

Условия задач 7-8 класс

В первых трех задачах решением являются текстовые данные заданного формата. Эти данные можно получать любым способом: набрать вручную в текстовом редакторе, использовать электронную таблицу или с помощью программы на любом языке программирования.

В последних трех задачах решением является программа, созданная на любом языке программирования. Баллы начисляются за каждый пройденный тест. Решение получает максимальное количество баллов в случае, когда все тесты будут пройдены.

Данные в программе можно считывать с консоли или из текстового файла input.txt. Результат можно выводить на консоль или в текстовый файл output.txt.

Каждая строка во входных данных завершается символом конца строки. Последняя строка во входном файле пустая.

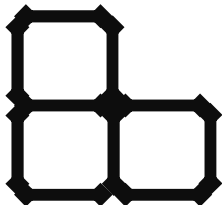
Максимальное время тестирования всех задач на любом тесте – 1 секунда.

1. «Лесенка из палочек» (100 баллов)

Второклассник Ваня еще до школы очень полюбил собирать различные картинки из счетных палочек. И вот теперь он решил собрать простую лесенку из квадратов. Если он соберет только одну ступеньку – из одного квадрата, то ему понадобится 4 палочки.



Если Ваня соберет лесенку из двух ступенек, то ему понадобится 10 палочек.



Ваня задумался: а как посчитать, сколько палочек ему понадобится, если он захочет собрать лесенку из n ступенек? Помогите Ване составить формулу для подсчета требуемого количества палочек.

Ответом на эту задачу является некоторое выражение, которое может содержать целые числа, переменную (записываемую английской буквой), 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 a неверная, нужно писать « $2 * n$ ».

Пример правильной формы записи ответа: $n / 2 + (n * 3 - 5) * 2$

Данная задача оценивается следующим образом:

Если записанная Вами формула возвращает правильное решение на всех данных, то Вы получаете полный балл. Если по Вашей формуле верный ответ получается только для некоторых данных, то Вы получите частичный балл за решение.

2. «Последовательности» (100 баллов)

Первая последовательность состоит из единственного числа 1.

Вторая последовательность получается заменой чисел первой последовательности на дополнительные до 10 (т.е. числами, равными разности 10 и чисел первой последовательности). Затем дописывается первая последовательность без изменений. И в конец добавляется последняя цифра номера последовательности. Таким образом получим вторую последовательность. Дополнение до 10 числа 1 равно $10 - 1 = 9$. Получим последовательность 912.

Третья последовательность получается заменой чисел второй последовательности на дополнительные до 10. Затем дописывается вторая последовательность без изменений. И в конец добавляется последняя цифра номера последовательности. Получим последовательность 1989123.

Каждая следующая последовательность получается заменой чисел предыдущей последовательности на дополнительные до 10. Затем дописывается предыдущая последовательность без изменений. И в конец добавляется последняя цифра номера последовательности.

Вам необходимо найти, какие цифры стоят на 1, 10, 16, 31, 119, 140, 141, 142, 187, 210 позициях в двадцатой последовательности.

Выходные данные:

Десять целых числа через пробел – цифры, которые находятся на указанных позициях.

Пример:

Входные данные	Выходные данные
В третьей строке на позициях 1, 2, 5, 7	1 9 1 3

Обратите внимание, что вам необходимо определить цифры на 10 указанных в условии позициях в двадцатой строке.

Данная задача оценивается следующим образом:

Полный балл ставится, если указаны верно все цифры. Если указаны верно только некоторые, то будет выставлен частичный балл.

3. «Возведение в степень»(100 баллов)

У исполнителя «Умножитель» есть три регистра памяти A , B и C , в которые он может записывать целые числа и выполнять над ними операции умножения. Обозначим R любой из регистров A , B или C . Система команд данного исполнителя следующая:

- $read(R)$ – получает от пользователя целое число и записывает его в регистр R .
- $print(R)$ – выводит число, которое содержится в регистре R , на экран.
- $R_2 = R_1$ – копирует значение регистра R_1 в регистр R_2 . Значение регистра R_1 не изменяется. Регистры R_1 и R_2 должны быть разными регистрами.
- $R_1 = R_1 * R_2$ – умножает числа, которые находятся в регистрах R_1 и R_2 и результат записывает в R_1 (имена регистров R_1 и R_2 не могут совпадать). Например, в регистре A находится число 3, а в регистре B находится число 2. Тогда команда $A = A * B$, получает в регистре A число 6, содержимое регистра B не меняется. А команда $B = B * A$ получает в регистре B число 6, содержимое регистра A не меняется.

Возведение в степень A (30 баллов)

Необходимо составить алгоритм, состоящий из наименьшего количества команд, по которому «Умножитель» вычислит x^{15} . Значение x необходимо получить у пользователя, а полученное значение вывести на экран.

Выходные данные:

Последовательность команд исполнителя, рассмотренных в условии задачи, которая возводит заданное пользователем число в **15-ю степень**. Не забывайте, что у исполнителя есть только регистры A , B и C . Каждая команда пишется с новой строки. Количество команд в предложенном алгоритме должно быть **минимально возможным**.

