

Задача А. Электронный счетчик

Правительство Байтландии приняло решение о поддержке отечественных производителей электронных компонент. Одна из компаний разработала жидкокристаллические дисплеи для электронных счетчиков, которые могут отображать цифры. Каждая цифра изображается с помощью семи сегментов, каждый из которых может быть включен или выключен (вы наверняка видели подобные на электронных часах). Изображение цифр приведено на рисунке:



Все счетчики будут вести отсчет от нуля (он отображается на экране в начале работы) до некоторого числа k , последовательно проходя по всем значениям. Ведущие нули в числах, отображаемых на счетчиках, не отображаются, все сегменты перед первой значащей цифрой выключены.

За одну операцию можно изменить состояние одного сегмента (включить или выключить). Компания получила заказ на изготовление n типов счетчиков, для каждого из типов известно максимальное значение счетчика k_i . Для каждого из типов счетчика определите, какое количество операций по изменению состояния сегмента необходимо будет произвести для прохода по всем значениям от нуля до k_i .

В первой строке входных данных записано число t — количество различных типов счетчиков. В следующих t строках записаны числа k_i . Ответ для каждого типа счетчика выведите в отдельной строке.

В первом тесте $t = 3$, k_i не превосходит 100. Оценка за этот тест: 30 баллов. За каждое правильно подсчитанное число операций начисляется 10 баллов. Проверка осуществляется в режиме online (результат виден сразу).

Во втором тесте $t = 700$, k_i не превосходит 10^9 . Оценка за этот тест: 70 баллов. За каждое правильно подсчитанное число операций начисляется 0.1 балла. Во время тура проверяется, что сданный файл содержит 700 чисел. Проверка правильности ответа осуществляется в режиме offline (результат виден после окончания тура).

Примеры

| Входные данные | Результат |
|----------------|-----------|
| 3 | 9 |
| 2 | 32 |
| 10 | 11 |
| 3 | |

Задача В. Эффективные скидки

В связи с тяжелой экономической ситуацией байтландский интернет-магазин стал получать меньше прибыли. Однако эффективные менеджеры нашли неожиданный способ увеличить доходы — предоставлять клиентам скидки!

Для каждого заказа известна стоимость купленных товаров, стоимость доставки, а также сумма, начиная с которой доставка бесплатна (то есть если стоимость купленных товаров больше либо равна этой сумме, то доставка не оплачивается). Все стоимости выражаются целым количеством байтландских бурлей.

На товары в заказе можно предоставить скидку, скидка также выражается целым количеством байтландских бурлей.

Для каждого заказа определите максимальную суммарную стоимость купленных товаров и стоимость доставки, с учетом возможности предоставления скидки.

В первой строке входных данных записано число t — количество различных заказов. В следующих t строках записаны по три числа $cost$ — стоимость купленных товаров, $deliverycost$ — стоимость доставки и $freedelivery$ — стоимость купленных товаров, начиная с которой доставка бесплатна. Ответ для каждого заказа выведите в отдельной строке.

В первом тесте $t = 3$. Оценка за этот тест: 30 баллов. За каждую правильно подсчитанную максимальную стоимость начисляется 10 баллов. Проверка осуществляется в режиме online (результат виден сразу).

Во втором тесте $t = 700$. Оценка за этот тест: 70 баллов. За каждую правильно подсчитанную максимальную стоимость начисляется 0.1 балла. Во время тура проверяется, что сданный файл содержит 700 чисел. Проверка правильности ответа осуществляется в режиме offline (результат виден после окончания тура).

Примеры

| Входные данные | Результат |
|----------------|-----------|
| 4 | 1000 |
| 1000 100 10 | 1000 |
| 1000 100 100 | 1099 |
| 1000 100 1000 | 1100 |
| 1000 100 10000 | |

Задача С. Черно-белый экран

Электронная промышленность Байтландии развивается и одна из компаний начала выпускать черно-белые квадратные экраны размером $n \times n$ пикселей.

Для экономии траффика был разработан специальный режим экрана, в котором при вызове функции с параметрами row, col изменяется состояние всех пикселей в строке row и столбце col (то есть черные пиксели становятся белыми, а белые — черными). Пиксел на пересечении строки row и col также меняет свое состояние.

Чтобы продемонстрировать возможности экономии траффика необходимо разработать последовательность вызовов функции для формирования определенного изображения. Изначально весь экран черный (все пиксели находятся в состоянии 0).

В первой строке входных данных записано число t — количество различных изображений, которые необходимо сформировать на экране. Затем описывается t блоков. Описание блока содержит число n — размер экрана, а затем n строк по n чисел 0 или 1 в каждой — изображение, которое необходимо сформировать на экране.

В качестве ответа необходимо сформировать t блоков с параметрами вызова функции для формирования изображения на экране. Описание каждого блока должно состоять из числа k — количества вызовов функции для генерации изображения и k пар чисел row, col , задающих параметры функции. Нумерация начинается с левого верхнего угла, с нуля.

Формула оценки за каждый блок, при условии, что в результате вызовов функции получается требуемое изображение: $5 + 5 \times (BestAns/PartAns)^4$, где $PartAns$ — количество вызовов функции в решении участника, а $BestAns$ — минимальное количество вызовов функции среди решений всех участников и жюри.

