

Условия задач 9-11 класс

Решением для задач является консольное приложение!

Данные можно считывать с консоли или из текстового файла `input.txt`. Результат можно выводить на консоль или в текстовый файл `output.txt`.

Каждая строка во входных данных завершается символом конца строки. Последняя строка во входном файле пустая.

Система оценивания

Если не пройдены все тесты из условия, то баллы за задачу **не начисляются**.

Баллы начисляются за каждый пройденный тест (кроме тестов из условия). Если все тесты для данной подзадачи пройдены, то начисляется указанное для этой подзадачи количество баллов. Если тесты пройдены не все, то начисляется меньшее количество баллов. Решение получает максимальное количество баллов в случае, когда **все** тесты будут пройдены.

Максимальное время тестирования всех задач на любом тесте – 1 секунда (если в условии не оговорено иное).

1. «Выставка цветов» (100 баллов)

Однажды ребята из одного класса посетили выставку цветов, где были представлены самые замысловатые букеты. Каждый ребенок запомнил точное количество цветов в одном букете. Несколько ребят могли запомнить количество цветов в одном и том же букете. Какое минимальное количество цветов могло использоваться для составления всех букетов?

Входные данные:

В первой строке записано одно натуральное число N – количество ребят, которые посещали выставку ($1 \leq N \leq 10^4$).

В каждой из следующих N строк записано одно натуральное число a_i – количество цветов в букете, которое запомнил один из учеников ($1 \leq a_i \leq 10^8$).

Выходные данные:

Одно целое число – минимальное количество цветов, которое могло использоваться для составления всех букетов.

Пример:

Входные данные	Выходные данные
3 5 3 7	15
5 3 11 3 11 7	21

Примечание:

Обратите внимание, что в данной задаче необходимо использовать **64-битный тип** данных, например `long long` в C++, `int64` в Free Pascal, `long` в Java

Описание подзадач:

Подзадача	Баллы	Дополнительные ограничения
1	30	$N=10$
2	70	$(1 \leq N \leq 10^4)$.

2. «Максимальная нагрузка на сервер» (100 баллов)

В олимпиаде по информатике хотят принять участие N человек. Для участия в олимпиаде необходимо выбрать ровно 2 часа непрерывного времени в интервале с 8:00 до 20:00. Каждый участник выбрал наиболее удобное время старта с точностью до минуты и сообщил организаторам олимпиады. Некоторые участники выбрали время начала после 18:00. Тогда на написание олимпиады у них меньше двух часов, т.к. олимпиада закончится в 20:00.

Организаторов олимпиады интересует, какое максимальное количество участников будет писать олимпиаду одновременно.

Входные данные:

В первой строке записано натуральное число N – количество участников олимпиады ($1 \leq N \leq 10^5$).

В каждой из следующих $2N$ строк последовательно записаны натуральные числа – часы h и минуты m времени старта очередного участника, дополненные при необходимости ведущими нулями до двух цифр ($8 \leq h \leq 19, 0 \leq m \leq 59$): во второй и третьей строках h_1 и m_1 – время старта 1-го участника; в четвертой и пятой строках h_2 и m_2 – время старта 2-го участника и т.д. (смотрите *Пример*).

Выходные данные:

В первой и единственной строке одно натуральное число – максимальное количество участников.

Пример:

Входные данные	Выходные данные
4	2
08	
00	
16	
15	
09	
00	
15	
20	

Описание подзадач:

Подзадача	Баллы	Дополнительные ограничения
1	30	$2 \leq N \leq 100$; время начало олимпиады не содержит минут, т.е. это время вида 8:00, 12:00 и т.д.
2	60	$1 \leq N \leq 10^4$
3	10	$1 \leq N \leq 10^5$

3. «Привередливые пассажиры» (100 баллов)

Однажды Игорь приехал в некоторый маленький город на электричке. В этом городе существует только один маршрут общественного транспорта. А вот с видами транспорта здесь существует чуть больше разнообразия. По этому маршруту ходят и автобусы вместимостью A человек, и маршрутные такси вместимостью B человек. После того как жители и гости города приезжают в город на электричке на вокзал, они выстраиваются на остановке в длинную очередь, чтобы уехать в город на автобусе или маршрутке.

Когда подходит автобус или маршрутка, то пассажиры последовательно занимают места от передних к задним. Как только автобус или маршрутка заполняются полностью, то они увозят пассажиров в город. К электричке всегда первым подается автобус, а потом маршрутка, далее они чередуются.

