

Задача А. Добавление задач

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Жюри Всегалактической олимпиады по информатике собралось, чтобы обсудить задачи.

На олимпиаду нужно выбрать набор из шести задач. Изначально набор пуст. Авторы рассказывают свои задачи по очереди и обсуждают их между собой.

Будем для простоты считать, что каждой задаче соответствует вещественное число — её сложность, — и что сложности всех задач различны. Когда в набор добавляется i -я задача, жюри определяет её номер d_i в порядке возрастания сложности среди задач, которые выбраны к этому моменту. Например, когда в набор добавляется первая задача, среди выбранных она всегда имеет номер 1 (так как в момент её добавления других выбранных задач нет). При добавлении второй задачи у неё может быть номер 1 (если она проще первой добавленной) или номер 2 (если она сложнее), и так далее.

Даны номера задач в момент их добавления: d_1, d_2, \dots, d_6 . Пронумеруйте задачи среди всех выбранных в порядке возрастания сложности.

Формат входных данных

В первой строке записаны шесть целых чисел $d_1, d_2, d_3, d_4, d_5, d_6$ — номера задач в момент добавления ($1 \leq d_i \leq i$).

Формат выходных данных

Выведите шесть целых чисел — номер первой, второй, ..., шестой добавленной задачи среди всех выбранных.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 2 1 3 2 2	4 6 1 5 3 2
1 2 3 4 5 6	1 2 3 4 5 6
1 1 1 1 1 1	6 5 4 3 2 1

Пояснения к примерам

В первом примере порядок задач от простых к сложным такой: третья, шестая, пятая, первая, четвёртая, вторая. А значит, в итоговом порядке третья задача должна иметь номер 1, шестая — номер 2, и так далее.

Вот как получились номера во входных данных:

- первая задача получает номер 1;
- вторая задача сложнее первой, поэтому получает номер 2;
- третья задача проще первой и проще второй, получает номер 1;
- четвёртая задача сложнее первой и третьей, но проще второй, так что ей достаётся номер 3;
- пятая задача сложнее третьей и проще всех остальных взятых, она получает номер 2;
- шестая задача сложнее третьей и проще всех других, она получает номер 2.

Система оценки

В этой задаче три примера и 25 основных тестов. Тесты оцениваются по отдельности и дают по 4 балла.

Задача В. Сумма в двух действиях

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Ника и Аня участвуют в командной эстафете по скоростному суммированию чисел.

Эстафета должна была проходить так. Сначала Ника получает n чисел a_1, a_2, \dots, a_n , считает их сумму x и посылает Ане. Затем Аня получает сумму x , получает свои n чисел b_1, b_2, \dots, b_n , и должна выдать ответ — сумму всех $2n$ заданных им чисел.

Но коварная Ева взломала систему проведения эстафеты! Теперь Аня вместо того, чтобы получать отдельно x и отдельно n чисел b_1, b_2, \dots, b_n , получает только n чисел, причём **к каждому** прибавлено x . Другими словами, Аня получает числа $b_1 + x, b_2 + x, \dots, b_n + x$.

К счастью, девочки вовремя узнали об изменениях в системе. Как им действовать, чтобы всё-таки правильно посчитать итоговую сумму?

Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза. В каждом тесте все числа зафиксированы заранее. В конце каждой строки входных данных следует символ перевода строки.

Первый запуск

При первом запуске решение действует за Нику. В первой строке записано слово «first». Во второй строке записано целое число n — количество чисел для суммирования ($1 \leq n \leq 100$). В третьей строке заданы сами числа a_1, a_2, \dots, a_n ($1 \leq a_i \leq 100$).

Выведите число x , которое Ника попытается переслать Ане. Это может быть любое целое число от -10^{18} до $+10^{18}$.

Второй запуск

При втором запуске решение действует за Аню. В первой строке записано слово «second». Во второй строке записано целое число n — количество чисел для суммирования, такое же, как при первом запуске. В третьей строке заданы числа $b_1 + x, b_2 + x, \dots, b_n + x$ (для исходных чисел верно $1 \leq b_i \leq 100$, а число x ровно то, что Ника вывела при первом запуске).

Выведите одно целое число: сумму $a_1 + a_2 + \dots + a_n + b_1 + b_2 + \dots + b_n$.

Пример

На каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске.

