

Общая информация по задачам олимпиады

Правила олимпиады

Олимпиада длится три часа. За это время вам будет предложено решить 7 задач по программированию. Решение задачи — это исходный код программы, которая без ошибок выполняет поставленную задачу.

Каждое решение будет проверено автоматически на некотором заранее заготовленном наборе тестов. Если ваша программа проходит все тесты, она считается решённой верно. Если ваша программа не проходит хотя бы один тест, она считается не решённой.

Штраф за задачу — это время сдачи задачи от начала соревнования в минутах плюс 20 штрафных минут за каждую неуспешную попытку сдачи. В первую очередь, старайтесь решить как можно больше задач, а во вторую сдавайте быстро и без ошибок, чтоб сделать штрафное время как можно меньше.

Задачи упорядочены по сложности, однако сложность — субъективное мнение жюри. Поэтому **обязательно прочтите все задачи!**

Ввод и вывод

Помните, что на олимпиаде нет ручной проверки ваших программ. Поэтому не выводите приветственные фразы (например, «Введите число n :»).

Соблюдайте формат ввода, описанный в соответствующем разделе условий задач. Данные будут введены с теми ограничениями, которые описаны в условиях. Не надо проверять «правильность» входных данных — об этом уже позаботилось жюри.

Выводить нужно именно то, что просят. Если требуется вывести целое число, вы получите ошибку при попытке вывести число с точкой или запятой. Если требуется вывести одну строку, вы получите ошибку при попытке вывести две строки. И так далее...

Примеры

Гарантируется, что примеры содержат правильный ответ на задачу. Так, вы можете у себя на компьютере протестировать программу ещё до того, как отправите её на проверку.

Также гарантируется, что тесты из условия — это первые тесты в задаче. Так, если вы получаете ошибку в системе на первом тесте (например, `wrong answer 1`), то вы получаете ошибку именно на первом тесте из условия. В этом случае, попробуйте запустить программу у себя на компьютере.

Вы можете убедиться, что ваше решение проходит тесты из условий в системе: для этого отправьте его на *тестирование*.

Ограничения

Ограничение по памяти во всех задачах — 64 Мегабайта.

Ограничение по времени указано в условиях каждой задачи отдельно.

Ограничения по времени и памяти распространяются на один конкретный тест. Обратите внимание, что время выполнения программы в системе может отличаться от времени выполнения на вашем компьютере (обычно система быстрее).

Вы не ограничены по количеству попыток, но вы можете отправлять решения только один раз в десять секунд.

Прочее

Не забывайте, что вы можете задать вопрос по задаче в системе.

Дополнительную информацию вы можете получить в «справке» в системе. Обратите внимание, что там находится **информация о вердиктах**: если вы получили вердикт по задаче, смысл которого вы не понимаете (или хотите уточнить), прочтите соответствующий пункт в справке.

Задача А. Путешествие в прошлое

Ограничение по времени: 0.5 секунд

Даниил живёт в Удивительном мире.

Долгое время в Удивительном мире использовался Юлианский календарь — это такой календарь, где год является високосным, только если номер года делится на 4. Так, например, 700-й год — високосный, потому что 700 делится на 4, а 1001-й — не високосный, потому что 1001 на 4 не делится.

Но в 1337 году ввели новый календарь, в котором високосные года исчезли (т.е. после 1337-го года уже никакой год не является високосным).

Сейчас в Удивительном мире уже 100 000-й год. Известно, что реформа календаря 1337-го года — это единственная календарная реформа в Удивительном мире за всё время.

Даниил собирается отправиться в прошлое из 100 000-го года в год y . Однако, чтобы верно задать точку прибытия, нужно знать, високосный ли год y . Помогите ему ответить на вопрос!

Формат входных данных

В единственной строке вводится целое число y — год, который надо проверить ($1 \leq y \leq 100\,000$).

Формат выходных данных

В единственной строке выведите «YES» (без кавычек), если год y — високосный, и «NO» (без кавычек), иначе.

