

Задача А. Приближенный двоичный поиск

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 6 секунд
Ограничение по памяти: 64 мегабайта

Формат входных данных

В первой строке входных данных содержатся числа N и K ($1 \leq N, K \leq 100\,000$). Во второй строке задаются N чисел первого массива, отсортированного по возрастанию, а в третьей строке — K чисел второго массива. Каждое число в обоих массивах по модулю не превосходит 10^9 .

Формат выходных данных

Для каждого из K чисел выведите в отдельную строку число из первого массива, наиболее близкое к данному. Если таких несколько, выведите меньшее из них.

Пример

стандартный ввод	стандартный вывод
5 5	1
1 3 5 7 9	3
2 4 8 1 6	7
	1
	5

Задача В. Такси

Имя входного файла: `taxi.in`
Имя выходного файла: `taxi.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

После затянувшегося совещания директор фирмы решил заказать такси, чтобы развезти сотрудников по домам. Он заказал N машин — ровно столько, сколько у него сотрудников. Но когда они подъехали, оказалось, что у каждого водителя такси свой тариф за 1 километр.

Каждый сотрудник сказал директору, сколько километров ему нужно проехать до дома. Разные сотрудники должны сесть в разные такси. Теперь директор хочет определить, какой из сотрудников на каком такси должен поехать домой, чтобы суммарные затраты на такси (а их несет фирма) были минимальны.

Формат входных данных

Сначала во входном файле записано натуральное число N ($1 \leq N \leq 1000$) — количество сотрудников компании (совпадающее с количеством вызванных машин такси). Далее записано N чисел, задающих расстояния в километрах от работы до домов сотрудников компании (первое число — для первого сотрудника, второе — для второго и т.д.). Все расстояния — положительные целые числа, не превышающие 1000. Далее записано еще N чисел — тарифы за проезд одного километра в такси (первое число — в первой машине такси, второе — во второй и т.д.). Тарифы выражаются положительными целыми числами, не превышающими 10000.

Формат выходных данных

В выходной файл выведите N чисел — оптимальное распределение сотрудников по такси. Первым выведите номер такси, в которое должен сесть первый сотрудник, вторым — номер такси, в которое должен сесть второй и т.д. Если есть несколько вариантов рассадки сотрудников, при которых затраты минимальны, выведите любой из них.

Примеры

<code>taxi.in</code>	<code>taxi.out</code>
3 10 20 30 50 20 30	1 3 2
5 10 20 1 30 30 3 3 3 2 3	2 3 1 5 4

Задача С. Детский праздник

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Организаторы детского праздника планируют надуть для него M воздушных шариков. С этой целью они пригласили N добровольных помощников, i -й среди которых надувает шарик за T_i минут, однако каждый раз после надувания Z_i шариков устает и отдыхает Y_i минут. Теперь организаторы праздника хотят узнать, через какое время будут надуты все шарики при наиболее оптимальной работе помощников, и сколько шариков надует каждый из них. (Если помощник надул шарик, и должен отдохнуть, но больше шариков ему надувать не придется, то считается, что он закончил работу сразу после окончания надувания последнего шарика, а не после отдыха).

Формат входных данных

В первой строке входных данных задаются числа M и N ($0 \leq M \leq 15000, 1 \leq N \leq 1000$). Следующие N строк содержат по три целых числа — T_i , Z_i и Y_i соответственно ($1 \leq T_i, Y_i \leq 100, 1 \leq Z_i \leq 1000$).

Формат выходных данных

Выведите в первой строке число T — время, за которое будут надуты все шарики. Во второй строке выведите N чисел — количество шариков, надутых каждым из приглашенных помощников. Разделяйте числа пробелами. Если распределений шариков несколько, выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
1 2 2 1 1 1 1 2	1 0 1
2 2 1 1 1 1 1 1	1 1 1

Задача D. Число

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

256 мегабайт

Вася написал на длинной полоске бумаги большое число и решил похвастаться своему старшему брату Пете этим достижением. Но только он вышел из комнаты, чтобы позвать брата, как его сестра Катя вбежала в комнату и разрежала полоску бумаги на несколько частей. В результате на каждой части оказалось одна или несколько идущих подряд цифр.

