

Ты решал(а) на школьном этапе и вчера задачи по комплекту 9-11. Верно? Если нет – этот комплект заданий не для тебя – обратись сейчас к организатору.

Если все верно, то вот твоя ссылка на проверяющую систему:

contest.yandex.ru/contest/71905

Задача 1. Ваня Гудвин и Good energy V.2 (100 баллов)

Имя входного файла: `energy2.in` или стандартный поток ввода

Имя выходного файла: `energy2.out` или стандартный поток вывода

Ограничение времени 1 секунда на тест

Ограничение по памяти 128 Мб

Вчера на пробном туре мы предложили решить задачу про Ваню Гудвина. Внимательные школьники обратили внимание на детские ограничения на значение числа n . Сегодня все будет по-взрослому. Банок Good energy будет много. Итак, начнем олимпиаду!

У Вани Гудвина есть n банок Good energy. В день он продает по x банок Good energy, а каждый второй день он может "войти в кондиции" и продать в 2 раза больше банок. За какое минимальное количество дней он сможет продать все свои банки Good energy?

Формат входных данных:

На вход программе в первой строке подаются число n , а во второй строке – число x . Оба числа натуральные, не превосходящие 9000000000.

Формат выходных данных:

В качестве результата Ваша программа должна вывести минимальное количество дней, за которое Ваня Гудвин сможет продать все банки Good energy.

Пример файла с входными данными и файла с результатом:

<code>energy.in</code>	<code>energy.out</code>
210	14
10	

Задача 2. Двоичная система счисления (100 баллов)

Имя входного файла: *binary.in* или стандартный поток ввода
Имя выходного файла: *binary.out* или стандартный поток вывода
Ограничение времени: *1 секунда на тест*
Ограничение по памяти: *256 Мб*

Ученики восьмого класса изучают в школе тему "Двоичная система счисления". Айнуру очень нравится эта тема. Он пытается полностью осознать фразу учительницы информатики о том, что любое натуральное число можно записать в двоичной системе счисления, т.е. представить в виде суммы некоторых степеней двойки. Еще Айнур хочет научиться вычитать двоичные числа столбиком. Пока ему не очень это удается – на уроках, выполняя вычитание двоичных чисел, он вынужден сначала переводить их в десятичную систему, вычитать в ней, а затем переводить обратно в двоичную. Учительница информатики посоветовала Айнуру такое упражнение: сначала Айнур записывает в двоичной системе счисления число A , являющееся суммой некоторых степеней двоек. Затем он вычитает число A из числа $2^n - 1$. Например, число A должно быть сформировано из 1-ой, 2-ой и 4-ой степеней двоек, $n=5$. Тогда число A будет такое $A = 2^1 + 2^2 + 2^4 = 22_{10} = 10110_2$. Вычитать его требуется из $2^5 - 1$. Айнур, правда, пока все еще пользуется переводом в десятичную систему: $2^5 - 1 = 31_{10}$. $31_{10} - 22_{10} = 9_{10} = 1001_2$. Итак, ответ Айнура 1001_2 . Помогите Айнуру изучить материал: напишите программу, на вход которой подаются степени двоек, подлежащих сложению для формирования числа A и число n . Ваша программа должна выводить в порядке возрастания через пробел степени двоек, из которых формируется результат вычитания.

Формат входных данных:

На вход программе в первой строке подаются два числа t и n . Где t – количество слагаемых для формирования исходного числа. ($1 \leq t, n \leq 200500$). Гарантируется, что $2^n - 1 > A$. В следующей строке записано t чисел через запятую – показатели степеней двоек в порядке возрастания.

Формат выходных данных:

В качестве результата Ваша программа должна вывести в одну строку через пробел в порядке возрастания показатели степени двоек, из которых состоит результат вычитания $(2^n - 1) - A$.

Пример входных данных и верного результата:

<code>binary.in</code>	<code>binary.out</code>
3 5 1, 2, 4	0 3

Пояснения к примеру:

3 – это количество чисел во второй строке.

$$A = 2^1 + 2^2 + 2^4$$

$$(2^5 - 1) - A = 9_{10} = 1001_2 = 2^0 + 2^3.$$

Задача 3. Подарок в кубе в кубе в кубе (100 баллов)

Имя входного файла: *boxes.in*
Имя выходного файла: *boxes.out*
Ограничение времени: *1 секунда на тест*
Ограничение по памяти: *256 Мб*

Леднезу очень понравилась идея подарка, последовательно упакованного в комически большое количество слоёв коробок. Очень скоро у его друга Кира день рождения, на котором имениннику предстоит процесс распаковки своего подарка. После долгих раздумий Леднез решил упаковать подарок ровно в три коробки (две коробки не очень интересно, а четыре уже трудоёмко).

Всего у Леднеза заготовлено n кубических коробок. i -ая по счёту коробка имеет длину стороны a_i . Одну коробку можно уместить в другую, если длина стороны большей коробки строго больше длины стороны меньшей. Сам подарок достаточно маленький и может поместиться в любой коробке из имеющихся.

