a*5 = *a*3 × *a*2.

Вам необходимо определить, за какое минимальное число умножений Дядя Федор может 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, проделанные 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 четырех 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.

***Система оценивания***

Баллы начисляются за правильное решение следующим образом:

**3 балла** – записан ответ для вычисления *a*7

**5 баллов** – записан ответ для вычисления *a*7 *, a*15

**7 баллов** – записан ответ для вычисления *a*7 *, a*15, *a*23

**10 баллов** – записан ответ для вычисления *a*7 *, a*15, *a*23, *a*63

**Задание №4. Улица (10 баллов)**

По одну сторону улицы Почтовой в деревне Простоквашино находятся дома с нечетными номерами (1, 3, 5, …), по другую сторону – с четными (2, 4, 6, …). Дом № 1 находится напротив дома № 2, дом № 3 – напротив дома № 4 и т. д. До соседнего дома нужно идти вдоль по улице одну минуту, неважно, с какой стороны улицы он находится (то есть от дома № 1 нужно идти одну минуту как до дома № 3, так и до дома № 4). До дома, стоящего напротив, идти не нужно.



Почтальон Печкин вышел на улицу из дома номер *A* и должен доставить посылку в дом номер *B*.

Составьте программу, которая определит, сколько минут ему нужно идти вдоль по улице.

Программа получает на вход два различных целых положительных числа *A* и *B*, где A<B и не превосходящие 2×109, – номера домов. Программа должна вывести одно число – искомое количество минут.

***Пример входных и выходных данных***

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 18 | 3 |

***Система оценивания***

Решение, правильно работающее только для случаев, когда все входные числа не

превосходят 100, будет оцениваться в **6 баллов**.

Решение, правильно работающее только для случаев, когда все входные числа не

превосходят 1000, будет оцениваться в **8 баллов**.

Решение, правильно работающее для всех случаев, описанных в условии задачи,будет оцениваться в **10 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 Phyton** | **Язык программирования Pascal** |
| a=int(input())b=int(input())…n=…print(n) | var a, b, n:longint;beginreadln(a); readln(b);…n:=…writeln(n);end. |
| **Язык программирования C** | **Язык программирования C++** |
| #include<stdio.h>int main() {int a, b, n;scanf(“%d%d”, &a, &b);…n=…printf(“%d\n”, n);return 0;} | #include <iostream>using namespace std;int main() {int a, b, n;cin >> a >> b;…n=…cout << n << endl;return 0;} |
| **Язык программирования C#** | **Язык программирования Java** |
| using System;using System.IO;class Program {static void Main() {int a, b, n;a=int.Parse(Console.ReadLine());b=int.Parse(Console.ReadLine());…n=…Console.WriteLine("{0}", n);}} | import java.io.\*;import java.util.\*;class solver{  public static void main(String[] args) throws IOException  {DataInputStream in=new DataInputStream(System.in);int a, b, n;a=Integer.parseInt(in.readLine());b=Integer.parseInt(in.readLine());…n=…System.out.println(n);}} |

**Всероссийская олимпиада школьников 2019-2020**

**Школьный этап**

**Информатика**

**7–8 класс**

***Продолжительность – 45 минут***

***Максимальный балл – 40***

**Задание №1. Семизначное число (10 баллов)**

Найдите натуральное число, которое удовлетворяет следующим условиям:

1. запись числа состоит из семи цифр;
2. сумма всех цифр числа равна 39;
3. в записи числа есть хотя бы одна цифра 4;
4. в записи числа есть хотя бы одна цифра 7;
5. запись числа является палиндромом, то есть одинаково читается как слева направо, так и справа налево (например, такими числами-палиндромами являются числа 121 и 7007, но не является число 123);
6. число является максимальным из всех чисел, удовлетворяющих пунктам 1-5.

В ответе запишите найденное вами число.

***Система оценивания***

Баллы начисляются за правильное решение следующим образом:

**5 баллов** – записан только правильный ответ.

**7 баллов** – записан правильный ответ и есть частичное обоснование решения.

**10 баллов** – записан правильный ответ и есть полное обоснование решения.

**Решение**

Семизначное число-палиндром состоит из трех цифр в начале числа, одной цифры посередине числа, и трех цифр в конце числа, которые совпадают с первыми тремя цифрами.

