

Памятка участникам заключительного тура

1. Заключительный тур проводится очно на заранее согласованных площадках проведения. Каждому участнику во время проведения заключительного тура предоставляется персональный компьютер с установленным программным обеспечением (ОС, компилятор, среда разработки) согласно предварительным заявкам. Сеть Интернет недоступна. Использование любых справочных материалов, за исключением установленных на предоставленном компьютере, а также средств связи запрещено.
2. Заключительный тур проводится в тестирующей системе Ejudge, развернутой на сервере факультета компьютерных наук – http://www.cs.vsu.ru/ejudge/cgi-bin/new-register?action=209&contest_id=15&locale_id=1 Несмотря на то, что в данный момент на сервере настроен упрощенный режим регистрации без подтверждения по электронной почте (режим регистрации может быть изменен), настоятельно рекомендуется при регистрации указывать адрес реально существующего почтового ящика участника (завести почтовый ящик в интернете совсем несложно). И обязательно запомните ваш пароль (а лучше запишите)!
3. В рамках заключительного тура участникам за ограниченное время (4 часа) предлагается решить 6 задач разного уровня сложности. Решением каждой задачи является программа на одном из языков программирования, поддерживаемых тестирующей системой (C++, Pascal, C#, Java, Python), которую участник должен отправить на тестирование (посылка).
4. **В заключительном туре засчитываются только абсолютно полные решения задач, т.е. такие решения, которые проходят все тесты** (если задача не пройдет хотя бы один тест, то решение задачи не засчитывается). Ранжирование участников осуществляется по количеству решенных задач, если же кол-во решенных задач одинаковое, то учитывается суммарное время решения задач (для решенных задач – сумма времени, прошедшего от начала олимпиады до правильного решения каждой задачи) – чем оно меньше, тем лучше. **За каждую неверную посылку к суммарному времени добавляется штраф в 10 минут (так называемые АСМ-правила проведения турнира).**
5. Кол-во тестов для каждой задачи – 50 (для отдельных задач может отличаться), тесты составлены так, чтобы учитывались всевозможные, в том числе граничные, случаи в решениях (как правило, часть тестов к каждой задаче составлена случайным образом).
6. Общее количество всех посылок по всем задачам ограничено числом 100.
7. Для организации ввода-вывода в программе можно использовать как стандартный ввод-вывод, так и работу с файлами `input.txt / output.txt`. Решение всех задач должно соответствовать ограничению по времени в 1 с и по памяти – в 64 Mb (для отдельных задач могут быть указаны индивидуальные ограничения по времени и памяти).
8. Участникам олимпиады во время проведения олимпиады доступны полные протоколы тестирования отправленных решений. В протоколе тестирования отражены входные данные нескольких первых тестов для каждой задачи (возможно, что и всех тестов, зависит от размера тестов), но правильный ответ не показан (только вердикт проверки решения также без указания правильного ответа). Этой информацией можно пользоваться для отладки своих решений.
9. Решения, в которых вместо честного решения будет подбираться / забиваться ответ под известные тесты, если такие будут обнаружены, не учитываются. Жюри также оставляет за собой право перетестировать все решения всех участников для любой задачи заключительного тура на другом наборе тестов и при подведении итогов учитывать новые результаты.

10. Жюри олимпиады оставляет за собой право проводить проверку работ на списывание, после которой вносить изменения в результаты участников олимпиады, полученные при автоматическом тестировании решений.
11. В процессе проведения олимпиады участники могут обращаться с вопросами к жюри посредством отправки сообщений в тестирующей системе. Жюри может не отвечать на вопросы участников, если сочтет, что вопрос задан некорректно или же ответ будет содержать подсказку участнику.
12. В условиях задач, приведенных ниже, присутствует Задача 0. Длина окружности. Решение данной задачи не учитывается в итоговом результате заключительного тура, данная задача приведена исключительно для ознакомления с тестирующей системой и правилами оформления решений участниками олимпиады.

Желаем вам успеха!