Примеры

входные данные	ответ
2020	NO
700	YES
1337	NO

Замечание

В первом примере Даниил хочет узнать, является 2020-й год високосным. В удивительном мире уже прошла реформа 1337-го года, после которой високосов не стало. Таким образом, 2020 год не високосный.

Во втором примере спрашивается про 700-й год. Так как реформа 1337-го года ещё не прошла, а 700 делится на 4 ($700 = 175 \cdot 4$), то 700-й год — високосный.

Обратите внимание на третий пример: 1337-й год високосным не считается!

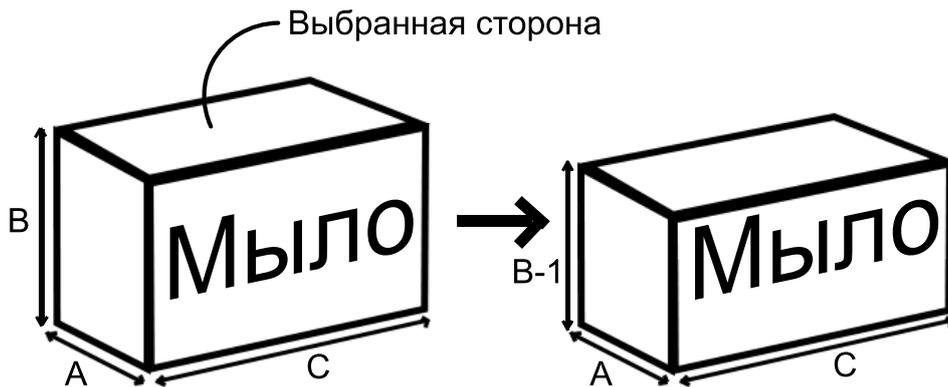
Задача В. Мыло

Ограничение по времени: 0.5 секунд

Мыло — это прямоугольный параллелепипед $A \times B \times C$ (в мм).

Витя любит мыться. За один приём душа он выбирает одну из шести сторон, которой он будет мыться. За приём душа Витя всегда расходует ровно 1 мм толщины мыла (т.е. размеры мыла в перпендикулярном выбранной стороне направлении уменьшатся на 1 мм), а сам Витя получает удовольствие от душа, равное площади (в квадратных миллиметрах) стороны, которой он мылся.

К примеру, выберем сторону как на рисунке. Тогда B уменьшится на один миллиметр, а полученное удовольствие составит $A \times C$. Аналогично с остальными сторонами.



Обратите внимание, что мыло закончится, если размер по какому-то направлению станет равным нулю. Если же оно не закончится, то в следующий раз Витя будет мыться уже уменьшенным куском мыла.

Витя может каждый день выбирать любую сторону. Какое наибольшее суммарное удовольствие от мытья он может получить за то время, пока мыло не кончится?

Формат входных данных

В единственной строке вводятся три целых числа A , B и C ($1 \leq A, B, C \leq 1000$).

Формат выходных данных

Выведите единственное целое число — наибольшее суммарное удовольствие от мытья.

Пример

входные данные	ответ
2 2 3	12

Задача С. Буквы на холодильнике

Ограничение по времени: 0.5 секунд

Наверное, вы знаете о магнитах в виде букв, которые можно прикрепить на холодильник.

У Вани есть 4 магнитика в виде заглавных латинских букв. Сейчас на холодильнике они висят друг за другом, образуя слово *s*. Магнитики можно свободно переставлять между собой.

Ваня хочет собрать из магнитиков его любимое слово «OZSH», однако, возможно, каких-то букв ему не хватит! Но он знает, что Витя может поменять любой магнит на другой (тоже в виде заглавной латинской буквы) за один доллар.

Долларов у Вани немного, поэтому ему хочется потратить их как можно меньше. Какое наименьшее количество долларов ему нужно потратить на обмены, чтобы в результате у него оказались буквы, переставив которые можно получить слово «OZSH»?