Далее показаны два запуска какого-то решения на первом тесте.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
first 5 1 2 3 4 5	3

<i>стандартный ввод</i>	<i>стандартный вывод</i>
second 5 9 10 11 12 13	55

Пояснение к примеру

В примере $a = [1, 2, 3, 4, 5]$ и $b = [6, 7, 8, 9, 10]$. Поскольку Ника попыталась передать $x = 3$, Аня получила числа $[6 + 3, 7 + 3, 8 + 3, 9 + 3, 10 + 3]$. Тем не менее, Аня как-то смогла узнать правильную сумму: $1 + 2 + \dots + 9 + 10 = 55$. Интересно, как она это сделала?

Система оценки

В этой задаче две подзадачи. Чтобы получить баллы за подзадачу, нужно пройти все тесты этой подзадачи и всех предыдущих подзадач.

В первой подзадаче (50 баллов) сумма всех a_i **делится нацело** на n .

Во второй подзадаче (тоже 50 баллов) дополнительных ограничений нет.

Задача С. Три коробки

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Есть три прямоугольные плоские коробки: одна имеет размер $a_1 \times b_1$, другая $a_2 \times b_2$, третья $a_3 \times b_3$. Каждую коробку **можно** поворачивать на 90 градусов. Стенки коробок будем считать пренебрежимо тонкими — в частности, две одинаковые коробки можно положить одну в другую.

Ксюша выкладывает коробки на клетчатую плоскость, так, чтобы их стенки лежали вдоль линий сетки. Ксюша может использовать любое подмножество коробок и выкладывать их в любом порядке. Первую выбранную коробку можно положить как угодно.

Пусть уже выложены коробки B_1, B_2, \dots, B_k ($k > 0$). Тогда следующую коробку C нужно выложить *устойчиво*:

- для каждой коробки B_i коробка C лежит либо целиком внутри B_i , либо целиком снаружи B_i (в любом случае коробки могут касаться стенками);
- на какой-то горизонтальной стенке C есть отрезок положительной длины, которым C касается какой-то коробки B_i ;
- на какой-то вертикальной стенке C есть отрезок положительной длины, которым C касается какой-то коробки B_i .

Другими словами, коробка C упирается в стенки предыдущих выложенных коробок и в каком-то направлении по горизонтали (влево или вправо), и в каком-то направлении по вертикали (вверх или вниз). Для лучшего понимания взгляните на пример в конце условия и пояснение к нему.

Ксюша хочет, чтобы в результате получилась пустая прямоугольная область, со всех сторон ограниченная стенками коробок. Формально ей нужен прямоугольник, у которого каждая точка границы лежит на стенке какой-то коробки, а внутри стенок нет. Какие прямоугольники могут получиться у Ксюши? Прямоугольники, отличающиеся лишь порядком сторон, считаются одинаковыми.

Формат входных данных

В первой строке записаны числа a_1 и b_1 . Во второй строке записаны числа a_2 и b_2 . В третьей строке записаны числа a_3 и b_3 . Все числа целые от 1 до 9 включительно.

Формат выходных данных

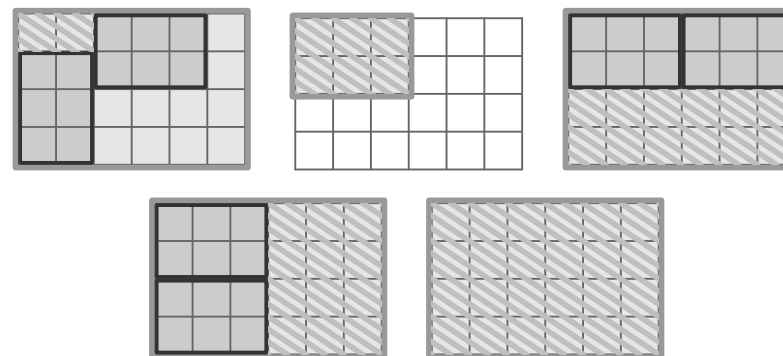
Выведите прямоугольные области, которые можно получить. В каждом прямоугольнике следует выводить стороны в порядке возрастания. Сами прямоугольники следует выводить упорядоченными по возрастанию как пары: по первой стороне, а при равенстве — по второй.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2	1 2
2 3	2 3
4 6	2 6
	3 4
	4 6

Пояснение к примеру

Как получить такие прямоугольники (на картинках заштрихованы):



На самой первой картинке Ксюша действует так:

- сначала кладёт коробку 4×6 ;
- затем в её нижний левый угол кладёт коробку 2×3 (при движении влево и вниз она упирается в стенки первой коробки);
- а затем кладёт оставшуюся коробку 2×3 (при движении влево она упирается во вторую коробку, а при движении вверх — в первую).