Сумма всех цифр числа равна 39, и поскольку число является семизначным числом-палиндромом, то посередине десятичной записи числа стоит нечетная цифра. Значит, цифра 4 не стоит в числе посередине, и в числе содержится минимум две цифры 4 (по одной цифре среди первых трех и среди последних трех цифр числа).

На первое место в числе поставим цифру 9, как максимальную цифру. Поскольку 9 + 4 + 7 =20, то число 7 не может встречаться среди первых трех цифр числа (сумма первых трех цифр числа не может быть больше 19), значит, цифра 7 стоит посередине числа. Тогда сумма первых трех цифр числа равна (39 - 7) / 2 = 16, и поскольку две цифры из первых трех уже известны (это 9 и 4), то третья цифра из первых трех равна 3.

Поэтому максимальное число, удовлетворяющее условиям, равно 9437349.

**Задание №2. Спиральная дорожка (10 баллов)**

Лужайка в деревне Простоквашино имеет форму прямоугольника размером *a* × *b* метров, разбитого на квадраты со стороной 1 метр. Дядя Федор решил поставить внутри лужайки ограждения между некоторыми квадратами так, чтобы образовалась спиральная дорожка, закручивающаяся к центру лужайки. Определите длину такого ограждения.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

На рисунке изображена лужайка размером 5 × 4 и ограждение, которое необходимо поставить на ней. Длина ограждения для такой лужайки будет равна 12.

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

Пример правильного (по форме записи) выражения: 2\* (*b* −*a*).

***Система оценивания***

Баллы начисляются за правильное решение следующим образом:

**5 баллов** – записан только правильный ответ.

**8 баллов** – записан ответ с объяснением хода решения, но выражение не упрощено.

**10 баллов** – записан правильный ответ с объяснением хода решения.

**Решение**

Посчитаем общее число горизонтальных и вертикальных отрезков длины 1 между клетками на рисунке лужайки. Их число равно *a* × (*b* −1) *+ b* × (*a* −1).

После того, как будет поставлено ограждение, отрезков, в которых не будет поставлено ограждение, останется *a* × *b* −1, так как получится цепочка из *a* × *b* клеток, между двумя соседними клетками в цепочке ограждение отсутствует.

Таким образом, ответом является выражение *a* \*(*b* − 1) + *b* \*(*a* − 1) − (*a* \* *b* − 1). Это выражение можно упростить до вида (*a* − 1) \* (*b* − 1), которое можно просто получить подбором, рассматривая примеры для разных значений *a* и *b*.

**Задание №3. Возведение в степень (10 баллов)**

Дядя Федор решал математическую задачу, в которой необходимо было возвести данное число *a* в некоторую целочисленную степень *n*. Но его калькулятор умеет только перемножать числа. Например, можно вычислить *a*2= *a* × *a*, затем можно вычислить *a*3 = *a*2 × *a* или *a*4 = *a*2 × *a*2.

Можно по-разному организовать вычисление значения *an*. Например, вычислить *a*5 можно за 4 умножения:

1) *a*2 = *a* × *a*,

2) *a*3 = *a*2 × *a*,

3) *a*4 = *a*3 × *a*,

4) *a*5 = *a*4 × *a*.

Но можно вычислить *a*5 всего лишь за 3 умножения:

1) *a*2 = *a* × *a*,

2) *a*3 = *a*2 × *a*,

3) *a*5 = *a*3 × *a*2.

Вам необходимо определить, за какое минимальное число умножений Дядя Федор может 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, проделанные 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 четырех 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.

