Задачи для пересдачи

Входные данные для всех задач группы 1 записаны в файле input.txt или на стандартном вводе, результат работы нужно вывести в файл output.txt или на стандартный вывод.

В случае использования файлового ввода-вывода, повторное открытие и считывания файла запрещен.

1A: ЕГЭ-2013, наибольшее произведение, кратное 21

Решение должно использовать *O*(1) памяти и иметь сложность *O*(*N*).

По каналу связи передаётся последовательность из *N* целых положительных чисел. Количество чисел может быть очень велико.

Необходимо вычислить контрольное значение последовательности — наибольшее число R, удовлетворяющее следующим условиям:

1) R – произведение двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел, произведения различных элементов последовательности, равных по величине, допускаются);

2) R делится на 21.

Если такого числа R нет, то контрольное значение полагается равным 0.

Напишите эффективную, в том числе по используемой памяти, программу которая будет вычислять контрольное значение.

Программа получает набор натуральных чисел, по одному в строке, и должна вывести исходное контрольное значение.

Ввод	Вывод
70 21 997 7 9 300	21000

1D: ЕГЭ-2012, Камера хранения

Решение должно использовать *O*(*K*) памяти и иметь сложность *O*(*NK*), где *N* — количество пассажиров, *K* — количество ячеек.

На вход программе подаются сведения о пассажирах, желающих сдать свой багаж в камеру хранения на заранее известное время до полуночи. В первой строке дано количество ячеек в камере хранения *K*. Каждая из следующих строк имеет следующий формат:

<Фамилия> <время сдачи багажа> <время освобождения ячейки>

где <Фамилия> – строка, состоящая не более чем из 20 непробельных символов; <время сдачи багажа> – через двоеточие два целых числа, соответствующие часам (от 00 до 23 – ровно 2 символа) и минутам (от 00 до 59 – ровно 2 символа); <время освобождения ячейки> имеет тот же формат. <Фамилия> и <время сдачи багажа>, а также <время сдачи багажа> и <время

освобождения ячейки> разделены одним пробелом. Время освобождения больше времени сдачи.

Сведения отсортированы в порядке времени сдачи багажа. Если два пассажира приходят в одно время, то сначала обслуживается тот пассажир, который указан раньше во входных данных. Каждому из пассажиров в камере хранения выделяется свободная ячейка с минимальным номером. Если в момент сдачи багажа свободных ячеек нет, то пассажир уходит, не дожидаясь освобождения одной из них.

Требуется написать программу, которая будет выводить на экран для каждого пассажира номер ему предоставленной ячейки (можно сразу после ввода данных очередного пассажира). Если ячейка пассажиру не предоставлена, то его фамилия не печатается.

Ввод	Вывод
10 Иванов 09:45 12:00 Петров 10:00 11:00 Сидоров 12:00 13:12	Иванов 1 Петров 2 Сидоров 1
1 Иванов 09:45 12:00 Петров 10:00 11:00 Сидоров 12:00 13:12	Иванов 1 Сидоров 1

1G: ЕГЭ-2009, проходной балл

Решение должно иметь сложность *O*(*N*+*K*) и использовать *O*(*K*) памяти (*K* — максимально возможный балл абитуриента, *N* — число абитуриентов). Чтение входных данных должно быть построчным, нельзя создавать массив, размер которого зависел бы от входных данных.

Для поступления в вуз абитуриент должен предъявить результаты трех экзаменов в виде ЕГЭ, каждый из них оценивается целым числом от 0 до 100 баллов. При этом абитуриенты, набравшие менее 40 баллов (неудовлетворительную оценку) по любому экзамену из конкурса выбывают. Остальные абитуриенты участвуют в конкурсе по сумме баллов за три экзамена.

В конкурсе участвует *N* человек, при этом количество мест равно *K*. Определите проходной балл, то есть такое количество баллов, что количество участников, набравших столько или больше баллов не превосходит *K*, а при добавлении к ним абитуриентов, набравших наибольшее количество баллов среди непринятых абитуриентов, общее число принятых абитуриентов станет больше *K*.