На остановке могут присутствовать «привередливые» пассажиры, которые хотят сидеть только в первой половине автобуса или маршрутки. В первой половине места имеют номера от 1 до $A / 2$ или от 1 до $B / 2$ (здесь “/” означает целочисленное деление). Если же им достается место во второй половине, то они просто пропускают всех стоящих далее по очереди вперед, а сами остаются ждать следующего вида транспорта. При этом в следующий поданный автобус или маршрутку они садятся первыми. Если маршрутка (автобус) не может быть заполнена (заполнен) полностью, то уезжает неполной (неполным).

Игорь знает, что он N -й в очереди на остановке. А также он знает, какие пассажиры перед ним «привередливые». Он хочет узнать, на каком по счету транспортном средстве сможет уехать в город.

Входные данные:

В первой строке записано натуральное число A – вместимость автобуса ($2 \leq A \leq 100$).

В второй строке записано натуральное число B – вместимость маршрутки ($2 \leq B \leq 100$).

В третьей строке записано натуральное число N – номер Игоря в очереди ($1 \leq N \leq 10^7$).

В четвертой строке записано натуральное число K – количество привередливых пассажиров ($1 \leq K \leq 10^7$).

В последующих K строках записаны номера привередливых пассажиров (по одному номеру в каждой строке) в том порядке, как они стоят в очереди.

Выходные данные:

Одно целое число – на каком по счету транспортном средстве Игорь сможет уехать в город.

Пример:

Входные данные	Выходные данные
5	3
2	
10	
2	
3	
9	

Пояснение к примеру

В первый автобус сядут 1, 2, 4, 5, 6 пассажиры (привередливый 3 пассажир садиться в него не будет).

В следующую маршрутку сядут 3, 7 пассажиры.

В следующем автобусе поедет Игорь (он зайдет в автобус после 8 и 9 пассажиров). Это третье транспортное средство по счету.

Описание подзадач:

Подзадача	Баллы	Дополнительные ограничения
1	30	$1 \leq N \leq 100$
2	70	$1 \leq N \leq 10^7$

4. «Вася и компьютерная игра» (100 баллов)

Вася играет в неизвестную компьютерную игру. В этой игре у него есть n персонажей, каждый из которых характеризуется своей силой a_i . Он может выбрать любого персонажа и отправить его захватывать крепости врагов. Всего есть m крепостей, каждая характеризуется двумя числами: d_i – уровень защиты крепости и c_i – количество очков, которое Вася получит в случае захвата крепости. Персонаж с силой a может захватить крепость с защитой d , если $a > d$. Вася хочет для каждого персонажа знать, сколько очков сможет набрать, если отправит этого персонажа захватывать крепости.

Входные данные:

В первой строке вводятся два целых числа n, m – количество персонажей и крепостей ($1 \leq n, m \leq 2 \cdot 10^5$).

Во второй строке вводятся n целых чисел – силы персонажей ($1 \leq a_i \leq 10^{12}$).

В третьей строке вводятся m целых чисел – защиты крепостей ($1 \leq d_i \leq 10^{12}$).

В четвертой строке вводятся m целых чисел – ценности крепостей ($1 \leq c_i \leq 10^{12}$).

Выходные данные:

В единственной выходной строке выведите через пробел n чисел: i -е число должно быть равно максимальному числу очков, которое может набрать i -й персонаж.

Примеры:

Входные данные	Выходные данные
1 10 30 5 10 90 15 80 30 29 95 25 13 1 2 3 4 5 6 7 8 9 10	33
10 10 1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20 1 1 1 1 1 1 1 1 1 1	0 1 2 3 4 5 6 7 8 9

Примечание:

После захвата крепости сила персонажа не уменьшается! То есть персонаж с силой 5 может последовательно захватить две крепости с защитой 3 и 4.

Для каждого персонажа Вася хочет знать ответ **независимо от других персонажей!**

Если вы используете C++ и `std::cin`: В самом начале функции `main` пропишите строчку

```
std::ios_base::sync_with_stdio(false);
std::cin.tie(nullptr);
```

Это ускорит ввод/вывод чисел.