***Система оценивания***

Баллы начисляются за правильное решение следующим образом:

**3 балла** – записан ответ для вычисления *a*7

**5 баллов** – записан ответ для вычисления *a*7 *, a*15

**7 баллов** – записан ответ для вычисления *a*7 *, a*15, *a*23

**10 баллов** – записан ответ для вычисления *a*7 *, a*15, *a*23, *a*63

**Решение**

В этой задаче также возникает желание использовать двоичную систему счисления, как и в алгоритме быстрого возведения в степень. Например, чтобы вычислить *a*7 степень, необходимо вычислить *a*2и *a*4, а затем перемножить *a*, *a*2, *a*4. Всего понадобится 4 умножения. Для вычисления *a*15 , *a*23 , *a*63 через двоичную систему счисления понадобится 6, 7 и 10 умножений соответственно.

Но есть способ вычислить *a*15 , *a*23 , *a*63 используя меньшее число умножений — 5, 6 и 8. Пример наилучшего решения.

2 3 5 7
2 3 6 12 15
2 3 5 10 20 23
2 3 6 12 15 30 60 63

**Задание №4. Улица (10 баллов)**

По одну сторону улицы Почтовой в деревне Простоквашино находятся дома с нечетными номерами (1, 3, 5, …), по другую сторону – с четными (2, 4, 6, …). Дом № 1 находится напротив дома № 2, дом № 3 – напротив дома № 4 и т. д. До соседнего дома нужно идти вдоль по улице одну минуту, неважно, с какой стороны улицы он находится (то есть от дома № 1 нужно идти одну минуту как до дома № 3, так и до дома № 4). До дома, стоящего напротив, идти не нужно.



Почтальон Печкин вышел на улицу из дома номер *A* и должен доставить посылку в дом номер *B*.

Составьте программу, которая определит, сколько минут ему нужно идти вдоль по улице.

Программа получает на вход два различных целых положительных числа *A* и *B*, где A<B, не превосходящие 2×109, – номера домов. Программа должна вывести одно число – искомое количество минут.

***Пример входных и выходных данных***

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 18 | 3 |

***Система оценивания***

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

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 1000, будет оцениваться в **8 баллов**.

Решение, правильно работающее для всех случаев, описанных в условии задачи,будет оцениваться в **10 баллов**.

**Решение**

Введем систему координат, направленную вдоль улицы. Пусть у дома номер 1 и 2

координата равна 1, у домов номер 3 и 4 координата равна 2 и т. д. Если номер дома равен k, то его координата будет равна (k + 1) div 2 (синтаксис соответствует языку Pascal), здесь используется операция целочисленного деление – div.

Тогда если координата одного дома равна s, а другого дома — t, то расстояние между

этими домами будет равно abs(s – t).

***Пример решения на языке Pascal:***

var A, B, N: longint;

begin

readln (A); readln (B);

N:= abs((A+1) div 2 - (B+1) div 2);

writeln (N);

end.

**Всероссийская олимпиада школьников 2019-2020**

**Школьный этап**

**Информатика**

**9-11 класс**

***Продолжительность – 60 минут***

***Максимальный балл – 400***

**Задание №1. Улица (100 баллов)**

По одну сторону улицы Почтовой в деревне Простоквашино находятся дома с нечетными номерами (1, 3, 5, …), по другую сторону – с четными (2, 4, 6, …). Дом № 1 находится напротив дома № 2, дом № 3 – напротив дома № 4 и т. д. До соседнего дома нужно идти вдоль по улице одну минуту, неважно, с какой стороны улицы он находится (то есть от дома № 1 нужно идти одну минуту как до дома № 3, так и до дома № 4). До дома, стоящего напротив, идти не нужно.



Почтальон Печкин вышел на улицу из дома номер *A* и должен доставить посылку в дом номер *B*.

Составьте программу, которая определит, сколько минут ему нужно идти вдоль по улице.

Программа получает на вход два различных целых положительных числа *A* и *B*, где A<B и не превосходящие 2×109, – номера домов. Программа должна вывести одно число – искомое количество минут.

***Пример входных и выходных данных***

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 18 | 3 |

***Система оценивания***

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 100, будет оцениваться в **60 баллов**.

