

## Problem A. Абстракционизм

Input file:            стандартный ввод  
Output file:         стандартный вывод  
Time limit:          2 секунды  
Memory limit:       256 мегабайт

Вася — художник-абстракционист. Он хочет нарисовать новую картину. Планируется, что картина будет иметь форму квадрата, разбитого на несколько квадратиков поменьше — итого  $n \times n$  квадратики, каждый из которых будет иметь свой собственный цвет.

Вася нарисовал эскиз, из которого видно, в какой цвет должен быть покрашен каждый квадратик. Для некоторых из них цвет задан единственным образом, остальные же могут быть покрашены в любой цвет, поскольку эти квадратики не влияют на смысл, который Вася хочет вложить в эту картину.

Есть одна сложность для написания самой картины. Вася хочет выполнить работу в особой технике рисования, которую он назвал *техникой непрерывного мазка*. Эта техника заключается в том, что за один взмах кистью в некоторый цвет закрашивается одна строчка или один столбик из квадратики. Каждая строчка и столбец должны быть покрашены ровно один раз — то есть, в итоге каждый квадратик будет покрыт ровно двумя слоями краски. При этом каждый квадратик будет иметь цвет *верхнего* слоя краски.

Помогите Васе определить последовательность совершения мазков, чтобы претворить его замысел в реальность.

### Input

В первой строке дано число  $n$  — размер картины ( $1 \leq n \leq 3000$ ).

В следующих  $n$  строках содержится по  $n$  символов в каждой — описание эскиза Васи.  $i$ -ый символ в  $j$ -ой строке задает цвет  $i$ -го квадрата  $j$ -ой строки будущей картины. Символами от «a» до «z» задаются цвета квадратики, которые нужно покрасить в один конкретный цвет. Символ «?» означает, что цвет данного квадрата не важен и его можно покрасить в любой цвет.

### Output

Выведите  $2n$  строк, каждая из которых задает мазок в формате:

«h y c» — закрасить строку  $y$  в цвет  $c$ ;

«v x c» — закрасить столбец  $x$  в цвет  $c$ .

Строки и столбцы нумеруются начиная с 1 слева-направо и сверху-вниз. Каждая строка и каждый столбец должны упоминаться ровно один раз.

Гарантируется, что решение всегда существует. Если возможных решений несколько — выведите любое из них. Смотрите примеры для более точного понимания.

## Example

стандартный ввод	стандартный вывод
3	h 1 p
ac?	h 3 q
ab?	v 2 c
?cz	h 2 b
	v 1 a
	v 3 z

## Problem C. Constant Ratio

Input file:            стандартный ввод  
Output file:         стандартный вывод  
Time limit:          1 секунда  
Memory limit:       256 мегабайт

Дано целое число  $N$ . Требуется найти количество способов разложить его на два или более целых слагаемых  $a_i$  так, чтобы отношение  $a_i/a_{i-1}$  не зависело от  $i$  и было целым для всех  $i > 1$ .

### Input

На вход подаётся одно целое число  $n$  ( $1 \leq n \leq 10^5$ ).

### Output

В выходной файл выведите одно число — количество искомых разбиений числа.

### Examples

стандартный ввод	стандартный вывод
1	0
5	2
567	21

### Note

В первом примере не существует ни одного подходящего разложения на слагаемые.

Во втором примере такими разложениями являются:

- 1 1 1 1 1
- 1 4

## Problem D. dir -C

Input file:            стандартный ввод  
Output file:          стандартный вывод  
Time limit:          1 секунда  
Memory limit:        256 мегабайт

Известный берляндский разработчик и предприниматель Линус Гейтс выпустил очередную версию своей проприетарной open-source системы «Winux 10.03.LTS».

В этой системе команда “`dir -C`” выводит список всех файлов текущего каталога в многоколоночном режиме.

Определим, как строится многоколоночный режим с количеством строк вывода  $l$ . Пусть все имена файлов уже отсортированы лексикографически.

- Список имён файлов разбивается на несколько непрерывных блоков так, чтобы все блоки, кроме, возможно, последнего состояли из  $l$  имён файлов, а последний блок состоял из не более чем  $l$  имён файлов. Каждый такой блок выводится в отдельной колонке.
- Ширина  $w_i$  каждой колонки определяется как наибольшая длина имени файла в соответствующем блоке.
- Соседние колонки разделяются столбцом  $1 \times l$ , состоящим из пробелов.
- Таким образом, ширина вывода определяется как  $\sum(w_i + 1) - 1$ , то есть как сумма ширин всех колонок плюс количество колонок минус единица.

Приведём пример многоколоночного вывода:

```
a      accd e t
aba    b    f wtrt
abacaba db  k
```

В вышеприведённом примере ширина вывода равна 19.

Команда “`dir -C`” выбирает наименьшее  $l$  такое, что ширина вывода не превосходит заданой ширины экрана  $w$ .

По заданной информации о длинах имён файлов и ширине экрана вычислите  $l$  — количество строк, выведенных командой “`dir -C`”.

### Input

В первой строке входного файла заданы два целых числа  $n$  и  $w$  — количество файлов в списке и ширина экрана, соответственно ( $1 \leq n \leq 10^5$ ,  $1 \leq w \leq 10^9$ ).

