

Задача А. Проблема сапожника

Имя входного файла: `cobbler.in`
Имя выходного файла: `cobbler.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В некоей воинской части есть сапожник. Рабочий день сапожника длится K минут. Заведующий складом оценивает работу сапожника по количеству починенной обуви, независимо от того, насколько сложный ремонт требовался в каждом случае. Дано n сапог, нуждающихся в починке. Определите, какое максимальное количество из них сапожник сможет починить за один рабочий день.

Формат входных данных

В первой строке вводятся числа K (натуральное, не превышает 1000) и n (натуральное, не превышает 500). Затем идет n чисел — количество минут, которые требуются, чтобы починить i -й сапог (времена — натуральные числа, не превосходят 100).

Формат выходных данных

Выведите одно число — максимальное количество сапог, которые можно починить за один рабочий день.

Примеры

<code>cobbler.in</code>	<code>cobbler.out</code>
10 3 6 2 8	2
3 2 10 20	0

Задача В. Выбор заявок

Имя входного файла: `request.in`
Имя выходного файла: `request.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вы прекрасно знаете, что в ЛКШ.Зима 2017 лекции читают лучшие преподаватели мира. К сожалению, лекционных аудиторий у нас не так уж и много, поэтому каждый преподаватель составил список лекций, которые он хочет прочитать ЛКШатам. Чтобы ЛКШата, утром идя на завтрак, увидели расписание лекций, необходимо его составить прямо сейчас. И без вас нам здесь не справиться.

У нас есть список заявок от преподавателей на лекции для одной из аудиторий. Каждая заявка представлена в виде временного интервала $[s_i, f_i)$ — время начала и конца лекции. Лекция считается открытым интервалом, то есть какая-то лекция может начаться в момент окончания другой, без перерыва. Необходимо выбрать из этих заявок такое подмножество, чтобы суммарно выполнить максимальное количество заявок. Учтите, что одновременно в лекционной аудитории, конечно же, может читаться лишь одна лекция.

Формат входных данных

В первой строке вводится натуральное число N , не более 1000 — общее количество заявок на лекции. Затем вводится N строк с описаниями заявок — по два числа в каждом s_i и f_i . Гарантируется, что $s_i < f_i$. Время начала и окончания лекции — натуральные числа, не превышают 1440 (в минутах с начала суток).

Формат выходных данных

Выведите одно число — максимальное количество заявок, которые можно выполнить.

Примеры

<code>request.in</code>	<code>request.out</code>
1 5 10	1
3 1 5 2 3 3 4	2

Задача С. Планирование заданий

Имя входного файла: `schedule.in`
Имя выходного файла: `schedule.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Имеется некоторое множество заданий и один исполнитель. На выполнение одного задания уходит единица времени. Задания можно выполнять начиная с момента времени 0. У каждого задания есть две характеристики: d_i и w_i . Если задание не было выполнено к моменту времени d_i , взимается штраф в размере w_i . Требуется минимизировать суммарный штраф.

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество заданий ($1 \leq n \leq 1000$). Следующие n строк содержат по два натуральных числа, разделенных пробелом — d_i и w_i ($0 \leq d_i, w_i \leq 1000$).

Формат выходных данных

Выведите одно число — минимальный суммарный штраф.

Примеры

<code>schedule.in</code>	<code>schedule.out</code>
2 1 1 1 2	1
1 0 5	5

Задача D. Роботы

Имя входного файла: `robot.in`
Имя выходного файла: `robot.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дизайнер Вадим как никогда близок к завершению своего грандиозного проекта — оформлению главной площади города!

Главная площадь представляет собой прямоугольник шириной W и длиной H метров. Вадим задумал покрасить некоторые из квадратов 1×1 метр в зелёный цвет. Для осуществления этого проекта уже заказано N специальных роботов. Роботы умеют перемещаться в соседние по горизонтали, вертикали и диагонали квадраты, и красят все квадраты, в которых побывали. Роботы очень маленькие, поэтому они не сталкиваются друг с другом, даже если красят один квадрат.

Уже почти всё готово к началу работ и Вадим уже написал программу для роботов, состоящую из M команд. Но перед её запуском он хочет убедиться, что в результате получит именно то изображение, которое задумал. Помогите Вадиму!