Леднез не знает, как лучше вложить коробки, и хочет рассмотреть все варианты. Вы, как эксперт в области расчётов, хотите отговорить его от перебора всего множества случаев, но для полноты своей аргументации придётся точно посчитать, сколько вариантов получится. Помогите Леднезу, а то он точно не успеет до дня рождения Кира.

Формат входных данных:

В первой строке входных данных задаётся число t ($1 \leq t \leq 100$) — количество наборов входных данных. Далее подаются сами наборы входных данных, состоящие из двух строк. В первой строке каждого набора задаётся единственное число n ($3 \leq n \leq 10^5$). В следующей строке перечислены числа a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$), обозначающие размеры коробок.

Гарантируется, что в каждом тесте коробок не менее трёх. Гарантируется, что сумма n по всем наборам входных данных не превышает 10^5 .

Формат выходных данных:

Для каждого набора входных данных в том же порядке выведите количество вариантов последовательно упаковать подарок в три коробки. Каждое число следует вывести в отдельной строке.

Система оценки:

группа	баллы	ограничения	необходимые группы
1	40	$n \leq 20$	-
2	20	$t \leq 10, n \leq 10^3$	1
3	20	$t \leq 10, a_i \leq 10^3$	1
4	20	без ограничений	1, 2, 3

Примеры входных данных и верных результатов:

<code>boxes.in</code>	<code>boxes.out</code>
1 5 1 2 2 3 3	4
2 8 7 2 4 1 9 9 1 8 7 5 5 8 8 9 9 3	44 20

Пояснение к примеру:

В первом примере возможны четыре варианта упаковки подарка. Подходят комбинации коробок с номерами (1, 2, 4), (1, 2, 5), (1, 3, 4), (1, 3, 5).

Задача 4. Парк развлечений (100 баллов)

Имя входного файла: *park.in* или стандартный поток ввода
 Имя выходного файла: *park.out* или стандартный поток вывода
 Ограничение времени: 2 секунды на тест
 Ограничение по памяти: 256 Мб

В новом современном парке развлечений все аттракционы разместили равномерно по кругу и пронумеровали от 0 до $n-1$, где n общее количество аттракционов в парке. Некоторые из аттракционов решили соединить между собой наикратчайшими прямолинейными участками детской железной дороги. Создали управление планирования детской железной дороги и назначили туда директором Ивана Ивановича. Архитекторы, дизайнеры и просто неравнодушные жители города стали присылать Ивану Ивановичу свои варианты проектов детской железной дороги. Жизнь Ивана Ивановича усложнилась: он даже не успевает проанализировать все эти проекты. Для начала Иван Иванович решил забраковать те проекты, в которых наикратчайшие прямолинейные участки железной дороги пересекаются между собой (касание в вершине допускается) реализация таких проектов категорически не впишется в бюджет.

Формат входных данных:

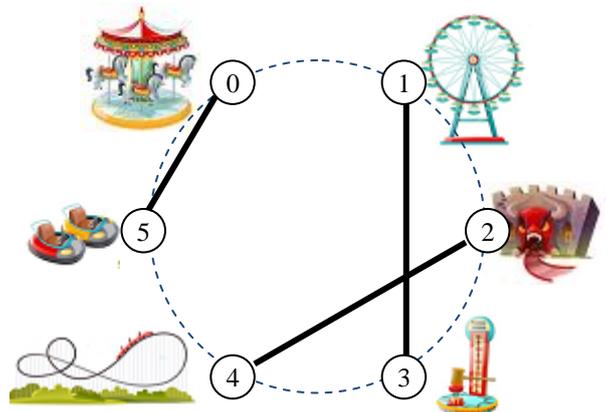
Каждый комплект входных данных является описанием одного проекта детской железной дороги. На вход программе в первой строке поступают два целых числа n и q – количество аттракционов в парке и количество прямолинейных участков железной дороги ($3 \leq n \leq 10^5$), ($1 \leq q \leq \min(10^5, n(n-1)/2)$). В следующих q строках записаны по два числа через пробел – номера аттракционов (нумерация с нуля), которые в данном проекте предлагается соединить наикратчайшим прямолинейным участком железной дороги. Никакой фрагмент железной дороги в проекте не повторяется более одного раза. Дорога не может соединять аттракцион сам с собой.

Формат выходных данных:

Выведите слово "discard", если проект следует забраковать, и слово "approve" в противном случае.

Примеры входных данных, верных результатов и пояснения к примеру №1:

park.in	park.out
6 3 5 0 2 4 3 1	discard
6 3 5 4 3 0 4 3	approve



Система оценки:

группа	баллы	ограничения	необходимые группы
1	40	$n \leq 50$	-
2	20	$n \leq 10^3$	1
3	20	$10^3 < n \leq 10^5$, от каждого аттракциона исходит не более одной дороги	-
4	20	без ограничения	1, 2, 3