Пример:

Пусть исполнителю нужно вычислить x^3 . Алгоритм для данных вычислений будет такой:

```
read(A)
B = A
B = B * A
B = B * A
print(B)
```

Данная задача оценивается следующим образом:

30 баллов ставятся, если указан верный алгоритм с минимальным количеством команд. Частичный балл ставится, если указан алгоритм не минимальной длины.

Возведение в степень В (70 баллов)

Необходимо составить алгоритм, состоящий из наименьшего количества команд, по которому «Умножитель» вычислит x^{70} . Значение x необходимо получить у пользователя, а полученное значение вывести на экран

Выходные данные:

Последовательность команд исполнителя, рассмотренных в условии задачи, которая возводит заданное пользователем число в 70-ю степень. Не забывайте, что у исполнителя есть только регистры A , B и C . Каждая команда пишется с новой строки. Количество команд в предложенном алгоритме должно быть **минимально возможным**.

Данная задача оценивается следующим образом:

70 баллов ставятся, если указан верный алгоритм с минимальным количеством команд. Частичный балл ставится, если указан алгоритм не минимальной длины.

4. «Уравнения» (100 баллов)

Учителю нужно составить самостоятельную работу для проверки умения решать уравнения вида $a \cdot x = b$, где a , b – целые числа. Учитель нашел много подобных уравнений. Среди найденных уравнений попадаются уравнения вида $x \cdot a = b$. Такие уравнения тоже подходят для самостоятельной работы. Но учитель хочет, чтобы единственный корень, если он существует был бы целым. Если уравнение имеет единственный нецелый корень, то такие уравнения учитель не хочет включать в самостоятельную работу. Уравнений много. Помогите учителю: напишите программу, которая будет решать подходящие уравнения или указывать, что данное уравнение не подходит для самостоятельной работы.

Входные данные:

В единственной строке записано уравнение вида $a * x = b$ или $x * a = b$, где a и b – коэффициенты уравнения ($1 \leq a, b \leq 1000$).

Выходные данные:

В единственной строке вывести один из следующих результатов:

- целое число – корень уравнения, если он существует;
- строку «NO», если уравнение не имеет корней;
- строку «ALL», если уравнение имеет бесконечно много корней;
- строку «WRONG», если уравнение не подходит.

Пример:

Входные данные	Выходные данные
$x * 2 = 4$	2
$0 * x = 7$	NO
$5 * x = 23$	WRONG

5. «Наборы пирожных» (100 баллов)

У хозяйки есть A штук пирожных безе и B штук пирожных эклеров. Все пирожные примерно одного размера. К приходу гостей на тарелку, на которую помещается не более C любых пирожных, выкладывают некоторое количество безе и некоторое количество эклеров. Тарелка при этом не обязательно должна быть заполнена полностью. Сколькими способами это можно сделать, если на тарелке должно быть хотя бы одно пирожное каждого вида?

Входные данные:

В первой строке записано одно натуральное число A – количество безе ($1 \leq A \leq 10^9$).

Во второй строке записано одно натуральное число B – количество эклеров ($1 \leq B \leq 10^9$).

В третьей строке записано одно натуральное число C – вместимость тарелки ($2 \leq C \leq 10^9$).

Выходные данные:

Одно целое число – количество способов выкладки пирожных.

Пример:

Входные данные	Выходные данные
2 4 3	3
1 3 10	3

Пояснение к примерам

В первом примере на тарелку вместимостью 3 можно положить 1 безе и 1 или 2 эклера, или 2 безе и 1 эклер. Получаем 3 варианта (1:1, 1:2, 2:1)

Во втором примере на тарелку вместимостью 10 можно положить 1 безе и 1, 2 или 3 эклера. Получаем 3 варианта (1:1, 1:2, 1:3).

Обратите внимание, что в данной задаче необходимо использовать 64-битный тип данных, например longlong в C++, int64 в Free Pascal, long в Java.

6. «Прибыль фирмы» (100 баллов)

Прибыль небольшой фирмы может быть очень нестабильной. В один месяц она есть, а в другой месяц оказываются только убытки. И вот один из предпринимателей задумался: «А за какой непрерывный промежуток времени суммарная прибыль фирмы была максимальной? Ну или хотя бы убытки были не такими большими, если уж прибыли не было». Напишите программу, которая по известной прибыли фирмы в каждый из N месяцев найдет максимальную суммарную прибыль, которую получала данная фирма за непрерывный интервал времени.

Входные данные:

В первой строке записано натуральное число N – количество месяцев ($1 \leq N \leq 10^5$).

В каждой из следующих N строк последовательно записаны натуральные числа a_i – прибыль фирмы в i -м месяце ($-10^6 \leq a_i \leq 10^6$).

Выходные данные:

Одно целое число – максимальная суммарная прибыль фирмы за непрерывный промежуток времени.

Пример:

Входные данные	Выходные данные
6 5 2 -1 3 -10 3	9

Пояснение к примеру

Максимальная прибыль достигнута в первые месяцы $5 + 2 - 1 + 3 = 9$

Обратите внимание, что в данной задаче необходимо использовать 64-битный тип данных, например longlong в C++, int64 в Free Pascal, long в Java.