

Задача А. Звёздный путь

Имя входного файла:	stdin
Имя выходного файла:	stdout
Ограничение по времени:	1 секунда
Ограничение по памяти:	128 МБ

Экспедиция готовится отправиться в путь на космическом корабле нового поколения. Планируется последовательно посетить N планет звёздной системы — от планеты Земля до планеты Победа. Планеты пронумерованы от 1 до N в порядке их посещения, Земля имеет номер 1, а Победа — номер N .

Для перелёта между планетами корабль может использовать любой тип топлива, существующий в звёздной системе. Перед началом экспедиции корабль находится на планете Земля, и бак корабля пуст. Существующие типы топлива пронумерованы целыми числами, на планете с номером i можно заправиться только топливом типа a_i . При посещении i -й планеты можно заправиться, полностью освободив бак от имеющегося топлива и заполнив его топливом типа a_i .

На каждой планете станция заправки устроена таким образом, что в бак заправляется ровно столько топлива, сколько потребуется для перелёта до следующей планеты с топливом такого же типа. Если далее такой тип топлива не встречается, заправиться на этой планете невозможно. Иначе говоря, после заправки на i -й планете топлива хватит для посещения планет от $(i + 1)$ -й до j -й включительно, где j — минимальный номер планеты, такой что $j > i$ и $a_j = a_i$. Для продолжения экспедиции дальше j -й планеты корабль необходимо снова заправить на одной из этих планет.

Требуется написать программу, которая по заданным типам топлива на планетах определяет минимальное количество заправок, требуемых для экспедиции.

Формат входных данных

В первой строке входного файла записано число N ($2 \leq N \leq 300\,000$) — количество планет.

Во второй строке входного файла записано N целых чисел a_1, a_2, \dots, a_N ($1 \leq a_i \leq 300\,000$) — типы топлива на планетах.

Формат выходных данных

В первой строке выходного файла выведите единственное число K — минимальное количество заправок, которые нужно произвести.

Во второй строке выведите K чисел, разделённых пробелами, — номера планет, на которых требуется заправиться. Номера планет требуется выводить в порядке времени заправок.

Если решений с минимальным количеством заправок несколько, выведите любое из них. Если решения не существует, выведите число 0.

Система оценивания

Данная задача содержит две подзадачи. Для оценки каждой подзадачи используется своя группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы успешно пройдены.

Подзадача 1

$N \leq 3000$. Подзадача оценивается в 50 баллов.

Подзадача 2

$N \leq 300\,000$. Подзадача оценивается в 50 баллов.

Примеры

expedition.in	expedition.out
7 1 3 2 1 3 2 3	3 1 3 5
7 4 3 2 4 3 2 1	0

Задача В. Киноакадемия

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: **1 секунда**
Ограничение по памяти: **128 МБ**

В финал конкурса Киноакадемии вышли n лучших кинофильмов 2014 года. В конкурсе награждаются фильмы в двух номинациях: лучшая режиссура и лучший сценарий. По правилам конкурса в каждой номинации должен быть награжден ровно один фильм, причём в разных номинациях — разные фильмы.

В ходе многочисленных опросов зрителей и кинокритиков удалось собрать данные, показывающие, какой уровень ликования вызовет победа каждого фильма в каждой из номинаций. Дотошные журналисты на этом не остановились и дополнительно выяснили, каким будет уровень ликования, если тот или иной фильм не выиграет ни в одной из номинаций.

Требуется написать программу, которая по результатам опросов определяет наибольший суммарный уровень ликования, которого можно добиться выбором фильмов для награждения в указанных номинациях.

Формат входных данных

В первой строке входного файла задано целое число n — количество кинофильмов, участвующих в финале конкурса Киноакадемии. В следующих n строках содержатся по три целых числа a_i, b_i, c_i — уровень ликования, если i -й фильм не выиграет ни в одной из номинаций, уровень ликования, если этот фильм выиграет в номинации на лучшую режиссуру, и уровень ликования, если этот фильм выиграет в номинации на лучший сценарий.