Решение, правильно работающее для всех случаев, описанных в условии задачи, будет оцениваться в **100 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 Phyton** | **Язык программирования Pascal** |
| a=int(input())b=int(input())…n=…print(n) | var a, b, n:longint;beginreadln(a); readln(b);…n:=…writeln(n);end. |
| **Язык программирования C** | **Язык программирования C++** |
| #include<stdio.h>int main() {int a, b, n;scanf(“%d%d”, &a, &b);…n=…printf(“%d\n”, n);return 0;} | #include <iostream>using namespace std;int main() {int a, b, n;cin >> a >> b;…n=…cout << n << endl;return 0;} |
| **Язык программирования C#** | **Язык программирования Java** |
| using System;using System.IO;class Program {static void Main() {int a, b, n;a=int.Parse(Console.ReadLine());b=int.Parse(Console.ReadLine());…n=…Console.WriteLine("{0}", n);}} | import java.io.\*;import java.util.\*;class solver{  public static void main(String[] args) throws IOException  {DataInputStream in=new DataInputStream(System.in);int a, b, n;a=Integer.parseInt(in.readLine());b=Integer.parseInt(in.readLine());…n=…System.out.println(n);}} |

**Задание №2. Надёжное крепление (100 баллов)**

Уличный рекламный щит прикреплён к опоре при помощи трёх креплений. Первое крепление может выдерживать ветер, скорость которого не превосходит A м/с, второе крепление – B м/с, третье – C м/с. Сам щит будет надёжно закреплён, если как минимум два крепления из трёх выдерживают ветер данной скорости. Определите максимальную скорость ветра, которую выдержит данный щит.

Программа получает на вход три целых положительных числа A, B, C, не превосходящие 2х109, - допустимые скорости ветра, которые выдерживают три крепления щита. Программа должна вывести одно число – максимальную скорость ветра, которую выдержит щит.

***Пример входных и выходных данных***

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 281510 | 15 |

***Система оценивания***

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 100, будет оцениваться в **60 баллов**.

Решение, правильно работающее для всех случаев, описанных в условии задачи, будет оцениваться в **100 баллов**.

**Задание №3. Парад (100 баллов)**

В параде принимают участие M военных. Командование парада решило, что наиболее эффектное посторенние военных – в форме квадрата, то есть число участников построения должно быть точным квадратом. Но поскольку число M может не быть точным квадратом, разрешается разбить военных на несколько полков, каждый из которых строится в форме квадрата. Для красоты все полки должны быть одинакового размера, также командование парада хочет, чтобы размер каждого полка был как можно больше. Определите максимально возможный размер полка.

Программа получает на вход одно целое положительное число M, не превосходящее 2х109, количество участников парада. Программа должна вывести одно число – максимально возможный размер полка.

***Пример входных и выходных данных***

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 180 | 36 |

***Система оценивания***

Решение, правильно работающее только для случаев, когда M не превосходит 100, будет оцениваться в **40 баллов**.

Решение, правильно работающее только для случаев, когда M не превосходит 10000, будет оцениваться в **60 баллов**.

Решение, правильно работающее для всех случаев, описанных в условии задачи, будет оцениваться в **100 баллов**.

**Задание №4. Ряд чисел (100 баллов)**

Легенда гласит, что Карл Фридрих Гаусс, учась в школе, смог быстро посчитать сумму целых чисел от 1 до 100, заметив, что 1+100=2+99=…=50+51. Теперь решите задачу посложнее: можно ли перед каждым из чисел от 1 до N расставить знаки «+» или «-» так, чтобы сумма получившихся чисел была равна 0? Например, для N=3 сумма -1-2+3 будет равна 0, а для N=2 этого сделать нельзя.

Программа получает на вход целое положительное число N, не превосходящее 105. Программа должна вывести «POSSIBLE», если задача имеет решение. Если задача не имеет решения для данного N, нужно вывести одно слово «IMPOSSIBLE».

***Пример входных и выходных данных***

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 3 | POSSIBLE |
| 2 | IMPOSSIBLE |

***Система оценивания***

Решение, правильно работающее только для случаев, когда N не превосходит 20, будет оцениваться в **40 баллов**.

Решение, правильно работающее только для случаев, когда N не превосходит 1000, будет оцениваться в **60 баллов**.

