

Задача A. Median

Имя входного файла: `median.in`
Имя выходного файла: `median.out`
Ограничение по времени: 7 seconds
Ограничение по памяти: 64 Mebibytes

Центральным элементом набора из k чисел называется такой элемент, который после сортировки набора будет занимать в нём центральную позицию (то есть позицию номер $\lceil k/2 \rceil$, считая с единицы).

Числа добавляются в изначально пустой набор в заданном порядке. Требуется определить значения центрального элемента после добавления каждого числа.

Формат входного файла

Входной файл содержит количество чисел n , за которым следуют n целых чисел a_i в порядке их добавления в набор ($1 \leq n \leq 10^6$, $-10^9 \leq a_i \leq 10^9$).

Формат выходного файла

В выходной файл выведите n целых чисел. i -е число — значение центрального элемента после i -го добавления.

Примеры

<code>median.in</code>	<code>median.out</code>
3 5 3 4	5 3 4
5 1 2 3 2 4	1 1 2 2 2

Задача В. Robots

Имя входного файла: `robots.in`
Имя выходного файла: `robots.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Сообщество роботов живет по следующим законам:

- один раз в начале года они объединяются в группы по три или пять роботов;
- за один год группа из 3 роботов собирает 5 новых, а группа из 5 роботов собирает 9 новых;
- роботы объединяются так, чтобы собрать за год наибольшее количество новых роботов;
- каждый робот живет три года после сборки.

Известно, что начальное количество роботов равно N и все они только что собраны. Напишите программу, определяющую, сколько роботов будет через K лет.

Формат входного файла

В первой строке входного файла записаны два натуральных числа N ($1 \leq N \leq 12$) и K ($1 \leq K \leq 12$).

Формат выходного файла

В первой и единственной строке выходного файла запишите количество роботов через K лет.

Пример

<code>robots.in</code>	<code>robots.out</code>
12 2	88
10 3	208

Задача C. Mythical Chess

Имя входного файла: `chess.in`
Имя выходного файла: `chess.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Компания «Jedi Games» занимается разработкой компьютерной игры «Мифические шахматы». Вам в рамках этого проекта требуется написать программу, определяющую минимальное количество ходов, необходимое кентавру, чтобы добраться из одной клетки в другую.

В мифические шахматы играют на шахматной доске размером 9×9 , угловые клетки которой окрашены в черный цвет. Кентавр — фигура мифических шахмат, объединяющая в себе свойства коня и слона. Когда кентавр стоит на белой клетке, он может ходить только как конь, а когда на черной — только как слон.

Формат входного файла

Во входном файле в первой строке записано натуральное число $N \leq 50$ — количество тестов. В следующих N строках для каждого теста записаны координаты (большая латинская буква и цифра) двух клеток доски для мифических шахмат, разделенных пробелом: сначала — клетки, на которой стоит фигура, а потом — клетки, до которой надо добраться.

Формат выходного файла

Для каждого теста в выходной файл выводится строка, содержащая одно целое число — минимальное количество ходов, необходимое кентавру, чтобы добраться из первой клетки во вторую. Если добраться невозможно, то выводится число -1 .

Пример

<code>chess.in</code>	<code>chess.out</code>
2	2
H6 E5	3
A6 F6	

Задача D. Double Bank

Имя входного файла: `double.in`
Имя выходного файла: `double.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Имеются N модулей памяти, способных работать только в паре. Некоторые из них исправны, некоторые - нет. Вставив два модуля в компьютер и включив его, можно получить один из двух результатов:

- Компьютер включается — оба модуля исправны;
- Компьютер не включается — какой-то модуль неисправен (или оба сразу). Какой именно модуль неисправен, неизвестно.

После проверки пары модулей принимается решение о том, какая пара модулей будет проверяться следующей. Требуется найти, какое минимальное число проверок в худшем случае потребуется выполнить, чтобы определить, какие именно модули исправны или чтобы убедиться, что точно определить набор исправных модулей нельзя.

Формат входного файла

В первой строке входного файла содержится одно число N ($1 \leq N \leq 100$) — количество модулей.

Формат выходного файла

Выведите в выходной файл минимальное количество включений компьютера, которого заведомо должно хватить для решения задачи.

Пример

<code>double.in</code>	<code>double.out</code>
2	1

Задача E. Earth

Имя входного файла: `earth.in`
Имя выходного файла: `earth.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Задано несколько точек на Земле (широтой и долготой). Требуется циклически облететь их все, преодолев минимально возможный путь. В данной задаче предполагается, что Земля — шар радиусом 6400 км.

Формат входного файла

В первой строке входного файла находится целое число N ($3 \leq n \leq 10$). Далее в N строках заданы целые координаты точек — широтой и долготой (широта по модулю меньше 90, долгота по модулю меньше 180).

Формат выходного файла

Вывести длину кратчайшего маршрута в километрах (с точностью до 10^{-3}).

Пример

<code>earth.in</code>	<code>earth.out</code>
3 -59 -138 -81 52 62 -37	37292.149

Задача F. Fool game

Имя входного файла: `fool.in`
Имя выходного файла: `fool.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Параллельно с «Мифическими шахматами» компания «Jedi Games» разрабатывает компьютерную реализацию популярной карточной игры «дурак».

Как известно, в «дурака» играют колодой из 36 карт. В данной реализации каждая карта представляется в виде строки из двух символов, где первый символ означает ранг (6, 7, 8, 9, T, J, Q, K, A) карты, а второй символ означает масть (S, C, D, H). Ранги перечислены в порядке возрастания старшинства.

Вам необходимо решить следующую задачу: сможет ли игрок, обладая набором из N карт, отбить M карт, которыми под него сделан ход? Для того чтобы отбиться, игроку нужно покрыть каждую из карт, которыми под него сделан ход, картой из своей колоды. Карту можно покрыть либо старшей картой той же масти, либо картой козырной масти. Если кроющаяся карта сама является козырной, то её можно покрыть только старшим козырем. Одной картой можно покрыть только одну карту.

Формат входного файла

Первая строка входного файла содержит целое положительное число $T \leq 50$ — число тестов. Каждый тест состоит из трех строк. На первой строке каждого теста находятся два целых числа N и M ($1 \leq N \leq 35$, $1 \leq M \leq 4$, $M \leq N$), а также символ R , означающий козырную масть. На второй строке теста перечислены N карт, находящихся на руках у игрока. На следующей строке теста перечислены M карт, которые необходимо отбить. Все карты, которые нужно отбить, будут иметь один ранг.

Формат выходного файла

Для каждого теста необходимо вывести «YES» в случае, если отбиться можно, либо «NO», если нельзя.

Пример

<code>fool.in</code>	<code>fool.out</code>
2	YES
6 2 C	NO
KD KC AD 7C AH 9C	
6D 6C	
4 1 D	
9S KC AH 7D	
8D	