Формат выходных данных

Первая строка выходного файла должна содержать одно число — наибольший возможный суммарный уровень ликования. Вторая строка должна содержать два целых числа — номера фильмов-победителей в номинациях лучшая режиссура и лучший сценарий соответственно. Фильмы нумеруются натуральными числами от 1 до n . Если оптимальных способов выбора награждаемых фильмов несколько, можно вывести любой из них.

Система оценки

Данная задача содержит три подзадачи. Для оценки каждой подзадачи используется своя группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы пройдены.

Подзадача 1

$2 \leq n \leq 100$
 $1 \leq a_i, b_i, c_i \leq 10^5$

Подзадача оценивается в 20 баллов.

Подзадача 2

$2 \leq n \leq 2000$
 $1 \leq a_i, b_i, c_i \leq 10^5$

Подзадача оценивается в 25 баллов.

Подзадача 3

$2 \leq n \leq 10^5$
 $1 \leq a_i, b_i, c_i \leq 10^9$

Подзадача оценивается в 55 баллов.

Пример

cinema.in	cinema.out
3	17
3 6 9	2 3
1 5 7	
1 3 9	

Пояснение к примеру

В приведенном примере наибольший суммарный уровень ликования равен $3 + 5 + 9 = 17$.

Задача С. Оборона крепости

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: **1 секунда**
Ограничение по памяти: **128 МБ**

Стена осаждённой крепости состоит из n участков, пронумерованных от 1 до n . Разведка доложила, что противник планирует отправить на штурм участка стены с номером i отряд из a_i нападающих. Для обороны крепости на участки стены будут направлены в общей сложности s защитников.

Участки стены различаются качеством укрепления, что приводит к различной эффективности обороны: на участке стены с номером i каждый защитник способен отразить атаку k_i нападающих.

Пусть на участок с номером i отправлено x_i защитников. Тогда если количество нападающих не превышает величину $x_i \cdot k_i$, то на этом участке ни один из нападающих не прорвётся в крепость. Иначе в крепость прорвутся $(a_i - x_i \cdot k_i)$ нападающих.

Требуется написать программу, распределяющую защитников по участкам стены так, чтобы их общее количество было равно s и в крепость прорвалось наименьшее количество нападающих.

Формат входных данных

Первая строка входных данных содержит целые числа n — количество участков стены и s — количество защитников крепости ($1 \leq n \leq 100\,000$; $1 \leq s \leq 10^9$).

Следующие n строк содержат по два целых числа a_i , k_i — общее количество нападающих на i -й участок стены и количество нападающих, атаку которых может отразить один защитник этого участка ($1 \leq a_i, k_i \leq 10^9$).

Формат выходных данных

Выходные данные должны содержать единственное целое число — минимальное количество нападающих, которые прорвутся в крепость.

Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения				Необх. подзадачи
		n	s	a	k	
1	17	$1 \leq n \leq 100$	$1 \leq s \leq 10\,000$	$1 \leq a_i \leq 100$	$k_i = 1$	
2	21	$1 \leq n \leq 100$	$1 \leq s \leq 10\,000$	$1 \leq a_i \leq 100$	$1 \leq k_i \leq 2$	1
3	23	$1 \leq n \leq 100$	$1 \leq s \leq 10\,000$	$1 \leq a_i \leq 100$	$1 \leq k_i \leq 100$	1, 2
4	39	$1 \leq n \leq 100\,000$	$1 \leq s \leq 10^9$	$1 \leq a_i \leq 10^9$	$1 \leq k_i \leq 10^9$	1 – 3

Примеры

<code>castle.in</code>	<code>castle.out</code>
1 10 8 1	0
3 3 4 2 1 1 10 8	3

Пояснения к примерам

В первом тесте ни один из нападающих не прорвется в крепость, если поставить всех 10 защитников на единственный участок, так как они смогут отбить всех нападающих. Во втором примере можно, например, направить двух защитников на первый участок и одного — на третий.

