

Задача A. Oil

Имя входного файла: `oil.in`
Имя выходного файла: `oil.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Вам задан вертикальный разрез возможного месторождения нефти, представленная в виде прямоугольника $h \times w$. Для каждого вертикального столбца прямоугольника требуется подсчитать стоимость скважины, которую необходимо пробурить, чтобы добраться до нефти. Стоимость скважины зависит от глубины залегания нефти, а также от прочности породы — скважина через мягкую породу стоит 1, скважина через прочную — 3. Если в данном столбце нефти нет, скважину бурить не надо.

Формат входного файла

Во входном файле заданы два целых числа h и w ($1 \leq h, w \leq 40$) — высота и ширина карты.

Далее задана карта в виде h строк по w символов в каждой строке. Наличие нефти обозначается символом 'X', прочная порода — символом 'H', мягкая — символом 'S'.

Формат выходного файла

Для каждого из w столбцов выведите одно число — стоимость строительства скважины в этом столбце или N , если скважину бурить не надо.

Примеры

<code>oil.in</code>	<code>oil.out</code>
<code>1 1</code> <code>S</code>	<code>N</code>
<code>3 5</code> <code>XHXSS</code> <code>XSHSX</code> <code>XXHSS</code>	<code>0 4 0 N 1</code>

Задача В. Sale

Имя входного файла: `sale.in`
Имя выходного файла: `sale.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Во время рекламной акции в одном из крупных магазинов были объявлены скидки тем, кто купил товаров на сумму, большую определённой величины, например:

- при покупке на 2000 или более рублей скидка 10%, но
- при покупке на 5000 или более рублей скидка 15%, но
- при покупке на 10000 или более рублей скидка 20%.

Очевидно, что при такой схеме скидок покупка на некоторую сумму будет невыгодна: например, если вы берёте товара на 1850 рублей, Вы можете докупить товара ещё на 150 рублей и получить 10-процентную скидку, то есть заплатить 1800 рублей за тот же товар, пусть даже товар на 150 рублей вам бесполезен. Считается, что в магазине есть товары по любой наперёд заданной цене.

Ваша задача — по заданной схеме скидок найти все «невыгодные» суммы. Считается, что при применении скидки итоговая сумма округляется до ближайшей целой копейки, половина копейки от цены отбрасывается (то есть при 50-процентной скидке и изначальной сумме 299 рублей 95 копеек покупатель должен заплатить 149 рублей 97 копеек).

Формат входного файла

В первой строке входного файла задано целое число N — количество уровней скидок. В последующих N строках заданы сами скидки парами чисел $d_i p_i$, где целое число d_i ($5 \leq d_i \leq 60$) — процент скидки, а заданное с двумя знаками после десятичной точки число p_i ($0 \leq p_i \leq 10^4$) — сумма покупки в рублях, начиная с которой скидка действует. При этом, если $d_i > d_j$, то $p_i > p_j$, и ни для каких двух различных строк i и j не выполняется равенство $d_i = d_j$.

Формат выходного файла

Выведите одну строку с «невыгодными» ценами, заданными в виде наименьшего возможного количества непересекающихся интервалов, отсортированных по возрастанию, по аналогии с примером.

Пример

sale.in	
3	
10	400.00
15	1000.00
20	2000.00
sale.out	
360.00-399.99, 944.44-999.99, 1882.35-1999.99	

Задача C. Schedule

Имя входного файла: `schedule.in`
Имя выходного файла: `schedule.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Студент Вася был так занят работой в различных институтах, что пропустил почти всю зачётную сессию и вынужден досдавать зачёты во время зачётов у других групп в последний день. Вам задано расписание зачётов по интересующим Васю предметам. При этом для получения зачёта Вася должен присутствовать на зачёте целиком. Требуется выяснить, сможет ли Вася сдать все оставшиеся зачёты, или же расписания каких-то двух зачётов «конфликтуют».

Формат входного файла

В первой строке входного файла задано целое число N ($1 \leq N \leq 100$) — количество зачётов, оставшихся несданными. Далее следуют N строк — расписание зачётов в формате `hh:mm-hh:mm` (24-часовой формат времени, время окончания зачёта строго больше времени его начала).

Формат выходного файла

Выведите `'Yes'`, если Вася сможет сдать все оставшиеся зачёты за один день, и `'No'` в противном случае.