Решение, правильно работающее для всех случаев, описанных в условии задачи, будет оцениваться в **100 баллов**.

**Всероссийская олимпиада школьников 2019-2020**

**Школьный этап**

**Информатика**

**9-11 класс**

***Продолжительность – 60 минут***

***Максимальный балл – 400***

**Задание №1. Улица (100 баллов)**

По одну сторону улицы Почтовой в деревне Простоквашино находятся дома с нечетными номерами (1, 3, 5, …), по другую сторону – с четными (2, 4, 6, …). Дом № 1 находится напротив дома № 2, дом № 3 – напротив дома № 4 и т. д. До соседнего дома нужно идти вдоль по улице одну минуту, неважно, с какой стороны улицы он находится (то есть от дома № 1 нужно идти одну минуту как до дома № 3, так и до дома № 4). До дома, стоящего напротив, идти не нужно.



Почтальон Печкин вышел на улицу из дома номер *A* и должен доставить посылку в дом номер *B*.

Составьте программу, которая определит, сколько минут ему нужно идти вдоль по улице.

Программа получает на вход два различных целых положительных числа *A* и *B*, где A<B, не превосходящие 2×109, – номера домов. Программа должна вывести одно число – искомое количество минут.

***Пример входных и выходных данных***

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 18 | 3 |

***Система оценивания***

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 100, будет оцениваться в **60 баллов**.

Решение, правильно работающее для всех случаев, описанных в условии задачи, будет оцениваться в **100 баллов**.

***Решение***

Введём систему координат, направленную вдоль улицы. Пусть у дома номер 1 и 2 координата равна 1, у домов номер 3 и 4 координата равна 2 и т. д. Если номер дома равен k, то его координата будет равна (k + 1) // 2. Под «//» здесь подразумевается целочисленное деление (синтаксис соответствует языку Python), в языке Pascal это операция div.

Тогда если координата одного дома равна s, а другого дома — t, то расстояние между этими домами будет равно abs(s – t).

Пример решения на языке Python.

a = int(input())

b = int(input())

print(abs((a + 1) // 2 - (b + 1) // 2))

**Задание №2. Надёжное крепление (100 баллов)**

Уличный рекламный щит прикреплён к опоре при помощи трёх креплений. Первое крепление может выдерживать ветер, скорость которого не превосходит A м/с, второе крепление – B м/с, третье – C м/с. Сам щит будет надёжно закреплён, если как минимум два крепления из трёх выдерживают ветер данной скорости. Определите максимальную скорость ветра, которую выдержит данный щит.

Программа получает на вход три целых положительных числа A, B, C, не превосходящие 2х109, - допустимые скорости ветра, которые выдерживают три крепления щита. Программа должна вывести одно число – максимальную скорость ветра, которую выдержит щит.

***Пример входных и выходных данных***

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 281510 | 15 |

***Система оценивания***

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 100, будет оцениваться в **60 баллов**.

Решение, правильно работающее для всех случаев, описанных в условии задачи, будет оцениваться в **100 баллов**.

***Решение***

Ответом является среднее из трёх чисел, если их упорядочить. Например, если для чисел A, B, C выполнены неравенства A ≤ B ≤ C, то ответом будет значение B. Реализовать алгоритм нахождения медианы можно разными способами. Пример решения, разбирающего разные способы упорядочить числа A, B, C.

a = int(input())

b = int(input())

c = int(input())

if a <= b <= c or c <= b <= a:

print(b)

elif b <= a <= c or c <= a <= b:

print(a)

else:

print(c)

В следующем примере используется функция min для поиска наименьшего из трёх чисел, функция max для поиска наибольшего из трёх чисел, а медианой является ставшееся из трёх чисел, которое можно найти, вычтя из суммы данных чисел значение наибольшего и наименьшего числа.

a = int(input())

b = int(input())

c = int(input())

print(a + b + c — min(a, b, c) — max(a, b, c))

Можно воспользоваться стандартной функцией сортировки и вывести значение среднего элемента упорядоченного списка.

a = int(input())

b = int(input())

c = int(input())

print(sorted([a, b, c])[1])

**Задание №3. Парад (100 баллов)**

В параде принимают участие M военных. Командование парада решило, что наиболее эффектное посторенние военных – в форме квадрата, то есть число участников построения должно быть точным квадратом. Но поскольку число M может не быть точным квадратом, разрешается разбить военных на несколько полков, каждый из которых строится в форме квадрата. Для красоты все полки должны быть одинакового размера, также командование парада хочет, чтобы размер каждого полка был как можно больше. Определите максимально возможный размер полка.