Задача 0. Длина окружности

Входной файл	стандартный ввод / input.txt
Выходной файл	стандартный вывод / output.txt
Ограничение времени (сек/тест)	1

Примечание

Данная задача является учебной для ознакомления с правилами оформления задач и системой тестирования. Решение данной задачи не учитывается.

Условие задачи

Вам необходимо посчитать длину окружности радиуса R .

Входные данные

Во входного файла записано единственное целое число R – радиус окружности, длину которой вам необходимо посчитать.

Выходные данные

В выходной файл требуется вывести одно вещественное число L – длину окружности радиуса R .

Абсолютная погрешность ответа не должна превышать 10^{-6} .

Пример входного файла (stdin / input.txt)	Пример выходного файла (stdout / output.txt)
1	6.283185

Пример решения

C++ (стандартный ввод/вывод)	Pascal (стандартный ввод/вывод)
<pre>#define _USE_MATH_DEFINES #include <fstream> #include <ios> #include <iomanip> #include <cmath> using namespace std; int main() { int r; cin >> r; double l = 2 * M_PI * r; cout << fixed << setprecision(6); cout << l << endl; // другой способ: // char buf[50]; // sprintf(buf, "%.6f", l); // cout << buf << endl; }</pre>	<pre>const PI = 3.14159265359; var R: Integer; L: Double; begin ReadLn(R); L := 2 * R * PI; WriteLn(L:20:6); // никаких ReadLn в конце // быть не должно end.</pre>

Задача 1. Друзья и учеба

Входной файл	стандартный ввод / input.txt
Выходной файл	стандартный вывод / output.txt
Ограничение времени (сек/тест)	1
Ограничение памяти (Мб)	64

Условие задачи

На прошлой неделе, когда весь класс писал тест по математике, Вовочка болел. Директор школы требует, чтобы данный тест был написан всеми учениками, поэтому завтра этот тест предстоит решать Вовочке. Для Вовочки результат данного теста ни на что не влияет, т.к. он в этом году занял призовое место на областной олимпиаде по математике и директор лично перед всем классом пообещал поставить Вовочке по математике пять в году.

Тест оценивается в баллах от 0 до некоторого максимально возможного количества баллов M_{max} с шагом в один балл. Вовочка отлично разбирается в математике и поэтому уверен, что может набрать в данном тесте максимально возможное количество баллов (соответственно, и любое другое количество баллов по своему желанию). Однако Вовочка опасается, что если его результат будет сильно отличаться от результатов других учеников, то последние перестанут с ним дружить. Если говорить более строго, Вовочка считает, что с ним перестанут дружить те ученики в классе, результат которых будет отличаться от результата Вовочки более, чем на K баллов в любую сторону.

Помогите Вовочке определить, сможет ли он написать тест так, чтобы сохранить всех своих друзей (в данный момент все ученики в классе – друзья Вовочки).

Входные данные

В первой строке через пробел записаны три целые числа N , M_{max} и K ($1 \leq N$, $M_{max} \leq 100$, $0 \leq K \leq 100$) – количество остальных учеников в классе, исключая Вовочку, максимальный балл, который можно набрать по тесту, правильно ответив на все вопросы и максимально допустимая разница в баллах с кем-то из одноклассников, чтобы последний остался другом Вовочки.

В следующей строке через пробел записаны N целых чисел M_i ($0 \leq M_i \leq M_{max}$) – баллы по тесту, набранные одноклассниками Вовочки.

Выходные данные

Выведите строку "yes" (без кавычек), если Вовочка сможет написать тест, сохранив всех своих друзей, "no" (также без кавычек) – в противном случае.

