Задача A. Dice and Map

Имя входного файла:	`dicemap.in`
Имя выходного файла:	`dicemap.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 mebibytes

Проблемы интеллектуальной собственности и защиты информации в средние века решались следующим образом — автор открытия зашифровывал сообщение о нём известным только ему ключом, а для подтверждения предъявлял ключ. В одной из средневековых библиотек, принадлежавшей иезуитам, историки нашли карту, сопровождаемую странными надписями. Изображённая в углу игральная кость помогла учёным понять суть шифра — на карте были зашифрованы сведения о недавно открытых землях. Перекатывание кости в соответствии с инструкциями указывало не только на место на карте, но и на тип расположенного на ней объекта (типы объектов занумерованы числами от 1 до 6 включительно).

Изначально игральная кость установлена в некоторой точке карты так, что единица находится сверху, шестёрка снизу, двойка «смотрит» на юг, тройка — на запад, пятёрка — на север и четвёрка — на восток.

Последовательность ходов записывается в виде строки, в которой команды следуют друг за другом. Существуют следующие команды:

- '`-`' меняет направления ходов на противоположные (запад вместо востока и юг вместо севера). Повторение этой команды не отменяет её (для отмены используется команда '`+`';
- '`+`' возвращает изначальные направления ходов (на восток и на север);
- '`X`' обозначает перекатывание кости на восток (например, если перед этим кость была в начальном положении, то сверху окажется тройка, а кость сместится на одну клетку восточнее) или на запад, если «действует» команда '`-`';
- '`Y`' обозначает перекатывание кости на север (например, если перед этим кость была в начальном положении, то сверху окажется двойка, а кость сместится на одну клетку севернее) или на юг, если «действует» команда '`-`';
- "`1`"—"`9999999`" может появляться только перед символами '`X`' и '`Y`' и обозначает, что идущая после числа команда повторяется соответствующее количество раз;
- "`.`" обозначает, что последовательность ходов закончена и можно выдать результат (число точек на грани кубика, оказавшейся сверху).

По заданной последовательности ходов вычислите точку, на которую эта последовательность указывает, и тип соответствующего ей объекта.

Формат входного файла

Во входном файле задана одна строка, состоящая из цифр и символов '`X`', '`Y`', '`+`', '`-`' и заканчивающаяся символом '`.`' — последовательность ходов. Длина последовательности не превосходит 1000 символов.

Формат выходного файла

Выведите три числа — координаты точки, на которую указывает заданная последовательность (считая, что изначально игральная кость ставится в точку (0,0), при перекатывании на север координата Y увеличивается на 1, при перекатывании на восток координата X увеличивается на 1 (при перекатывании на юг и на запад соответствующаяя координата уменьшается на 1), а также количество точек на верхней грани игральной кости.

Примеры

dicemap.in	dicemap.out
6X-YXY+Y.	5 -1 3
YX-YX.	0 0 2
11X11Y-11X11Y.	0 0 4

Задача B. Fast dialing

Имя входного файла:	`fastdial.in`
Имя выходного файла:	`fastdial.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 mebibytes

Клавиатура у некоторого сотового телефона имеет указанную на рисунке форму.

```
1  2  3

4  5  6

7  8  9

*  0  #
```

При наборе номера владелец телефона берёт аппарат, и, используя лишь большие пальцы двух рук, последовательно нажимает соответствующие цифрам номера кнопки.

Ваша задача — по заданному номеру вычислить минимально возможную суммарную длину путей, по которым будут перемещаться пальцы.

При этом полагаем, что размеры кнопок настолько велики, а пальцы настолько малы, что можно беспрепятственно перемещать палец с одной кнопки на другую по прямой линии. Палец при нажатии на кнопку должен располагаться точно по центру этой кнопки. Расстояния между центрами соседних кнопок по вертикали и горизонтали равны 1. Первоначально пальцы можно установить на произвольные кнопки.

Формат входного файла

В первой и единственной строке входного файла задан набираемый номер — последовательность из не более, чем 10^5 цифр от 0 до 9.

Формат выходного файла

Выведите требуемую минимальную сумму длин путей, по которым перемещались пальцы при наборе заданного номера с точностью до 10^{-4}.