Программа получает на вход одно целое положительное число M, не превосходящее 2х109, количество участников парада. Программа должна вывести одно число – максимально возможный размер полка.

***Пример входных и выходных данных***

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 180 | 36 |

***Система оценивания***

Решение, правильно работающее только для случаев, когда M не превосходит 100, будет оцениваться в **40 баллов**.

Решение, правильно работающее только для случаев, когда M не превосходит 10000, будет оцениваться в **60 баллов**.

Решение, правильно работающее для всех случаев, описанных в условии задачи, будет оцениваться в **100 баллов**.

***Решение***

В этой задаче нужно просто перебрать все возможные ответы. Нам нужно найти такое максимальное число K, которое было бы делителем числа M и полным квадратом: K = d2 . Между тем решение, которое будет перебирать все числа от 1 до M и проверять, является ли оно подходящим ответом, будет работать слишком долго. Такое решение будет набирать 60 баллов.

Для того, чтобы написать решение на 100 баллов, необходимо сразу же перебирать квадраты чисел (то есть значения d), до тех пор, пока рассматриваемый квадрат не станет больше числа M.

Пример решения на языке Python.

n = int(input())

ans = 1

d = 2

while d \* d <= n:

if n % (d \* d) == 0:

ans = d \* d

d += 1

print(ans)

**Задание №4. Ряд чисел (100 баллов)**

Легенда гласит, что Карл Фридрих Гаусс, учась в школе, смог быстро посчитать сумму целых чисел от 1 до 100, заметив, что 1+100=2+99=…=50+51. Теперь решитезадачу посложнее: можно ли перед каждым из чисел от 1 до N расставить знаки «+» или «-» так, чтобы сумма получившихся чисел была равна 0? Например для N=3 сумма -1-2+3 будет равна 0, а для N=2 этого сделать нельзя.

Программа получает на вход целое положительное число N, не превосходящее 105. Программа должна вывести «POSSIBLE», если задача имеет решение. Если задача не имеет решения для данного N, нужно вывести одно слово «IMPOSSIBLE».

***Пример входных и выходных данных***

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 3 | POSSIBLE |
| 2 | IMPOSSIBLE |

***Система оценивания***

Решение, правильно работающее только для случаев, когда N не превосходит 20, будет оцениваться в **40 баллов**.

Решение, правильно работающее только для случаев, когда N не превосходит 1000, будет оцениваться в **60 баллов**.

Решение, правильно работающее для всех случаев, описанных в условии задачи, будет оцениваться в **100 баллов**.

***Решение***

К идее решения задачи можно прийти, если попробовать построить ответы для маленьких значений N. При N = 1 и N = 2 задача не имеет решения, при N = 3 и при N = 4 есть решения («++-» и «+--+»). Далее заметим, что для любых 4 подряд идущих чисел (не только для чисел 1, 2, 3, 4) можно расставить знаки требуемым образом: «+--+». Поэтому если N делится на 4, то задача имеет решение. Если N даёт остаток 3 при делении на 4, то задача тоже имеет решение. Если N даёт остаток 1 или 2 при делении на 4, то задача не имеет решения. Как доказательство можно заметить, что в этом случае сумма чисел от 1 до N будет нечётной, а замена знака одного слагаемого не меняет чётность суммы, поэтому при любой расстановке знаков сумма будет нечётной.

В решении необходимо разобрать случаи различного остатка от деления N на 4 и для каждого случая вывести свой вариант ответа.

Пример решения на языке Python.

n = int(input())

if n % 4 == 1 or n % 4 == 2:

print("IMPOSSIBLE")

else:

print("POSSIBLE")