Теперь Вася не может вспомнить, какое именно число он написал. Только помнит, что оно было очень большое. Чтобы утешить младшего брата, Петя решил выяснить, какое максимальное число могло быть написано на полоске бумаги перед разрезанием.

Помогите ему!

Формат входных данных

Входной файл содержит одну или более строк, каждая из которых содержит последовательность цифр. Количество строк во входном файле не превышает 100, каждая строка содержит от 1 до 100 цифр. Гарантируется, что хотя бы в одной строке первая цифра отлична от нуля.

Формат выходных данных

Выведите в выходной файл одну строку — максимальное число, которое могло быть написано на полоске перед разрезанием.

Примеры

стандартный ввод	стандартный вывод
2 20 004 66	66220004
3	3

Задача Е. Сокровище

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Это интерактивная задача.

Дана полоска из n клеток, в одной из которых закопан клад. У Ёши есть металлоискатель, который Ёша помещает последовательно в одну из клеток. Начиная со второго шага металлоискатель показывает, стал ли прибор ближе или дальше от клада по сравнению с предыдущим шагом:

- если на текущем шаге металлоискатель стал ближе, чем на предыдущем, он показывает «+»;
- если на текущем шаге металлоискатель стал дальше, чем на предыдущем, он показывает «-»;
- если расстояние до клада не изменилось, металлоискатель показывает «0».

Ёша хочет определить клетку, в которой закопан клад, не более чем за 60 шагов. Помогите Ёше.

Протокол взаимодействия

Первая строка содержит целое число n ($1 \leq n \leq 10^9$). Далее программа участника выводит запросы.

Чтобы поместить металлоискатель в клетку i , выведите в отдельной строке «? i » ($1 \leq i \leq n$). Начиная со второго запроса, в ответ решение получит строку, содержащую один символ: «+», «-» или «0». Всего можно вывести такой запрос не более 60 раз.

Чтобы дать ответ i , выведите в отдельной строке «! i ». После этого сразу завершите работу программы.

После вывода каждого запроса не забудьте вывести перевод строки и сбросить буфер вывода. В противном случае вы получите вердикт «Решение зависло». Для сброса буфера используйте:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `stdout.flush()` в Python;
- смотрите документацию для других языков.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
6	? 1
	? 3
+	? 5
0	? 6
-	! 4

Задача G. When democracy fails

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В одной демократической стране приближаются парламентские выборы. Выборы проходят по следующей схеме: каждый житель страны, достигший восемнадцатилетнего возраста, отдает свой голос за одну из политических партий. После этого партия, которая набрала максимальное количество голосов, считается победившей на выборах и формирует правительство. Если несколько партий набрали одинаковое максимальное количество голосов, то они должны сформировать коалиционное правительство, что обычно приводит к длительным переговорам.

Один бизнесмен решил выгодно вложить свои средства и собрался поддержать на выборах некоторые партии. В результате поддержки он планирует добиться победы одной из этих партий, которая затем сформирует правительство, которое будет действовать в его интересах. При этом возможность формирования коалиционного правительства его не устраивает, поэтому он планирует добиться строгой победы одной из партий.

Чтобы повлиять на исход выборов, бизнесмен собирается выделить деньги на агитационную работу среди жителей страны. Исследование рынка показало, что для того, чтобы один житель сменил свои политические воззрения, требуется потратить одну условную единицу. Кроме того, чтобы i -я партия в случае победы сформировала правительство, которое будет действовать в интересах бизнесмена, необходимо дать лидеру этой партии взятку в размере p_i условных единиц. При этом некоторые партии оказались идеологически устойчивыми и не согласны на сотрудничество с бизнесменом ни за какие деньги.