Пример входного файла (input.txt)	Пример выходного файла (output.txt)	Комментарий
7 10 3 3 5 8 5 8 7 4	yes	Чтобы сохранить всех своих друзей, Вовочка может написать тест на 5 или на 6 баллов.
3 20 5 15 3 10	no	

Задача 2. Система уравнений

Входной файл	стандартный ввод / input.txt
Выходной файл	стандартный вывод / output.txt
Ограничение времени (сек/тест)	1
Ограничение памяти (Мб)	64

Условие задачи

Учитель математики составляет индивидуальные задания на контрольную для учеников седьмого класса. В качестве одного из заданий должна быть система, состоящая из двух уравнений, вида:

$$ax + by = c$$

$$dx + ey = f$$

Вместо букв a , b , c , d , e , f учитель хочет подставить числа так, чтобы единственным решением системы были целые числа x и y ($-99999 \leq x, y \leq 99999$). Чтобы ученикам не было слишком легко, требуется, чтобы все шесть коэффициентов были ненулевыми. Так как учитель не желает пугать учеников длинными числами, дополнительно он хочет, чтобы коэффициенты состояли не более, чем из пяти цифр ($-99999 \leq a, b, c, d, e, f \leq 99999$). Учитель просит Вовочку разработать компьютерную программу для составления систем уравнений согласно заданным требованиям.

Входные данные

Единственная строка содержит два целых числа, разделенных пробелом – x и y , единственное решение системы уравнений.

Выходные данные

Если невозможно составить систему уравнений, удовлетворяющую заданным требованиям, выведите единственное слово "no" (без кавычек).

Если можно составить систему уравнений, удовлетворяющую заданным требованиям, в первой строке выведите слово "yes" (без кавычек). Во второй строке через пробел выведите три целых числа – коэффициенты первого уравнения a , b и c , а в третьей строке аналогичным образом – коэффициенты второго уравнения d , e и f .

Пример входного файла (input.txt)	Пример выходного файла (output.txt)
1 5	yes 10 -1 5 20 30 170
0 99999	yes -3 1 99999 5 -1 -99999
0 0	no

Задача 3. Матрешки

Входной файл	стандартный ввод / input.txt
Выходной файл	стандартный вывод / output.txt
Ограничение времени (сек/тест)	1
Ограничение памяти (Мб)	64

Условие задачи

Вовочка не только хорошо учится, но и участвует в школьной художественной самодеятельности. Недавно, на фестивале русской народной культуры, в качестве сувенира Вовочке вручили набор из четырех матрешек ручной работы. Рассматривая сувенир, Вовочка придумал оригинальную задачу для участников школьного кружка по информатике. Вовочка считает, что знает решение данной задачи, но хочет проверить себя. Для этого необходимо, чтобы кто-то еще решил данную задачу для сравнения ответов, и желательно, чтобы это был не один человек.

Итак условие задачи: четыре матрешки разного размера стоят в один ряд и необходимо как можно быстрее собрать их в одну (вложить друг в друга). Скорость сборки оценивается в количестве *простых* действий. Возможны все два простых действия:

- поменять местами соседние матрешки;
- меньшую соседнюю матрешку вложить в большую.

Если в какой-то матрешке A уже вложена меньшая матрешка B , то ни в матрешку A (между A и B), ни в матрешку B вложить некую третью матрешку C уже нельзя (это не будет простым действием, т.к. в какой-то момент такой операции надо будет одновременно работать с тремя матрешками, не вложенными друг в друга). Но при этом матрешку A (с вложенными в нее матрешками), можно вкладывать в матрешку C (естественно, если матрешка C больше матрешки A).

Входные данные

В единственной строке через пробел записаны целые числа A, B, C, D ($1 \leq A, B, C, D, \leq 99$) – размеры матрешек в порядке их расположения в ряду слева направо.

Выходные данные

Выведите единственное число N_{min} – минимальное количество простых действий, которые приведут к сбору данного набора матрешек с учетом их первоначального взаимного расположения.