Программа получает на вход количество мест *K*. Далее идут строки с информацией об абитуриентах, каждая из которых состоит из фамилии, имени и трех чисел от 0 до 100, разделенных пробелами.

Программа должна вывести проходной балл в конкурсе. Выведенное значение должно быть минимальным баллом, который набрал абитуриент, прошедший по конкурсу.

Также возможны две ситуации, когда проходной балл не определен.

Если будут зачислены все абитуриенты, не имеющие неудовлетворительных оценок, программа должна вывести число 0.

Если количество абитуриентов, имеющих равный максимальный балл больше чем *K*, программа должна вывести число 1.

Ввод	Вывод

5 Иванов Сергей 70 70 70 Сергеев Петр 100 100 0 Петров Василий 70 60 70 Васильев Андрей 70 60 70 Андреев Денис 100 30 100 Денисов Роман 50 50 50 Романов Григорий 60 70 70 Григорьев Станислав 50 50 50 Станиславский Иван 40 40 40	200
1 Иванов Сергей 40 40 40 Сергеев Петр 100 100 39	0
1 Иванов Сергей 60 60 60 Сергеев Петр 100 40 40	1

1K: ЕГЭ-2014, минимальное произведение, на расстоянии не менее 6

Решение должно использовать *O*(1) памяти и иметь сложность *O*(*N*).

На спутнике «Фотон» установлен прибор, предназначенный для измерения энергии космических лучей. Каждую минуту прибор передаёт по каналу связи неотрицательное вещественное число — количество энергии, полученной за последнюю минуту, измеренное в условных единицах. Временем, в течение которого происходит передача, можно пренебречь.

Необходимо найти в заданной серии показаний прибора минимальное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут.

Каждая из *N* строк входных данных содержит одно неотрицательное вещественное число – очередное показание прибора, $N \geq 7$. Входные числа не превосходят 40.000 каждое.

Программа должна вывести одно число – описанное в условии произведение.

Ввод	Вывод
12 45 5.5 4 25 23 21 20 10 12 26	48.0

1N: ЕГЭ-2019, максимальная сумма, делящаяся на 120

На вход программы поступает последовательность из *n* целых положительных чисел. Рассматриваются все пары элементов последовательности *ai* и *aj*, такие что $i<j$ и $ai>aj$ (первый элемент пары больше второго; *i* и *j* — порядковые номера чисел в последовательности входных данных). Среди пар, удовлетворяющих этому условию, необходимо найти и напечатать пару с максимальной суммой элементов, которая делится на $m=120$. Если среди найденных пар максимальную сумму имеют несколько, то можно напечатать любую из них.

В качестве результата программа должна напечатать элементы искомой пары. Если таких пар несколько, можно вывести любую из них.

Если такой пары в последовательности нет, программа должна вывести два нуля.

Ввод	**Вывод**
60 140 61 100 300 59	140 100
1 2 120	0 0

В задачах группы 2 (если не оговорено иное) нельзя использовать арифметические операторы сложения, умножения, вычитания, деления, взятия остатка. Вместо них используем побитовые операторы &, |, ~, ^, <<, >>.

2F: Инвертировать бит

Дано целое число A и целое неотрицательное число k. Выведите число, которое получается из числа A инвертированием k-го бита.

Ввод	Вывод
15 2	11

2G: Обнулить бит

Дано целое число A и целое неотрицательное число k. Выведите число, которое получается из числа A установкой значения k-го бита равному 0.

Ввод	Вывод
14 1	12

2H: Вырезать бит

Дано целое число A и целое неотрицательное число k. Выведите число, которое получается из числа A удалением k-го бита. Старшие биты, чьи номера больше чем k, сдвигатся вправо.

В решении можно использовать операции +1 и -1.

Ввод	Вывод
21 2	9

2I: Сменить регистр символа

Программа получает на вход символ, являющийся заглавной или строчной буквой латинского алфавита. Используя битовые операции с ASCII-кодом символа, поменяйте регистр этого символа.

Ввод	Вывод
A	a
a	A

2J: Обрезать старшие биты

Дано целое число A и натуральное число k. Выведите число, которое состоит только из k последних бит числа A (то есть обнулите все биты числа A, кроме последних k).