Обратите внимание, что в данной задаче необходимо использовать **64-битный тип** данных, например `long long` в C++, `int64` в Free Pascal, `long` в Java

Описание подзадач

Подзадача	Баллы	Дополнительные ограничения
1	10	$n = 1, m \leq 200, 1 \leq a_i, d_i, c_i \leq 100$
2	10	$n \leq 200, m \leq 200, 1 \leq a_i, d_i, c_i \leq 100$
3	20	$n \leq 200, m \leq 200, 1 \leq a_i, d_i, c_i \leq 10^9$
4	10	$1 \leq a_i, d_i, c_i \leq 100$
5	10	$1 \leq a_i, d_i, c_i \leq 10^9$
6	40	Ограничения из условия

5. «Красная Шапочка и система счисления» (100 баллов)

Красная Шапочка стоит на краю леса, через который ей нужно пройти, чтобы принести пирожки бабушке. Лес можно представить как прямоугольное поле размера $N \times M$, в каждой клетке которого либо растет дерево, либо находится поляна, на которой спит волк. Красная Шапочка уже не раз ходила через этот лес и точно знает, где растут деревья, а где спят волки.

В этот раз она, проходя через лес, посчитала произведение высот всех деревьев на своем пути, но сильно расстроилась – на конце результата было слишком много нулей (как все знают, Красная Шапочка не любит круглые числа, потому что их слишком легко перемножать). В следующей раз она решила пройти по пути, который сделает ее число наименее круглым, причем в системе счисления ее любимого числа q .

Более формально: пусть Красная Шапочка выбрала некоторый маршрут (на котором нет волков, разумеется), посчитаем произведение высот всех деревьев на этом маршруте, переведем его в систему счисления по основанию q , посмотрим на количество нулей на конце числа. Красная Шапочка хочет, чтобы это количество было минимальным!

Красная Шапочка может переходить только между соседними по стороне клетками. Причем, она перемещается только в направлении домика бабушки. Более формально: из клетки (i, j) она может пойти либо в клетку $(i+1, j)$, если $i+1 \leq N$, либо в клетку $(i, j+1)$, если $j+1 \leq M$.

Помогите Красной Шапочке: найдите такой маршрут через лес, который не проходит через логова волков и произведение всех высот деревьев на пути имеет минимальное число нулей на конце в системе счисления с основанием q .

Изначально Красная Шапочка находится в позиции $(1, 1)$ (левая верхняя клетка), а домик бабушки – в позиции (N, M) .

Входные данные

В первой строчке находятся три числа N , M , и q – размерность леса и любимое число Красной Шапочки ($1 \leq N, M \leq 5 \cdot 10^4$, $2 \leq q \leq 10^{12}$).

В следующих N строках вводятся по M чисел: j -е число в i строке задает состояние a_{ij} клетки (i, j) . При этом $a_{ij} = 0$, если в клетке находится поляна. В противном случае a_{ij} задает высоту дерева в клетке.

Гарантируется, что $2 \leq N \cdot M \leq 2 \cdot 10^5$.

Все числа во входных данных представлены в десятичной системе счисления!

Гарантируется, что на начальной и конечной клетке растут деревья!

Выходные данные

Если Красная Шапочка где-то ошиблась и маршрута до домика бабушки не существует, выведите -1 .

Иначе в первой строке выведите минимальное число нулей на конце полученного произведения высот деревьев.

В следующих $N+M-1$ строках выведите по 2 числа – координаты очередной клетки на пути Красной Шапочки (включая начальную и конечную).

Примеры:

Входные данные	Выходные данные
3 3 10 5 2 5 5 10 5 5 5 5	0 1 1 2 1 3 1 3 2 3 3
1 8 6 1 3 2 6 0 4 12 7	-1
1 7 10 1 2 6 4 5 10 15	3 1 1 1 2 1 3 1 4 1 5 1 6 1 7
4 4 5 1 5 15 8 2 3 25 17 0 0 4 3 5 5 125 3	2 1 1 1 2 1 3 1 4 2 4 3 4 4 4

Примечание:

Возможны случаи, когда маршрут от клетки $(1, 1)$ до (N, M) существует, но для этого необходимы «запрещенные» переходы, например $(i, j) \rightarrow (i - 1, j)$. В таком случае считается, что маршрута нет.

Если вы используете C++ и `std::cin`: В самом начале функции `main` пропишите строчку

```
std::ios_base::sync_with_stdio(false);  
std::cin.tie(nullptr);
```

Это ускорит ввод/вывод чисел.

Обратите внимание, что в данной задаче необходимо использовать **64-битный тип** данных, например `long long` в C++, `int64` в Free Pascal, `long` в Java

Описание подзадач

Подзадача	Баллы	Дополнительные ограничения
1	5	$N = 1, M \leq 8, q < 10, a_{ij} < 10$
2	15	$N = 1, M \leq 1000, q = 5$
3	15	$N = 1000, M \leq 1000, q = 5$
4	15	$N = 1000, M \leq 1000, q = 10$
5	50	Ограничения из условия