Пример входного файла (input.txt)	Пример выходного файла (output.txt)	Комментарий
3 7 2 4	4	Возможная последовательность действий (номер действия: взаимное расположение оставшихся матрешек после данного действия): 1: 7 3 2 4 2: 7 3 4 3: 7 4 4: 7
7 3 5 9	3	1: 7 5 9 2: 7 9 3: 9

Задача 4. Проектирование моста

Входной файл	стандартный ввод / input.txt
Выходной файл	стандартный вывод / output.txt
Ограничение времени (сек/тест)	1
Ограничение памяти (Мб)	64

Условие задачи

Для закрепления знаний по теме «Оптимизация» на дополнительных занятиях по информатике учитель раздал ученикам индивидуальные практические задания. Вовочке досталась задача о проектировании моста.

На прямой линии между берегами, вдоль которой будет проходить мост, существует N ($3 \leq N \leq 1000$) равномерно распределенных точек (т.е. точки расположены на одинаковом расстоянии D друг от друга). В каждой из этих точек был проведен замер глубины реки. Мост строится на некотором количестве опор, каждая из опор должна находиться в одной из точек с измеренной глубиной. Для повышения надежности моста максимально допустимое расстояние между двумя соседними опорами – $3D$ (другими словами, между соседними опорами может быть максимум две пропущенных точки). Также опоры обязательно должны быть размещены рядом с берегами реки – в первой и последней точке.

Известно, что глубина реки вдоль линии моста сначала с каждой точкой увеличивается, а затем уменьшается. Требуется спроектировать мост таким образом, чтобы суммарная длина L опор моста (другими словами, суммарное расстояние от дна до поверхности реки в точках с опорами) была минимально возможной. Помогите Вовочке определить величину L .

Входные данные

Первая строка содержит целое число N ($3 \leq N \leq 1000$). Во второй строке через пробел записано N целых чисел A_i ($1 \leq A_i \leq 1000$) – результат измерения глубины в точке i . Значение A_i в строке сначала увеличивается, затем уменьшается.

Выходные данные

Выведите одно число L – суммарное расстояние от дна до поверхности реки в точках с опорами при оптимальном размещении опор.

Пример входного файла (input.txt)	Пример выходного файла (output.txt)
6 1 3 4 7 3 2	7

Задача 5. Инвестиции

Входной файл	стандартный ввод / input.txt
Выходной файл	стандартный вывод / output.txt
Ограничение времени (сек/тест)	1
Ограничение памяти (Мб)	64

Условие задачи

Вовочка очень любит конфеты кондитерской фабрики, которая находится в паре кварталов от его дома. Недавно Вовочка узнал, что он может стать ее совладельцем, скупая акции кондитерской фабрики на фондовом рынке. Счастью Вовочки не было предела!

Вовочка твердо решил, что каждый день он будет экономить M рублей и покупать на них максимально возможное количество акций компании. Можно купить или продать только целое число акций. Вовочка будет действовать по простому алгоритму: каждый день он будет покупать максимальное число акций на все доступные ему в этот день деньги, сумма которых будет составлять сумму, оставшуюся с прошлого дня, и M рублей, сэкономленных в данный день. При этом Вовочка никогда не будет продавать купленные акции и на что-либо еще тратить деньги, оставшиеся в какой-либо день после покупки акций.

Прошло N дней и Вовочка хочет подсчитать два числа: сколько акций он купил и сколько максимально акций он мог бы купить, если бы действовал оптимально. Под оптимальностью подразумевается, что Вовочка бы совершал только такие покупки и продажи акций, которые в итоге привели бы его к владению максимальным количеством акций по прошествии N дней.

Для каждого из N дней известен курс акций на фондовом рынке – P_i рублей за одну акцию. Курс покупки не отличается от курса продажи.

Входные данные

В первой строке через пробел записаны два целых числа: N ($1 \leq N \leq 10^5$) – количество прошедших дней и M ($1 \leq M \leq 10^6$) – сумма рублей, которые Вовочка сэкономил каждый день.

Во второй строке записаны N целых чисел P_i ($1 \leq P_i \leq 10^6$) – стоимость покупки (продажи) акций в каждый из N дней.

Выходные данные

В единственной строке через пробел выведите два числа X и X_{max} – фактическое количество купленных акций и максимальное количество акций, которыми Вовочка мог бы владеть по прошествии N дней при оптимальной стратегии.