Примеры

fastdial.in	fastdial.out
16904	3.414214
222336555330	4.000000

Задача C. Game

Имя входного файла:	`game.in`
Имя выходного файла:	`game.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 mebibytes

Два демона Максвелла играют в следующую игру. В начале игры имеется некоторое количество элементарных частиц. Каждая частица принадлежит к одному из четырёх типов: A, B, C или D. Во время своего хода каждый демон может выбрать некоторые частицы так, чтобы они вступили ровно в одну реакцию и при этом полностью взаимоуничтожились. Ходы делаются по очереди до тех пор, пока не сложится ситуация, в которой выбрать непустое множество частиц подобным образом невозможно. В этом случае демон, который первым не сможет сделать ход, проигрывает.

Возможны следующие реакции (для каждой реакции указаны частицы, в ней участвующие; результатом каждой реакции является взаимоуничтожение участвующих в ней частиц).

- $A + A + B + D + D$
- $A + B + C + D$
- $C + C + D$
- $B + B + B$
- $A + D$

То есть каждый ход должен представлять собой одну из вышеупомянутых реакций.

По заданному стартовому набору частиц определите, кто выиграет при оптимальной игре: демон, начинающий игру или же демон, делающий первый ход вторым.

Формат входного файла

В первой строке входного файла содержится целое число n — количество тестовых примеров ($1 \leqslant n < 100$). Каждый тестовый пример расположен на одной строке и состоит из четырёх целых чисел n_a, n_b, n_c, n_d — количество частиц соответственно типа A, типа B, типа C и типа D ($0 \leqslant A, B, C, D \leqslant 30$).

Формат выходного файла

Для каждого тестового примера выведите в отдельной строке “`First`”, если выиграет первый демон, и “`Second`” в противном случае.

Пример

game.in	game.out
5 0 3 0 3 2 5 0 4 7 7 7 7 9 9 7 8 10 10 10 10	First First First First Second

Задача D. Loyality

Имя входного файла:	`loyal.in`
Имя выходного файла:	`loyal.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 mebibytes

Несколько кораблей отправились в дальнюю экспедицию. В какой-то момент кораблям пришлось проходить через узкий пролив, так что командующий экспедицией решил выстроить корабли в кильватерную колонну (то есть так, чтобы они шли друг за другом).

Экспедиция продолжается уже довольно долго, так что некоторые участники экспедиции стали выражать недовольство действиями командующего. Таких недовольных набралось уже значительное количество. Командующему удалось узнать детали зревшего заговора. Заговорщики планировали действовать очень осторожно: если количество лояльных командующему участников экспедиции на некотором корабле N меньше, чем суммарное количество недовольных на нём и на двух соседних с ним кораблях (или одном, если корабль N расположен в начале или в конце колонны), то недовольные высаживаются на корабль N и захватывают его. Кроме того, если на корабле нет ни одного лояльного командующему участника экспедиции, корабль также захватывается недовольными.

Зная настроения каждого участника экспедиции, капитан хочет перераспределить участников по кораблям таким образом, чтобы ни один корабль не был захвачен недовольными (и на каждом корабле был как минимум один участник экспедиции). При каком максимальном количестве недовольных это можно сделать?

Формат входного файла

В первой строке входного файла задано одно целое число — количество тестовых примеров T ($1 \leqslant T \leqslant 15$).

Каждый тестовый пример состоит из одной строки, содержащей два числа n и k ($1 \leqslant n \leqslant 15$, $n \leqslant k \leqslant 40$) — соответственно количество кораблей и общее количество участников экспедиции.

Формат выходного файла

Для каждого тестового примера в отдельной строке выведите одно число — максимальное количество нелояльных участников экспедиции, которых капитан ещё сможет распределить по кораблям так, чтобы ни один корабль не был захвачен.

Пример

loyal.in	loyal.out
3 1 3 3 4 3 16	1 1 5

Задача E. Mouse and Windows

Имя входного файла:	`mouse.in`
Имя выходного файла:	`mouse.out`
Ограничение по времени:	7 seconds
Ограничение по памяти:	256 Mebibytes

Задан экран размером $C \times R$, состоящий из R строк по C пикселей каждая. На экране расположены n окон. Каждое из окон представляет собой прямоугольник из пикселов со сторонами, параллельными краям экрана. Окно задаётся своей верхней левой и нижней правой точками (x_l, y_t) и (x_r, y_b) соответственно, при этом $x_r > x_l$ и $y_b > y_t$ (тем самым все окна невырожденные).Окно включает в себя все точки как внутри окна, так и непосредственно на границе. Если два окна перекрываются, то изначально сверху находится то окно, которое было задано во входном файле позже.