Примеры

<code>schedule.in</code>	<code>schedule.out</code>
3 09:10-09:55 14:00-17:00 09:55-10:40	Yes
2 11:00-12:00 10:00-13:00	No

Задача D. Segments

Имя входного файла: `segments.in`
Имя выходного файла: `segments.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Заданы N горизонтальных отрезков, никакие два из которых не принадлежат одной прямой. Требуется отметить по одной точке на конце каждого из отрезков так, чтобы выпуклая оболочка отмеченных точек имела максимальную площадь.

Формат входного файла

В первой строке входного файла задано целое число N ($3 \leq N \leq 100$). В последующих N строках заданы отрезки тремя целыми числами x_1, x_2, y , по модулю не превосходящими 100 — соответственно x -координатами левого и правого конца отрезка и y -координатой всех точек отрезка.

Формат выходного файла

Выведите одно число — максимальную площадь соответствующей выпуклой оболочки с точностью до 0.01.

Пример

<code>segments.in</code>	<code>segments.out</code>
5 1 6 1 2 3 2 1 6 4 4 5 5 0 8 6	19.5

Задача E. Session

Имя входного файла: `session.in`
Имя выходного файла: `session.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Переэкзаменовка состоит из n тестов. Результат r_k каждого из тестов измеряется в процентах, а финальная оценка строится как сумма по всем тестам произведений $p_k \cdot (r_k/100)$, где p_k — «вес» данного теста, также задаваемый в процентах. При этом сумма всех n чисел p_k равна 100.

Также известно время проведения каждого теста, при этом между любыми двумя тестами проходит чётное количество часов. Для подготовки к переэкзаменовке студент может пойти в библиотеку, заплатив 1 тугрик за посещение. При этом, если студент сдаёт тест сразу после подготовки, он получает 100%, на два часа позже — 99% и так далее (оценка уменьшается на один процент за каждые два часа).

По заданному расписанию тестов и значениям p_k вычислить, какую максимальную оценку студент может получить, потратив не более m тугриков.

Формат входного файла

Во входном файле в одну строку заданы несколько целых чисел. Первое число n ($1 \leq n \leq 36$) — количество тестов, второе — m ($1 \leq m \leq 36$) — количество имеющихся у студента тугриков. Далее идут n пар целых положительных чисел t_i, p_i , где t_i — время i -ого теста ($0 < t_1 < \dots < t_n < 2300$) и p_i — вклад данного теста в итоговую оценку в процентах.

Формат выходного файла

Выведите одно число — максимальную оценку, которую студент может получить на переэкзаменовке, с точностью до 0.01.

Пример

<code>session.in</code>	<code>session.out</code>
4 2 100 10 700 10 1540 35 1700 45	80.00
4 1 30 10 60 40 80 20 100 30	82.00

Задача F. Turtle

Имя входного файла: `turtle.in`
Имя выходного файла: `turtle.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Черепаша движется по бесконечному листу клетчатой бумаги из клетки $(0,0)$ по следующим правилам:

- Первоначально черепаха смотрит в направлении положительной полуоси оси x .
- За каждый ход черепаха продвигается в текущем направлении на 1.
- Если клетка слева от черепахи по направлению только что завершившегося перехода свободна, она сразу после продвижения поворачивается налево (на 90 градусов против часовой стрелки), в противном случае направление остаётся прежним.

Пронумеруем клетки в порядке их обхода черепахой. Получим спираль типа

```
16 15 14 13 12
17  4  3  2 11
18  5  0  1 10
19  6  7  8  9
20 21 22 23 24
```

Вам задан прямоугольник на размеченной таким образом плоскости. Какое минимальное число будет записано в этом прямоугольнике?

Формат входного файла

В первой строке входного файла задано количество тестовых примеров $T \leq 10^4$. Каждый тестовый пример задан в отдельной строке и представляет собой четыре целых числа $x_1 y_1 x_2 y_2$, по модулю не превосходящих 10^6 — координаты соответственно левой нижней и правой верхней клеток прямоугольника.

Формат выходного файла

Для каждого тестового примера выведите наименьшее число, которое записано в клетках заданного прямоугольника.

Пример

<code>turtle.in</code>	<code>turtle.out</code>
2 -1 -1 1 1 1 3 4 5	0 30