Формат входных данных

В единственной строке вводится слово *s* длины 4, состоящее только из заглавных букв латинского алфавита.

Формат выходных данных

Выведите единственное целое число — наименьшее количество долларов, которое нужно потратить Ване, чтоб получить слово «OZSH».

Пример

входные данные	ответ
ZOGH	1

Замечание

В первом примере сначала у Вани есть магниты Z, O, G и H. Давайте поменяем у Вити букву G на букву S. Теперь в наличии есть буквы Z, O, S и H, из которых можно сложить слово OZSH.

Задача D. Крутые числа

Ограничение по времени: 0.5 секунд

У Елены Вениаминовны есть цифра a . Чтоб цифре a не было одиноко, она хочет приписать к ней справа одну цифру, чтобы полученное двузначное число k было *крутым*.

Двузначное число k назовём *крутым*, если существует такое натуральное число d , что выполнены все следующие условия:

- Число k делится на d ;
- Сумма цифр числа k делится на d ;
- $d > 1$.

Например, число 24 — крутое, потому что при $d = 2$ все три условия выполнены:

- 24 делится на 2;
- Сумма цифр числа 24, равная 6, делится на 2;
- $2 > 1$.

Вам дана цифра a . Выведите в порядке возрастания все (двузначные) крутые числа, начинающиеся с a .

Формат входных данных

В единственной строке вводится целое число a — первая цифра числа ($1 \leq a \leq 9$).

Формат выходных данных

Выведите через пробел одно или несколько целых чисел — все крутые числа, начинающиеся с цифры a , в порядке возрастания.

Пример

входные данные	ответ
2	20 21 22 24 26 27 28

Замечание

В примере $a = 2$, поэтому нужно вывести все крутые числа, начинающиеся с двойки, в порядке возрастания.

Задача Е. Удачная позиция

Ограничение по времени: 0.5 секунд

В Удивительном мире есть n программистов. Даниил решил собрать их вместе и построить в шеренгу.

Даниил смотрит на *крутость* программистов: каждый программист, стоящий в шеренге на позиции i , характеризуется крутостью — целым числом a_i . Так, в шеренге стоят программисты с крутостями a_1, a_2, \dots, a_n .

Не крайняя позиция i в шеренге называется *хорошей*, если на этой позиции стоит программист, чья крутость больше, чем крутость его соседей. Формально, позиция i — хорошая, если $1 < i < n$ и $a_{i-1} < a_i > a_{i+1}$.

Даниил решил поменять каких-то программистов местами. Для этого он написал n чисел: b_1, b_2, \dots, b_n . Затем, он попросил всех программистов поменять свою позицию: для любого i программист, стоявший на позиции i , перемещается на позицию b_i .

Причём Даниил сделал числа такими, чтобы он мог вернуться к исходной шеренге, переставив программистов (с теми же числами b_1, b_2, \dots, b_n) ещё раз. Т.е., формально, гарантируется, что $b_{b_i} = i$.

Позиция i называется *удачной*, если она хорошая и до перестановки, и после.

Посчитайте количество удачных позиций.

Формат входных данных

В первой строке вводится целое число n — количество программистов в Удивительном мире ($1 \leq n \leq 100$).

Во второй строке вводятся n целых чисел a_1, \dots, a_n — крутости программистов в исходной шеренге ($-10^9 \leq a_i \leq 10^9$).

В третьей строке вводятся n целых чисел b_1, \dots, b_n — числа, которые придумал Даниил. ($1 \leq b_i \leq n, b_{b_i} = i$).

Формат выходных данных

Выведите единственное целое число — количество удачных позиций в шеренге.

Пример

входные данные	ответ
5 1 3 2 5 4 1 4 5 2 3	1

Замечание

В первом примере крутости программистов в шеренге были: 1, 3, 2, 5, 4.