Система оценки

В этой задаче один пример и 20 основных тестов. Тесты оцениваются по отдельности и дают по 5 баллов.

Задача D. Конфеты и перезагрузка

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Вероятно, вы хорошо знаете следующую игру. Есть вазочки с конфетами. Двое играют в игру, делая ходы по очереди. На каждом ходу игрок выбирает одну вазочку и съедает из неё любое целое положительное количество конфет. Проигрывает тот, кто не может сделать ход.

Локи и Мёбиус играют почти в такую же игру. Единственное отличие — на каждой вазочке изначально есть кнопка «перезагрузки». На каждом ходу игрок по-прежнему выбирает одну вазочку — но теперь может либо взять конфеты, либо нажать на кнопку. При нажатии в вазочке становится ровно столько конфет, сколько было изначально, а кнопка исчезает.

Локи ходит первым. Сыграйте за него в игру, и выиграйте, если это возможно.

Протокол взаимодействия

Это интерактивная задача. Ваша программа играет за Локи, а программа жюри — за Мёбиуса.

В первой строке записано целое число n — количество вазочек с конфетами ($1 \leq n \leq 50$). Во второй строке заданы числа a_1, a_2, \dots, a_n — начальное количество конфет в вазочках ($1 \leq a_i \leq 50$).

Далее решение делает ходы за Локи, а программа жюри — за Мёбиуса. Локи ходит первым, далее игроки ходят по очереди. Каждый ход записывается строкой из двух чисел: первое — номер вазочки, а второе — количество конфет, которые следует съесть из неё этим ходом. Чтобы нажать на кнопку «перезагрузки», следует вывести 0 вместо количества конфет. Чтобы сдать, следует вывести -1 вместо номера вазочки и -1 вместо количества конфет.

Чтобы пройти тест, где Локи побеждает при правильной игре, следует добиться того, что Мёбиус сдастся. Чтобы пройти тест, где побеждает Мёбиус, следует сдать — неважно, сразу или после нескольких ходов.

Если решение выводит строку, не содержащую два числа, и завершает работу, оно получает вердикт «Presentation Error». Если решение выводит некорректный ход и завершает работу, оно получает вердикт «Wrong Answer».

Помните, что после вывода каждого хода нужно выводить **перевод строки**,

а затем **очищать буфер** вывода: например, функцией «`fflush(stdout)`» в языке C, «`cout.flush()`» в C++ или «`sys.stdout.flush()`» в Python. В противном случае очередной ход может не дойти до проверяющей программы, и тогда решение получит вердикт «Idleness Limit Exceeded».

Примеры

Пустые строки добавлены для удобства чтения.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1	
1	
1 0	1 1
-1 -1	1 1
2	
3 3	
1 0	2 0
2 2	1 2
	-1 -1

Система оценки

В этой задаче четыре подзадачи. Чтобы получить баллы за подзадачу, нужно пройти все тесты этой подзадачи и всех предыдущих подзадач. Чтобы решение проверялось на основных тестах, нужно пройти первый пример (не обязательно проходить оба примера).

В первой подзадаче (10 баллов) $n = 1$.

Во второй подзадаче (20 баллов) $n = 2$.

В третьей подзадаче (30 баллов) $n = 3$.

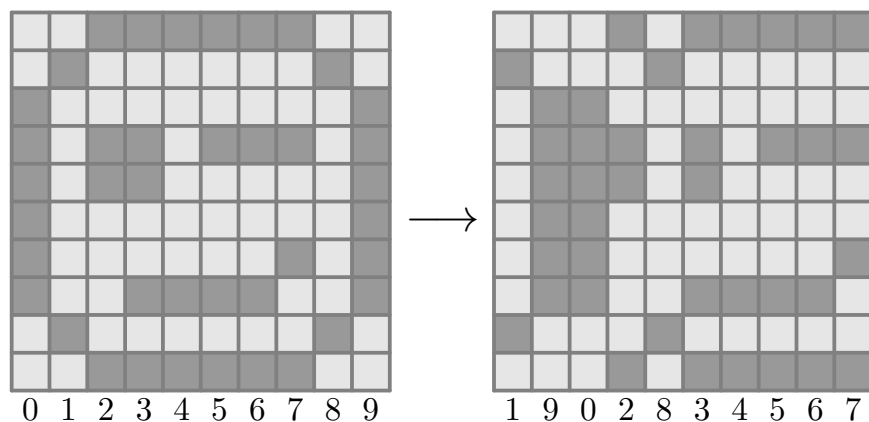
В четвёртой подзадаче (40 баллов) дополнительных ограничений нет.