Формат входных данных

В первой строке записаны 4 числа: W , H , N , M . В каждой из последующих N строк находятся по два целых числа x_i и y_i ($0 \leq x_i < W$, $0 \leq y_i < H$) — начальные координаты роботов. В каждой из следующих M строк записана одна из следующих команд для одного из роботов:

- `left i k` — i -й робот должен k раз повернуть налево на 45° ($0 \leq i < N$, $0 < k \leq 100$)
- `right i k` — i -й робот должен k раз повернуть направо на 45° ($0 \leq i < N$, $0 < k \leq 100$)
- `forward i d` — i -й робот должен сделать вперёд d шагов ($0 \leq i < N$, $d > 0$)

Все роботы изначально смотрят направо. Гарантируется, что при выполнении программы ни один робот не выйдет за пределы площади.

Формат выходных данных

В каждой из H строк выведите по W символов — "#", если в соответствующей клетке побывал хотя бы один из роботов, и "." иначе. Клетке с координатами $(0, 0)$ соответствует левый нижний угол.

Примеры

robot.in	robot.out
10 7 3 10
3 4	...#..#...
6 5	...#..#...
8 2
left 0 2	.#.....#.
forward 0 1	..#####..
right 1 2
forward 1 1	
right 2 3	
forward 2 1	
right 2 1	
forward 2 5	
left 2 31	
forward 2 1	

Замечание

Для решения этой задачи необходимо использовать классы.

Задача Е. Алиса и яблоки

Имя входного файла: `apples.in`
Имя выходного файла: `apples.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Алисе в стране чудес попались n волшебных яблок. Про каждое яблоко известно, что после того, как его съешь, твой рост сначала уменьшится на a_i сантиметров, а потом увеличится на b_i сантиметров. Алиса очень голодная и хочет съесть все n яблок, но боится, что в какой-то момент ее рост станет равным нулю или еще меньше, и она пропадет совсем. Помогите ей узнать, можно ли съесть яблоки в таком порядке, чтобы в любой момент времени рост Алисы был больше нуля.

Формат входных данных

В первой строке вводятся натуральные числа n и s ($1 \leq n \leq 1000$, $1 \leq s \leq 100\,000$) — число яблок и начальный рост Алисы. В следующих n строках вводятся пары натуральных чисел a_i , b_i , не больших 1000.

Формат выходных данных

Если яблоки съесть нельзя, выведите число -1 . Иначе выведите n чисел — номера яблок, в том порядке, в котором их нужно есть.

Примеры

<code>apples.in</code>	<code>apples.out</code>
3 5 2 3 10 5 5 10	1 3 2
3 5 2 3 10 5 5 6	-1

Задача F. Распечатка условий

Имя входного файла: `printing.in`
Имя выходного файла: `printing.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Популярность окружной олимпиады по информатике растёт год от года. При этом организаторы должны заранее распечатать как условия задач, так и другие материалы олимпиады (анкеты, памятки и т.п.). В этом году они оценили объём печатной продукции в N листов.

Фирма, готовая размножить печатные материалы, предлагает следующие финансовые условия. Один лист она печатает за A_1 рублей, 10 листов — за A_2 рублей, 100 листов — за A_3 рублей, 1000 листов — за A_4 рублей, 10000 листов — за A_5 рублей, 100000 листов — за A_6 рублей, 1000000 листов — за A_7 рублей. При этом не гарантируется, что один лист в более крупном заказе обойдется дешевле, чем в более мелком. И даже может оказаться, что для любой партии будет выгодно воспользоваться тарифом для одного листа.

Печать конкретного заказа производится или путем комбинации нескольких тарифов, или путем заказа более крупной партии. Например, 980 листов можно распечатать, заказав печать 9 партий по 100 листов плюс 8 партий по 10 листов, сделав 98 заказов по 10 листов, 980 заказов по 1 листу или заказав печать 1000 (или даже 10000 и более) листов, если это окажется выгоднее. Требуется по заданному объёму заказа в листах N определить минимальную сумму денег в рублях, которой будет достаточно для выполнения заказа.

Формат входных данных

На вход программе сначала подается число N ($1 \leq N \leq 2 \times 10^9$) — количество листов в заказе. В следующих 7 строках ввода находятся натуральные числа $A_1, A_2, A_3, A_4, A_5, A_6, A_7$ соответственно ($1 \leq A_i \leq 10^6$).

Формат выходных данных

Выведите одно число — минимальную сумму денег в рублях, которая нужна для выполнения заказа. Гарантируется, что правильный ответ не будет превышать 2×10^9 .

Примеры

<code>printing.in</code>	<code>printing.out</code>
980 1 9 90 900 1000 10000 10000	882
980 1 10 100 1000 900 10000 10000	900