Пользователь может переместить указатель мыши в любую точку экрана (x, y) и нажать левую кнопку. В этом случае окно, которое было «сверху» в данной точке, попадает «в фокус», при этом оно перерисовывается поверх всех остальных окон.

Для каждго из m нажатий мыши выясните, какое окно окажется «в фокусе» после соответствующего нажатия.

Формат входного файла

В первой строке входного файла задано целое число C, задающее ширину экрана, во второй строке — целое число R, задающее высоту экрана ($1 \leqslant R, C \leqslant 10^4$). В третьей строке задано целое число n — количество окон на экране ($1 \leqslant n \leqslant 5 \cdot 10^4$).

В последующих n строках описаны окна. i-я из этих строк описывает окно с номером i (окна нумеруются с единицы) и состоит из четырёх чисел: x_l, y_t, x_r, y_b — координат соответственно верхнего левого и правого нижнего угла соответствующего окна ($1 \leqslant x_l < x_r \leqslant C,\ 1 \leqslant y_b < y_t \leqslant R$).

Далее следует целое число m, задающее количество нажатий ($1 \leqslant m \leqslant 10^4$). В последующих m строках описаны сами нажатия. Каждое из них задаётся двумя целыми координатами x и y ($1 \leqslant x \leqslant C,\ 1 \leqslant y \leqslant R$).

Формат выходного файла

Для каждого нажатия на кнопку мыши в отдельной строке выведите целое число w — номер окна, которое окажется «в фокусе» после данного нажатия, или 0, если в точке, в которой было осуществлено нажатие, окна не было.

Пример

mouse.in	mouse.out
400 200 3 140 120 360 20 100 100 160 40 20 180 200 80 3 120 40 300 180 300 60	2 0 1

Задача F. Oil stations

Имя входного файла:	`oil.in`
Имя выходного файла:	`oil.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 mebibytes

В столице Байтландии $n + m$ улиц. n улиц проходят с севера на юг, m — с запада на восток. Улицы, проходящие с запада на восток, занумерованы от 1 до m (чем больше номер, тем южнее), улицы, проходящие с севера на юг, занумерованы от 1 до n (чем больше номер, тем восточнее). Перекрёстки в этом городе обозначаются как (x, y), где x — номер улицы, проходящей с запада на восток, а y — номер улицы, проходящей с севера на юг. Вы находитесь на перекрёстке $(1, 1)$ в автомобиле, бензобак которого имеет объём f литров и заправлен полностью.

На преодоление одного квартала (расстояния между соседними перекрёстками) автомобиль тратит 1 литр бензина. На k перекрёстках в городе имеются бензозаправочные станции, при этом цена бензина на каждой станции своя. На станции вы можете дозаправить автомобиль любым объёмом бензина, при условии, что суммарный объём купленного и оставшегося бензина не превосходит объёма бензобака.

Ваша задача — доехать до перекрёстка (m, n), затратив наименьшее количество денег на покупку дополнительного топлива.

Формат входного файла

В первой строке входного файла задано целое число T — количество тестовых примеров ($1 \leqslant T \leqslant 5$).

В первой строке каждого тестового примера заданы четыре целых числа m, n, f и k — соответственно количество улиц, проходящих с запада на восток, количество улиц, проходящих с севера на юг, объём бензобака в литрах и количество заправочных станций ($1 \leqslant m, n \leqslant 100,\ 1 \leqslant f \leqslant 10^5,\ 1 \leqslant k \leqslant 1000$).

В последующих k строках описаны заправочные станции. Каждая станция описывается своими координатами — двумя целыми числами p_i и q_i ($1 \leqslant p_i \leqslant m,\ 1 \leqslant q_i \leqslant n$) и ценой на бензин o_i (число не более, чем с двумя знаками после десятичной точки) ($0 < o_i \leqslant 100$);

Формат выходного файла

Для каждого тестового примера в случае, если доехать от $(1, 1)$ до (m, n) возможно, выведите минимальную сумму, которую при этом придётся потратить на топливо, с точностью до 10^{-2}. Если же топлива не хватит в любом случае, выведите -1.

Пример

oil.in	oil.out
2 6 6 7 2 3 3 0.6 4 2 0.4 12 13 6 2 1 2 2 8 12 4.8	1.20 -1