В первом тесте $t = 3$. Оценка за этот тест: 30 баллов. Проверка осуществляется в режиме online (результат виден сразу).

Во втором тесте $t = 7$. Оценка за этот тест: 70 баллов. Во время тура проверяется, что сданный файл содержит корректное описание t блоков (не обязательно формирующих правильное изображение). Проверка правильности ответа осуществляется в режиме offline (результат виден после окончания тура).

Примеры

| Входные данные | Результат |
|----------------|-----------|
| 3 | 1 |
| 4 | 2 2 |
| 0 0 1 0 | 2 |
| 0 0 1 0 | 2 2 |
| 1 1 1 1 | 1 1 |
| 0 0 1 0 | 2 |
| 4 | 2 2 |
| 0 1 1 0 | 0 0 |
| 1 1 0 1 | |
| 1 0 1 1 | |
| 0 1 1 0 | |
| 3 | |
| 1 1 0 | |
| 1 0 1 | |
| 0 1 1 | |

Задача D. Закупка стройматериалов

Василий делает ремонт дома и хочет потратить на закупку стройматериалов как можно меньше денег.

Всего Василию нужно закупить m различных товаров. В его городе работает $n \leq 10$ магазинов стройматериалов. В каждом магазине продается каждый из необходимых товаров, но цена на них может быть различной в различных магазинах. Также для каждого магазина известна стоимость доставки, стоимость которой не зависит от количества купленных товаров.

Помогите Василию сэкономить деньги и для каждого товара определить, в каком магазине его необходимо купить.

В первой строке входных данных записано число t — количество различных наборов товаров и магазинов. Каждое описание набора описывается двумя числами n и m — количество магазинов и товаров, которые необходимо купить. В следующей строке записано n чисел — стоимость доставки для каждого из магазинов. В следующих m строках записано по n чисел, в i -ой из этих строк записана стоимость покупки i -го товара в каждом из магазинов.

В качестве ответа необходимо сформировать t блоков. Каждый блок в первой строке должен содержать суммарную стоимость покупки товаров с доставкой, а во второй строке m чисел от 1 до n — номера магазинов для покупки каждого из товаров. Если правильных ответов несколько — можно выбрать любой из них.

В первом тесте $t = 3$. Оценка за этот тест: 30 баллов. Каждый правильно сформированный блок оценивается в 10 баллов. Проверка осуществляется в режиме online (результат виден сразу).

Во втором тесте $t = 70$. Оценка за этот тест: 70 баллов. Каждый правильно сформированный блок оценивается в 1 балл. Во время тура проверяется, что сданный файл содержит корректное описание t блоков (не обязательно оптимальных по цене). Проверка правильности ответа осуществляется в режиме offline (результат виден после окончания тура).

Примеры

| Входные данные | Результат |
|----------------|-----------|
| 2 | 16 |
| 3 3 | 1 3 2 |
| 5 1 3 | 22 |
| 1 7 7 | 2 2 |
| 5 8 1 | |
| 10 5 8 | |
| 2 2 | |
| 10 10 | |
| 3 4 | |
| 10 8 | |

Задача Е. Черные ящики

Василий нашел несколько черных ящиков, поддерживающих операции «добавить элемент» и «извлечь элемент». Он предположил, что каждый из ящиков является либо стеком, либо очередью и провел над каждым из ящиков эксперимент.

Операция извлечения элемента из стека извлекает элемент добавленный позже остальных, а в очереди — раньше остальных.

По результатам эксперимента над каждым ящиком определите, может ли он являться стеком, очередью, стеком и очередью одновременно (по результатам эксперимента он может быть как стеком, так и очередью) или чем-то иным.

В первой строке входных данных записано число t — количество различных черных ящиков, над которыми проводился эксперимент. Каждое описание эксперимента состоит из числа n — количества выполненных операций и описания n операций. Операция добавления элемента состоит из слова *push* и числа x — добавляемого элемента. Операция извлечения элемента состоит из слова *pop* и числа x — значения извлеченного элемента.

В качестве ответа необходимо вывести t строк, каждая из которых описывает возможный тип черного ящика, определенный по результатам эксперимента. Если эксперимент соответствует стеку, то необходимо вывести *stack*, очереди — *queue*, очереди и стеку одновременно — *both*, если по результатам эксперимента черный ящик не может быть ни очередью, ни стеком, следует вывести *none*.

В первом тесте $t = 3$. Оценка за этот тест: 30 баллов. Баллы начисляются только в случае, если все типы черных ящиков определены верно. Проверка осуществляется в режиме online (результат виден сразу).

Во втором тесте $t = 70$. Оценка за этот тест: 70 баллов. Баллы начисляются только в случае, если все типы черных ящиков определены верно. Во время тура проверяется, что сданный файл содержит t строк, каждая из которых содержит одно слово из набора *stack*, *queue*, *both* или *none*. Проверка правильности ответа осуществляется в режиме offline (результат виден после окончания тура).

Примеры

| Входные данные | Результат |
|---|--------------------------------|
| 4 6 push 28 pop 28 push 16 push 68 pop 68 pop 16 8 push 30 push 28 pop 30 push 52 push 79 pop 28 pop 52 pop 79 6 push 46 push 52 push 31 pop 31 pop 19 pop 46 4 push 97 pop 97 push 25 pop 25 | stack queue none both |