Задача Е. Почта в Сказочном мире

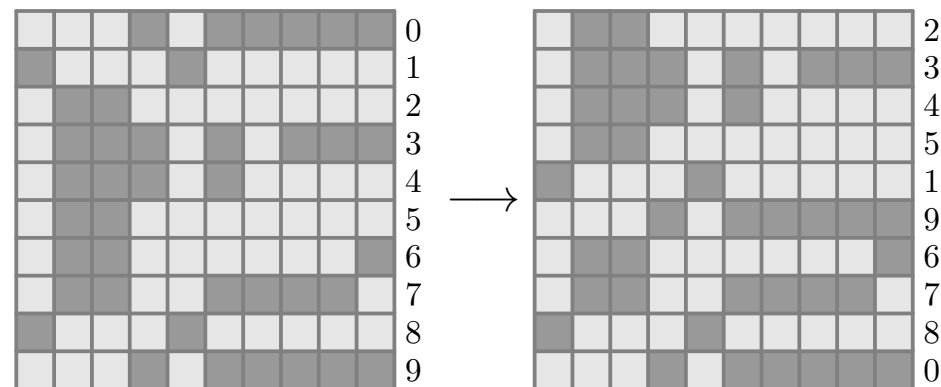
Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 512 мегабайт

В Сказочном мире, в Волшебной стране, живут Иван-Царевич и Василиса Премудрая. Единственное средство связи в этом мире — почта, которая пересылает небольшие сообщения: картинки, состоящие из 10×10 клеток, каждая из которых либо остаётся белой, либо красится в чёрный.

Иван-Царевич отправляется на Край света. Почта оттуда идёт через Тридевятое царство. Там каждое сообщение делят на 10 вертикальных полосок, и каждую полоску переносит белочка. Потом полоски заново склеивают в картинку 10×10 . Но разным белочкам требуется на транспортировку полосок разное время. Так что вертикальные полоски оказываются склеены в случайном порядке. Вот пример того, что может произойти с картинкой в Тридевятом царстве:



Далее почта попадает в Тридесятое государство. Там каждое сообщение делят на 10 горизонтальных полосок, и каждую полоску переносит зайчик. Потом полоски заново склеивают в картинку 10×10 . Но разным зайчикам требуется на транспортировку полосок разное время. Так что и горизонтальные полоски оказываются склеены в случайном порядке. Вот пример того, что может произойти с картинкой в Тридесятом государстве:



После всего этого сообщение приходит в Волшебную страну.

Василиса Премудрая хочет, чтобы Иван-Царевич мог передать ей одной картинкой целое число. Помогите им договориться, как передавать картинками числа, чтобы Василиса Премудрая всегда могла понять, что за число хотел передать Иван-Царевич.

Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза. В каждом тесте все числа и перемешивания зафиксированы заранее. В конце каждой строки входных данных следует символ перевода строки.

Первый запуск

При первом запуске решение получает число n и рисует картинку. В первой строке записано целое число t — количество тестовых случаев ($1 \leq t \leq 1000$). Во второй строке записано слово «transmit». В следующих t строках заданы тестовые случаи: каждый — одно целое число n , которое надо передать ($1 \leq n \leq 10^9$).

Выведите t картинок, по одной на каждый тестовый случай. Каждая картинка занимает 10 строк, каждая из которых состоит из 10 символов, а каждый символ — 0 или 1 (цифры соответствуют белому и чёрному цвету). Для удобства можно выводить любое количество пустых строк до и после каждой картинки.

Второй запуск

При втором запуске решение получает картинки, нарисованные при первом запуске и прошедшие через Тридевятое Царство и Тридесятое Государство. В первой строке записано целое число t — количество тестовых случаев,

Олимпиада СПбГУ по информатике среди школьников
Финальный тур, воскресенье, 26 февраля 2023 года

то же, что при первом запуске. Во второй строке записано слово «receive». Далее следуют тестовые случаи. Каждый тестовый случай занимает 11 строк: первая строка пустая, а следующие 10 — картинка в описанном выше формате.

Выведите t строк, по одной на каждый тестовый случай. Ответом на каждый тестовый случай должно быть целое число n , которое надо было передать.

Пример

На каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске.