Задача D. Школа олимпийского резерва

Имя входного файла:	stdin
Имя выходного файла:	stdout
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	128 МБ
Максимальная оценка	100 баллов

Для подготовки к чемпионату мира по футболу 2018 года создается школа олимпийского резерва. В нее нужно зачислить M юношей 1994–1996 годов рождения. По результатам тестирования каждому из N претендентов был выставлен определенный балл, характеризующий его мастерство. Все претенденты набрали различные баллы. В составе школы олимпийского резерва хотелось бы иметь A учащихся 1994 г.р., B – 1995 г.р. и C – 1996 г.р. ($A + B + C = M$). При этом минимальный балл зачисленного юноши 1994 г.р. должен быть больше, чем минимальный балл зачисленного 1995 г.р., а минимальный балл зачисленного 1995 г.р. должен быть больше, чем минимальный балл зачисленного 1996 г.р. Все претенденты, набравшие балл больше минимального балла для юношей своего года рождения, также должны быть зачислены.

В базе данных для каждого претендента записаны год его рождения и тестовый балл. Требуется определить, сколько нужно зачислить юношей каждого года рождения M_{94} , M_{95} и M_{96} ($M_{94} + M_{95} + M_{96} = M$), чтобы значение величины $F = |M_{94} - A| + |M_{95} - B| + |M_{96} - C|$ было минимально, все правила, касающиеся минимальных баллов зачисленных, были соблюдены, и должен быть зачислен хотя бы один юноша каждого требуемого года рождения.

Формат входных данных

В первой строке входного файла находится число K – количество наборов входных данных. Далее следуют описания каждого из наборов. В начале каждого набора расположены три натуральных числа A , B , C . Во второй строке описания находится число N – количество претендентов (гарантируется, что $N \geq A + B + C$). В каждой из следующих N строк набора содержатся два натуральных числа – год рождения (число 1994, 1995 или 1996 соответственно) и тестовый балл очередного претендента.

Формат выходных данных

Ответ на каждый тестовый набор выводится в отдельной строке. Если хотя бы одно из требований выполнить невозможно, то в качестве ответа следует вывести только число -1 . В противном случае соответствующая строка сначала должна содержать минимальное значение величины F , а затем три числа M_{94} , M_{95} и M_{96} , на которых это минимальное значение достигается, удовлетворяющие всем требованиям отбора. Если искомым вариантов несколько, то разрешается выводить любой из них.

Примеры

school.in	school.out
3	-1
1 1 1	0 1 1 1
4	-1
1994 3	
1994 4	

XXIII Всероссийская олимпиада школьников по информатике.

Заключительный этап. Пермь.

Первый тур, 13 апреля 2011 года

1996 1	
1996 2	
1 1 1	
3	
1995 2	
1994 3	
1996 1	
1 1 1	
3	
1994 1	
1995 2	
1996 3	
1	2 3 2 1
2 3 1	
7	
1996 2	
1994 7	
1994 4	
1996 1	
1995 3	
1994 5	
1995 6	

Комментарий

В первом примере на первом наборе ответ не существует, потому что нельзя пригласить хотя бы одного юношу 1995 г.р. Во втором наборе ответ существует и единственный, в третьем – нельзя выполнить правило относительно минимальных баллов.

Во втором примере правильным является также ответ 2 2 2 2.

Подзадачи и система оценки

Данная задача содержит четыре подзадачи. Для оценки каждой подзадачи используется своя группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы пройдены.

Подзадача 1 (25 баллов)

$K = 1; N \leq 100$; каждый претендент характеризуется своим баллом от 1 до N .

Подзадача 2 (25 баллов)

Сумма значений N по всем тестовым наборам не превосходит 10 000, каждый претендент характеризуется своим баллом от 1 до 10^9 .

Подзадача 3 (25 баллов)

Сумма значений N по всем тестовым наборам не превосходит 100 000, каждый претендент характеризуется своим баллом от 1 до N .

Подзадача 4 (25 баллов)

Сумма значений N по всем тестовым наборам не превосходит 300 000, каждый претендент характеризуется своим баллом в диапазоне от 1 до 10^9 .

Обратная связь

В течение тура можно не более 10 раз запросить баллы, которые набирает программа на тестах жюри. Запрос можно делать не чаще одного раза в 5 минут. Для каждой подзадачи сообщаются баллы за эту подзадачу. Детальные результаты на тестах не сообщаются.

В этой задаче можно выбрать, какое решение будет оцениваться. В этом случае баллы начисляются за лучшее решение из следующих:

- выбранного явно;
- последнего принятого на проверку решения.