После перестановки первый программист остался на месте, второй перешёл на четвёртую позицию, третий — на пятую, четвёртый — на вторую и пятый — на третью. Так, крутости программистов в шеренге стали: 1, 5, 4, 3, 2.

Так, вторая позиция является хорошей и до перестановки (потому что крутость стоящего на этой позиции программиста больше, чем у соседей: $3 > 2$ и $3 > 1$), и после ($5 > 4$ и $5 > 1$). Заметим, что это единственная удачная позиция.

Задача F. Азас

Ограничение по времени: 1 секунда

Дана строка S , состоящая из заглавных букв латинского алфавита.

Тролльная строка — это строка, получаемая сложением (конкатенацией) какого-то количества строк «AZ». Например, строки «AZ», «AZAZAZ» — тролльные, а строки «STRING», «AZA», «ZAZA» — не тролльные.

Затролленностью строки S будем называть длину наибольшей тролльной строки, которая входит в S в качестве подстроки.

Строка S состоит из заглавных латинских букв. Вы можете заменить некоторые буквы строки S на другие заглавные латинские буквы.

Требуется получить строку с наибольшей возможной затролленностью. Если можно получить несколько различных строк с наибольшей затролленностью, то среди таких нужно вывести ту, что получается наименьшим количеством замен символов. Если и таких строк несколько, то можно вывести любую.

Формат входных данных

В первой строке вводится целое число N ($1 \leq N \leq 100\,000$) — длина строки S .

Во второй строке вводится строка S , состоящая из заглавных букв латинского алфавита.

Формат выходных данных

В единственной строке выведите единственную строку — ответ на задачу.

Пример

входные данные	ответ
7 ABACABA	AZAZAZA

Замечание

Затролленность строки «AZAZAZA» равна 6, потому что в нее входит тролльная строка «AZAZAZ». Можно показать, что нельзя сделать строку с большей затролленностью.

Чтоб получить строку «AZAZAZA» из строки «ABACABA», нужно три замены. Можно показать, что нельзя сделать строку с затролленностью 6 меньше, чем за три замены.

Задача G. Суперкомбинация

Ограничение по времени: 0.5 секунд

Дан массив $[a_1, a_2, \dots, a_n]$, состоящий только из 1 и 2.

Суперкомбинация — это следующая операция: берём и меняем данный массив на массив всевозможных попарных произведений чисел. В результате будет образован новый массив:

$$\begin{bmatrix} a_1 \cdot a_2, & a_1 \cdot a_3, & \dots, & a_1 \cdot a_n, \\ & a_2 \cdot a_3, & \dots, & a_2 \cdot a_n, \\ & & \ddots & \vdots \\ & & & a_{n-1} \cdot a_n \end{bmatrix}$$

Сделаем всего k суперкомбинаций: сначала над исходным массивом, затем над результатом, и т. д. Сколько чисел в получившемся массиве (т.е. после k суперкомбинаций) делится на 2?

Формат входных данных

В первой строке заданы целые числа n и k ($1 \leq n \leq 20, 1 \leq k \leq 4$).

Во второй строке вводятся n целых чисел: a_1, a_2, \dots, a_n — исходный массив ($1 \leq a_i \leq 2$).

Формат выходных данных

Выведите единственное целое число: ответ на задачу.

Примеры

входные данные	ответ
3 2 1 2 2	3
2 4 1 1	0

Замечание

В первом примере массив меняется так: $[1, 2, 2] \xrightarrow{\text{суперкомб.}} [1 \times 2, 1 \times 2, 2 \times 2] = [2, 2, 4] \xrightarrow{\text{суперкомб.}} [2 \times 2, 2 \times 4, 2 \times 4] = [4, 8, 8]$. Все числа делятся на 2.

Во втором примере массив меняется так: $[1, 1] \xrightarrow{\text{суперкомб.}} [1 \times 1] = [1] \xrightarrow{\text{суперкомб.}} [] \xrightarrow{\text{суперкомб.}} []$. Массив пуст!