Далее показаны два запуска какого-то решения на первом тесте.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	0000000000
transmit	0011111000
5	0010000000
2	0010000000
	0011111000
	0000001000
	0000001000
	0010001000
	0001110000
	0000000000
	0000000000
	0001110000
	0010001000
	0000001000
	0000001000
	0000010000
	0000100000
	0001000000
	0011111000
	0000000000

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	5
receive	2
0000000001	
0000000000	
0001000000	
0101101001	
0001000001	
0100101000	
0000000000	
0000000001	
0101101000	
0001000000	
1001000111	
0000000000	
0000000010	
0001000101	
1000000010	
0000000001	
0000000000	
0000000010	
0000000100	
0001000000	

Система оценки

В этой задаче один пример и 25 основных тестов с номерами от 2 до 26, которые оцениваются отдельно и дают по 4 балла каждый. Ограничения:

1	$1 \leq n \leq 5$	2	$1 \leq n \leq 10$	3	$1 \leq n \leq 20$
4	$1 \leq n \leq 50$	5	$1 \leq n \leq 100$	6	$1 \leq n \leq 200$
7	$1 \leq n \leq 500$	8	$1 \leq n \leq 1000$	9	$1 \leq n \leq 2000$
10	$1 \leq n \leq 5000$	11	$1 \leq n \leq 10\,000$	12	$1 \leq n \leq 20\,000$
13	$1 \leq n \leq 50\,000$	14	$1 \leq n \leq 100\,000$	15	$1 \leq n \leq 200\,000$
16	$1 \leq n \leq 500\,000$	17	$1 \leq n \leq 1\,000\,000$	18	$1 \leq n \leq 2\,000\,000$
19	$1 \leq n \leq 5\,000\,000$	20	$1 \leq n \leq 10\,000\,000$	21	$1 \leq n \leq 20\,000\,000$
22	$1 \leq n \leq 50\,000\,000$	23	$1 \leq n \leq 100\,000\,000$	24	$1 \leq n \leq 200\,000\,000$
25	$1 \leq n \leq 500\,000\,000$	26	$1 \leq n \leq 1\,000\,000\,000$		

Задача F. Сложение сжатых чисел

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 512 мегабайт

Заданы два положительных целых числа, a и b . Найдите их сумму.

В этой задаче для чисел используется *сжатая запись*. В такой записи могут встречаться обычные цифры, «0123456789», в любом количестве и в любом порядке. А ещё могут встречаться *повторения*: блоки вида « $(d|r)$ », означающие, что последовательность цифр d нужно повторить r раз. К примеру, запись «123» — это просто число (сто двадцать три), а запись «1(23|4)56» соответствует числу 12323232356 в обычной записи.

В обеих частях *повторения* — до и после разделителя — записаны только цифры, то есть *повторения не вкладываются* друг в друга. Левая часть *повторения* может быть любой непустой строкой из цифр. Правая часть *повторения* — положительное целое число без ведущих нулей. Эти правила соблюдаются для входных данных. Их же следует соблюдать при выводе.

Формат входных данных

В первой строке записано число a , а во второй — число b . Оба числа целые, положительные и заданы в *сжатой записи*. Длина каждой строки не превосходит 100 символов.

Формат выходных данных

Выведите результат сложения a и b в *сжатой записи* в отдельной строке. Можно вывести любую запись правильного ответа, состоящую не более чем из 100 000 символов.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
2 3	5	2 + 3 = 5
(2 11)(3 2) (4 12)(5 1)	6666666666(67 1)8	222222222233 + 4444444444445 = 6666666666678
1(23 4)56 7(8 11)(0 1)	7(9 1)0(12 4)36	12323232356 + 7888888888880 = 7901212121236
0(01 1234567890) 0(02 1234567890)	0(03 1234567890)	1010101...10101 + 2020202...20202 = 3030303...30303

Система оценки

В этой задаче четыре подзадачи. Чтобы получить баллы за подзадачу, нужно пройти все тесты, удовлетворяющие её ограничениям. Примеры соответствуют подзадачам. Чтобы решение проверялось на основных тестах, нужно пройти первый пример (не обязательно проходить все четыре примера).

В первой подзадаче (10 баллов) записи a и b не содержат *повторений*.

Во второй подзадаче (55 баллов) левая часть каждого *повторения* состоит из одной цифры, а правая — не более чем из девяти цифр.

В третьей подзадаче (32 балла) правая часть каждого *повторения* состоит не более чем из девяти цифр.

В четвёртой подзадаче (3 балла) дополнительных ограничений нет.