Ввод	Вывод
126 3	6

2K: 2^k+2^n — 2

Даны два **возможно равных** целых неотрицательных числа: k и n. Вычислите 2^k+2^n.

Ввод	Вывод
0 1	3
1 1	4

В каждом задании группы 3, на проверку необходимо сдать документ, подготовленный в среде электронных таблиц: Microsoft Excel или LibreOffice Calc. При этом файл необходимо сдавать в формате Microsoft Excel 2003 или более новый (расширение .xlsx), даже если вы работаете в Libre Office Calc необходимо при сохранении указать формат Microsoft Excel.

3D: Графики степенной функции

На одной диаграмме постройте графики функции *y=xn* для *n*=0.5,1,1.5,2,3. Образец.

Используйте файл с шаблоном таблицы.

Блок A2:A32 заполните значениями *x* от 0 до 3 с шагом 0.1. В блоке B2:F32 запишите значение функции *xn*, где значение *x* берётся из столбца A, а значение *n* из строки 1. Значения в блоке B2:F32 должны быть получены единой формулой, скопированной по всем ячейкам блока.

Для вычисления значения степени *ab* используется функция POWER(a; b) (русское название — СТЕПЕНЬ) или бинарный оператор “^”.

Сдайте на проверку файл с построенным графиком для значений *n*=0.5,1,1.5,2,3. На одной диаграмме должны быть все пять графиков. При изменении значения *n* должен изменяться вид графика.

3E: Треугольные числа

Заполните блок A1:A100 числами от 1 до 100. В блок B1:B100 запишите треугольные числа, используя функцию SUM(СУММ), которая считает сумму чисел в блоке (в нескольких блоках, если ей передать несколько аргументов). *n*-е треугольное число есть сумма 1+2+...+*n*. В блок C1:C100 запишите тетраэдрические числа — число точек в тетраэдре со стороной *n*. *n*-е тетраэдрическое число равно сумме первых *n* треугольных чисел. Наконец, в блоке D1:D100 запишите четырёхмерные тетраэдрические числа (сумма первых *n* тетраэдрических чисел).

Образец.

Все значения в блоке B1:D100 должны быть получены при помощи единой формулы.

3K: Оценки — 2

Скачайте файл с таблицей.

1. Подсчитайте средний балл по русскому языку для всех учащихся. Ответ запишите в ячейку C1311.
2. Посчитайте средний балл по математике для тех учащихся, которые имеют удовлетворительную оценку по математике. Ответ запишите в ячейку C1312.
3. Посчитайте средний балл по информатике для учащихся 9 класса. Ответ запишите в ячейку C1313.
4. Подсчитайте средний балл по информатике для тех учащихся, которые имеют оценку по математике выше, чем по русскому языку. Ответ запишите в ячейку C1314.
5. Подсчитайте средний балл по русскому языку для всех школьников, не имеющих двоек по всем предметам. Ответ запишите в ячейку C1315.
6. Подсчитайте средний балл по математике для учащихся 10 класса, которые не имеют двоек по всем предметам. Ответ запишите в ячейку C1316.

3L: Количество участников регионального этапа

В этой задаче используются данные о результатах регионального этапа всероссийской олимпиады школьников по всем предметам в г. Москве, взятых отсюда.

Скачайте файл с таблицей.

На втором листе (data) хранится информация об участниках регионального этапа по всем предметам. Посчитайте количество школьников, участвовавших в региональном этапе (один школьник считается один раз, если он участвовал в нескольких олимпиадах). Участники идентифицируются по логину. С данными на этом листе можете делать, что угодно.

Ответ запишите в ячейку A1 первого листа (answer). Этот лист должен остаться первым. В этой ячейке должна быть записана какая-либо формула, возможно, просто ссылка на ячейку другого листа, но просто ввести руками число в эту ячейку нельзя.

В 4 группе задач используются понятия систем счисления с основанием больше 10, цифры записываются так: 0, 1, 2, ..., 9, A, B, C, ...