По результатам последних опросов известно, сколько граждан планируют проголосовать за каждую партию перед началом агитационной компании. Помогите бизнесмену выбрать, какую партию следует подкупить, и какое количество граждан придется убедить сменить свои политические воззрения, чтобы выбранная партия победила, учитывая, что бизнесмен хочет потратить на всю операцию минимальное количество денег.

Формат входных данных

В первой строке вводится целое число n – количество партий ($1 \leq n \leq 10^5$). Следующие n строк описывают партии. Каждая из этих строк содержит по два целых числа: r_i – количество жителей, которые собираются проголосовать за эту партию перед началом агитационной компании, и b_i – взятка, которую необходимо дать лидеру партии для того, чтобы сформированное ей в случае победы правительство действовало в интересах бизнесмена ($1 \leq r_i \leq 10^6$, $1 \leq b_i \leq 10^6$ или $b_i = -1$). Если партия является идеологически устойчивой, то $b_i = -1$. Гарантируется, что хотя бы одно b_i не равно -1 .

Формат выходных данных

В первой строке выведите минимальную сумму, которую придется потратить бизнесмену. Во второй строке выведите номер партии, лидеру которой следует дать взятку. В третьей строке выведите n целых чисел – количество голосов, которые будут отданы за каждую из партий после осуществления операции. Если оптимальных решений несколько, выведите любое.

Примеры

стандартный ввод	стандартный вывод
3 7 -1 2 8 1 2	6 3 3 2 5
2 239 239 238 -1	239 1 239 238

Задача Н. Битовая угадка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Это интерактивная задача.

Кира загадал некоторое положительное целое число n , а Хаято требуется его угадать.

Изначально Кира дает Хаято значение `cnt` — количество единичных битов в двоичной записи n . Для того чтобы угадать n , Хаято может делать лишь операции одного типа: выбрать некоторое целое число x и вычесть его из n . Обратите внимание, что после каждой операции число n **изменяется**. Кира не любит плохих запросов, поэтому если Хаято попытается вычесть число x большее, чем n , то он проиграет Кире. После каждой операции Кира сообщает обновленное значение `cnt` — количество единичных битов в двоичной записи обновленного значения n .

У Кире не так много терпения, поэтому Хаято должен угадать **изначальное** значение числа n за не более чем 30 запросов.

Так как Хаято учится в начальной школе, он просит вашей помощи — напишите программу, которая угадывает число n . Кира — честный человек, поэтому он зафиксирует значение n изначально и **не станет** как-то менять начальное число в зависимости от запросов Хаято.

Формат входных данных

Во входных данных находятся несколько наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 500$) — количество наборов входных данных. Далее следуют наборы входных данных.

Первая строка каждого набора входных данных содержит одно целое число `cnt` — изначальное количество единичных битов в двоичной записи n .

Загаданное целое число n удовлетворяет следующему ограничению $1 \leq n \leq 10^9$.

Протокол взаимодействия

Для угадывания n вы можете не более 30 раз произвести описанную операцию. Для этого выведите строку в формате «- x » ($1 \leq x \leq 10^9$).

После этой операции из n вычитается число x , и поэтому значение n **изменяется**. Если число x **оказалось больше, чем текущее число n** , то такой запрос считается **некорректным**.

После каждой операции считайте одно неотрицательное целое число `cnt` — количество единичных битов в двоичной записи текущего n после совершения операции.

Когда вы узнаете число n , выведите одну строку следующего формата: «! n » ($1 \leq n \leq 10^9$).

После этого переходите к решению следующего набора входных данных или завершите программу, если таких нет.

Если ваша программа делает более 30 операций для одного набора входных данных, вычитет число x большее, чем n , или сделает некорректный запрос, то ответом на запрос будет -1, после получения такого ответа ваша программа должна немедленно завершиться, чтобы получить вердикт **Неправильный ответ**. Иначе она может получить любой другой вердикт.