Гарантируется, что при любой стратегии ни в какой из N дней у Вовочки не могла оказаться на руках сумма, большая $2 * 10^9$ рублей.

Пример входного файла (input.txt)	Пример выходного файла (output.txt)	Комментарий
3 100 10 100 4	36 300	Оптимально в первый день на сэкономленные деньги купить 10 акций, продать их во второй день за 1000 рублей и на третий день за 1200 рублей (с учетом продажи и сэкономленных денег во второй и третий день) купить 300 акций
5 100 5 1 3 4 2	228 566	

Задача 6. Треугольник Серпинского

Входной файл	стандартный ввод / input.txt
Выходной файл	стандартный вывод / output.txt
Ограничение времени (сек/тест)	1
Ограничение памяти (Мб)	64

Условие задачи

На прошлой неделе на дополнительном занятии по математике Вовочка узнал про фракталы. Если говорить простыми словами (насколько это возможно в одном предложении), фракталы – это объекты, структура которых повторяется в составных частях объекта и так до бесконечности. Некоторые объекты в природе, а также математические объекты, обладают свойствами фракталов.

В качестве наглядных примеров учитель показал классу несколько примеров фракталов, построенных на плоскости. Среди этих примеров был треугольник Серпинского, на рисунке ниже показан принцип построения данной фрактала.



Вовочке идея фракталов очень понравилась, так что он в качестве тренировки решил написать программы для построения нескольких фракталов – первым он хочет нарисовать как раз треугольник Серпинского. Однако, вот незадача, компьютерную графику на уроках информатики еще не проходили, поэтому Вовочка решил «рисовать» некий аналог рассматриваемого фрактала с помощью псевдографики, т.е. текстовых символов, имитирующих графические примитивы.

Входные данные

Единственная строка содержит одно целое число N ($1 \leq N \leq 10$) – размерность «рисуемой» фигуры.

Выходные данные

С помощью символов псевдографики и пробелов должна быть построена фигура, напоминающая треугольник Серпинского. Допустимы только следующие символы:

- косая черта (слеш) – '/', ASCII-код символа 47 (0x2F);
- обратная косая черта (бекслеш) – '\', ASCII-код символа 92 (0x5C);
- подчеркивание – '_', ASCII-код символа 95 (0x5F);
- символы перевода строк.

Правила построения фигур, а также понятие их размерности, лучше всего иллюстрируются приведенными ниже примерами входных и выходных данных для $N = 1, 2, 3$ и 4 .

Построенный треугольник должен касаться верхней и левой границы «рисунка». Каждая строка должна заканчиваться обратной косой чертой, без пробелов после, т.е. все строки будут разной длины. После последнего «графического» символа (обратной косой черты), должен быть перевод строки. Другими словами, правильный ответ в данной задаче должен полностью посимвольно совпадать с эталонным решением жюри.

Пример входного файла (input.txt)	Пример выходного файла (output.txt)	Комментарий
1	<pre> /\ / \ _/ _ </pre>	<pre> / \ / \ _/__ </pre> <p>Для наглядности в виде сетки обозначены границы символов, т.е. каждая клеточка – один символ (пустые клетки – пробелы, символы перевода строк не показаны)</p>
2	<pre> /\ / \ _/ _ /\ /\ _/_ _/_ </pre>	<pre> / \ / \ _/__ / \ / \ _/__ _/__ </pre>
3	<pre> /\ / \ _/ _ /\ /\ _/_ _/_ /_/_ /_/_ _/___/___/___ </pre>	<pre> / \ / \ _/__ / \ / \ _/__ _/__ / \ / \ _/__ _/__ / \ / \ / \ _/__ _/__ _/__ / \ / \ / \ / \ _/__ _/__ _/__ _/__ </pre>
4	<pre> /\ / \ _/ _ /\ /\ _/_ _/_ /_/_ /_/_ _/___/___/___ _/___/___/___ _/___/___/___ _/___/___/___ </pre>	