В тексте программ на языке Python можно использовать целочисленные константы, записанные в двоичной (префикс 0b), восьмеричной (префикс 0o) и шестнадцатеричной (префикс 0x) системах счисления. После указанного префикса идут цифры, которые в двоичной системе счисления могут быть только 0 или 1, в восьмеричной — от 0 до 7, в шестнадцатеричной — от 0 до 9, а также буквы латинского алфавита от A до F (могут быть как строчными, так и заглавными). Например, десятичной число 179 можно задать несколькими способами.

A = 179
A = 0b10110011
A = 0o263
A = 0xB3
Если вы знаете стандартные функции языка Python для перевода представления чисел между различными системами счисления, то этими функциями пользоваться нельзя. Также нельзя использовать функции типа eval, exec и т.д.

Если программа выводит результат в системе счисления с основанием больше 10, то цифры записываются так: 0, 1, 2, ..., 9, A, B, C, ...

В решениях нельзя использовать списки, словари, строки типа "ABCDEF" и т.д.

4G: Из любой в любую

Напишите программу, переводящую запись числа между двумя произвольными системами счисления.

На вход программа получает три величины: *n*, *A*, *k*, где *n* и *k* – натуральные числа от 2 до 36: основания системы счисления, *A* – число, записанное в в системе счисления с основанием *n*, $A<2^{31}$.

Необходимо вывести значение *A* в системе счисления с основанием *k* без лидирующих нулей.

Решение должно содержать две функции перевода — из числа в произвольной системе счисления, записанного в переменной типа string в переменную типа int и обратно. Сдайте на проверку всю программу.

Ввод	Вывод
2 101111 16	2F
10 35 36	Z

4J: Из уравновешенной троичной в int

В уравновешенной троичной системе счисления используется основание 3 и три цифры: 0, 1 и -1. Цифру -1 будем обозначать знаком $. Достоинство уравновешенной троичной системы счисления: простота хранения отрицательных чисел и удобство нахождения числа, противоположному данному.

Вот как записываются небольшие числа в уравновешенной троичной системе счисления:

Десятичная	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9
Уравнов. троичная	$00	$01	1	$10	$11	$$	$0	$1	$	0	1	1$	10	11	1$$	1$0	1$1	10$	100

Подробней о уравновешенной троичной системе счисления можно прочитать в Википедии (статья Троичная система счисления, там используется термин "троичная симметричная система счисления").

Дана запись числа в уравновешенной троичной системе счисления. Переведите её в значение типа int.

Решение оформите в виде функции int ter2int(string s). Сдайте на проверку только тело функции.

Вызов функции	Возвращаемое значение
ter2int("$01")	-8

4L: Из int в уравновешенную троичную

Дано целое число от −2×109 до 2×109. Найдите его представление в уравновешенной троичной системе счисления без лидирующих нулей (за исключением числа 0, запись которого имеет вид 0).

Решение оформите в виде функции string int2ter(int n). Сдайте на проверку только тело функции.

Вызов функции	Возвращаемое значение
int2ter(-8)	"$01"

4P: Инкремент в уравновешенной троичной системе счисления

Реализуйте инкремент числа в уравновешенной троичной системе счисления.

Решение оформите в виде функции void inc_ter(string & n). Сдайте на проверку только тело функции.

Входная строка может иметь длину до 100000 символов.

Вызов функции	Вывод
string n = "$01"; inc_ter(n); cout << n << endl;	1

4Q: Декремент в уравновешенной троичной системе счисления

Реализуйте декремент числа в уравновешенной троичной системе счисления.

Решение оформите в виде функции void dec_ter(string & n). Сдайте на проверку только тело функции.

Входная строка может иметь длину до 100000 символов.

Вызов функции	Вывод
string n = "1"; dec_ter(n); cout << n << endl;	$01

4W: Шестнадцатеричное вычитание

Реализуйте алгоритм вычитания двух чисел, записанных в шестнадцатеричной системе счисления.

Решение оформите в виде функции string dif_hex(string n1, string n2).

Сложность решения: линейная от длины входных чисел (длина чисел до 100000 знаков, ограничение по времени — 1 секунда).

Вызов функции	Возвращаемое значение
dif_hex("100", "F")	"F1"