После вывода запроса не забудьте вывести перевод строки и сбросить буфер вывода. В противном случае вы получите вердикт **Решение «зависло»**. Для сброса буфера используйте:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `stdout.flush()` в Python;

- смотрите документацию для других языков.

Взломы

Чтобы сделать взлом, используйте следующий формат.

Первая строка должна содержать одно целое число t ($1 \leq t \leq 500$).

Каждый набор входных данных должен содержать одно целое число n ($1 \leq n \leq 10^9$) на отдельной строке.

Пример

стандартный ввод	стандартный вывод
3	- 1
1	! 1
0	- 1
1	- 1
1	! 2
0	- 2
2	- 1
1	! 3
0	

Замечание

К примеру, количество единичных битов в числе 6 это 2, потому что двоичное представление 6 это 110. Для 13 количество единичных битов 3, так как $13_{10} = 1101_2$.

В первом примере $n = 1$, поэтому на вход дается число 1. После вычитания из n единицы оно становится нулевым, поэтому и количество единичных битов у него 0.

В третьем примере $n = 3$, которое в двоичном представлении выглядит как $3_{10} = 11_2$, следовательно на вход даётся количество единиц, то есть 2. После вычитания числа 2 мы имеем $n = 1$, поэтому количество единичных битов теперь 1. После вычитания из n единицы оно становится равно нулю.

Задача I. Сделайте перестановку

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У вас есть массив целых чисел a длины n . Вы можете выполнять два вида операций.

- Удалить целое число из a . Эта операция стоит c .
- Вставить произвольное натуральное число x в любую позицию a (в начало, в конец или между любыми двумя соседними элементами). Эта операция стоит d .

Вы хотите сделать конечный массив перестановкой **любой** положительной длины. Пожалуйста, выведите минимальную стоимость, чтобы сделать из массива перестановку. Обратите внимание, что вы можете сделать массив пустым во время операций, но окончательный массив должен содержать хотя бы одно целое число.

Перестановкой длины n называется массив, состоящий из n различных целых чисел от 1 до n в произвольном порядке. Например, $[2, 3, 1, 5, 4]$ — это перестановка, но $[1, 2, 2]$ не является перестановкой (2 встречается в массиве дважды), также как и $[1, 3, 4]$ не является перестановкой ($n = 3$, но в массиве есть элемент равный 4).

Формат входных данных

Каждый тест содержит несколько наборов входных данных. Первая строка содержит количество наборов входных данных t ($1 \leq t \leq 10^4$). Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит три целых числа n, c, d ($1 \leq n \leq 10^5$, $1 \leq c, d \leq 10^9$).

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$).

Гарантируется, что сумма всех n не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите минимальную стоимость, которую нужно потратить, чтобы сделать из массива перестановку.

Пример

стандартный ввод	стандартный вывод
8	0
3 3 3	2
1 2 3	8
5 1 5	14
1 2 3 5 6	20
5 2 3	3
1 1 1 3 3	12
5 1 10	999999998
2 4 6 8 10	
6 2 8	
7 3 5 4 4 8	
4 10 1	
1 2 6 7	
4 3 3	
2 5 8 7	
2 1000000000 1	
1000000000 1	

Замечание

В первом наборе входных данных массив уже является перестановкой, поэтому операции не нужны.

Во втором наборе входных данных мы можем удалить числа 5, 6 и получить перестановку $[1, 2, 3]$. Стоимость таких операций будет равна 2. Обратите внимание, что мы также можем получить перестановку, вставив число 4, но это стоит 5.

В третьем наборе входных данных мы можем просто удалить все числа, кроме первой 1. Это стоит 8, а окончательный массив $[1]$ представляет собой перестановку длины 1.

В четвертом наборе входных данных мы можем удалить все числа, кроме 2, и вставить одно число 1 на первую позицию. Это стоит $4 + 10 = 14$, а окончательный массив $[1, 2]$ представляет собой перестановку длины 2.