Если выбор не сделан, то будет оцениваться лучшее решение из следующих:

- тех решений, по которым просмотрены баллы;
- последнего принятого на проверку решения.

Задача Е. ЛОВИТЬ ИЛИ НЕ ЛОВИТЬ

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: **1 секунда**
Ограничение по памяти: **128 МБ**

Владельцы рыболовецкого судна, ведущего промысел на реке Кама, решили в летнем сезоне оптимизировать свой бизнес.

Они получили сезонное разрешение на лов рыбы в n точках русла реки на расстояниях x_1, x_2, \dots, x_n километров от устья. При этом в точке с номером i разрешается выловить не более a_i тонн рыбы.

Выловленную рыбу можно продавать на m оптовых базах, расположенных вдоль берега реки в точках на расстояниях y_1, y_2, \dots, y_m километров от устья. При этом база в точке номер j готова в этом сезоне закупить не более b_j тонн рыбы по цене c_j рублей за тонну.

Расстояния от устья до точек вылова и оптовых баз измеряются вдоль русла реки.

Судно отправляется на лов из устья реки и должно вернуться туда же после окончания сезона. В течение сезона судно может произвольным образом плавать вверх и вниз по реке, останавливаясь для лова или продажи рыбы. Грузоподъёмность судна достаточна для перевозки любого количества выловленной рыбы. При удалении от устья судно движется против течения, расходуя на один километр пути топливо стоимостью p рублей. При перемещении в сторону устья судно движется по течению и поэтому не расходует топлива.

По итогам сезона прибыль за улов будет равна суммарной стоимости проданной рыбы за вычетом суммарной стоимости затраченного топлива.

Требуется написать программу, определяющую максимальную прибыль, которую можно получить за сезон.

Формат входных данных

Первая строка входных данных содержит три целых числа n , m и p — количество точек лова рыбы, количество оптовых баз и стоимость топлива ($1 \leq n, m \leq 500\,000$; $0 \leq p \leq 10^9$).

Следующие n строк содержат по два целых числа x_i и a_i — расстояние от устья и максимальный улов для каждой точки лова рыбы ($0 < x_1 < x_2 < \dots < x_n \leq 10^9$; $0 < a_i \leq 10^6$).

Следующие m строк содержат по три целых числа y_j , b_j , c_j — расстояние от устья, максимальное закупаемое количество тонн рыбы и цена закупки за тонну для каждой оптовой базы ($0 < y_1 < y_2 < \dots < y_m \leq 10^9$; $0 < b_j, c_j \leq 10^6$).

Формат выходных данных

Выходные данные должны содержать единственное целое число — максимальную возможную прибыль.

Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения		Необх. подзадачи
		n, m	Дополнительные	
1	16	$1 \leq n, m \leq 50\,000$	$p = 0$	
2	9	$1 \leq n, m \leq 50\,000$	$y_m < x_1$	1
3	16	$1 \leq n, m \leq 50\,000$	$x_n < y_1$	1
4	11	$1 \leq n, m \leq 1\,000$	—	
5	9	$1 \leq n, m \leq 8\,500$	—	4
6	20	$1 \leq n, m \leq 50\,000$	—	1–5
7	6	$1 \leq n, m \leq 200\,000$	—	1–6
8	7	$1 \leq n, m \leq 320\,000$	—	1–7
9	6	$1 \leq n, m \leq 500\,000$	—	1–8

Примеры

fisher.in	fisher.out
3 2 0 1 5 2 3 4 5 2 2 10 3 6 5	50
2 1 100 6 5 100 4 5 100 2000	9400
3 3 10 1 1 10 100 20 10 2 1000 1 11 50 50 17 50 2	2441

Пояснения к примерам

Во втором примере оптимальными будут следующие действия. Следует доплыть до точки на расстоянии 6 километров от устья, потратив 600 рублей на топливо, и выловить в ней 5 тонн рыбы. После этого следует спуститься на 1 километр по реке к базе на расстоянии 5 километров от устья и продать выловленную рыбу по цене 2000 рублей за тонну. Затем следует вернуться в устье реки. Суммарная прибыль составит 9400 рублей.