Во второй строке даны  $n$  чисел.  $i$ -ое из них равно длине имени  $i$ -го файла из списка, который уже упорядочен в алфавитном порядке.

Гарантируется, что длина имени любого файла — целое число от 1 до  $w$ .

### Output

В выходной файл выведите одно число — количество строк, которые выведет команда «`dir -C`».

## Examples

стандартный ввод	стандартный вывод
11 20 1 3 7 4 1 2 1 1 1 1 4	3

## Problem F. Кратчайшие циклы

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 2 секунды (5 секунд для Java)  
Memory limit: 256 мегабайт

Дан неориентированный взвешенный граф. Для каждой вершины найдите длину кратчайшего простого цикла, проходящего через данную вершину.

### Input

Первая строка содержит целое число  $n$  — количество вершин ( $1 \leq n \leq 300$ ).

В следующих  $n$  строках находится по  $n$  целых чисел, которые задают матрицу смежности.  $j$ -ое число в  $i$  строке задает число  $a_{ij}$  — вес ребра, соединяющего вершины  $i$  и  $j$ . Если  $a_{ij}$  равно  $-1$ , то это означает, что между  $i$  и  $j$  ребра нет. Матрица симметрична, т.е.  $a_{ij} = a_{ji}$  для любых  $1 \leq i, j \leq n$ . Для всех  $1 \leq i \leq n$   $a_{ii} = 0$ , при этом петель граф не содержит.

Все ребра имеют вес от 1 до  $10^6$ .

### Output

Выведите  $n$  строк, каждая из которых содержит одно целое число.  $i$ -ое из этих чисел должно быть равно длине кратчайшего простого цикла, проходящего через вершину  $i$ . Если через вершину  $i$  не проходит ни одного простого цикла — выводите  $-1$ .

### Examples

стандартный ввод	стандартный вывод
4	11
0 9 1 1	11
9 0 -1 1	-1
1 -1 0 -1	11
1 1 -1 0	

## Problem G. Игра “Сапёр”

Input file:            стандартный ввод  
Output file:         стандартный вывод  
Time limit:           1 секунда  
Memory limit:        256 мегабайт

Это интерактивная задача.

Правила игры “Сапёр” достаточно просты:

Есть поле размера  $16 \times 16$ . На поле расположено некоторое число мин. Некоторые клетки пустые. Цель игры — пометить все мины и открыть пустые клетки. Если открыть пустую клетку, то в ней будет написано число — количество мин в соседних клетках. Две различных клетки являются соседними, если соприкасаются не менее чем одним углом. При открытии клетки, в которой стоит число 0, сразу открываются все её соседние клетки.

Игрок может делать три различных вида кликов.

1. Одинарный клик по непроверенной клетке с целью открыть её. Если в клетке находится мина — игрок проигрывает. Иначе — клетка открывается.
2. Двойной клик по клетке, которая была открыта и содержит в себе число. Если количество меток в соседних клетках совпадает с числом, которое записано в клетке, то все закрытые соседние клетки открываются. Если метки были расставлены неверно, то игрок проигрывает.
3. Правый клик — пометить непроверенную клетку как клетку, в которой содержится мина. Если клетка уже была помечена ранее, то метка снимается.
4. Завершить игру.

Если все пустые клетки открыты, а клетки с минами помечены, игрок побеждает. Иначе — проигрывает.

Вам даны координаты клетки, в которой заведомо нет мины. Гарантируется, что в данной игре не будет возникать ситуаций, в которых неочевиден следующий ход (то есть приходится угадывать), и что выигрыш возможен.

Ваша задача — открыть все пустые клетки и пометить все мины.

### Input

На ввод дано 2 целых числа:  $x, y$  — координаты клетки, в которой гарантированно нет мины.  $1 \leq x, y \leq 16$ .  $x$  — номер строки,  $y$  — номер столбца.

Далее будут содержаться ответы на запросы. Ответ на запрос выглядит следующим образом. Первая строка ответа содержит число  $N$  — число клеток поля, в которых изменилось значение. Далее следует  $N$  строк вида  $x_i, y_i, c_i$ .  $x_i, y_i$  — координаты изменившейся клетки.  $c_i$  — значение, которое появилось в этой клетке.  $1 \leq x_i, y_i \leq 16$ .  $0 \leq c_i \leq 8$ , если клетка пуста.  $c_i = “*”$  если игрок поставил в клетку метку мины.  $c_i = “-”$  если игрок снял из клетки метку мины.

## Output

Запрос на клик выглядит следующим образом. Сначала выведите тип клика. Если это не завершающий клик, то выведите координаты клетки, в которую совершается клик. Не забывайте сбрасывать буфер вывода после вывода каждого запроса. Это делается при помощи функции “`fflush(stdout)`”; в “C++” и “`flush(output)`”; в “Pascal”.

## Example

стандартный ввод	стандартный вывод
1 1	1 1 1
6	
1 1 0	
1 2 0	
2 1 1	
2 3 1	
1 3 0	
2 2 2	3 3 2
1	
3 2 *	3 3 2
1	
3 2 -	3 3 1
1	
3 1 *	2 2 1
1	
3 2 2	3 3 3
1	
3 3 *	4



## Problem I. Interactive Casino

Input file:            стандартный ввод  
Output file:          стандартный вывод  
Time limit:           1 секунда  
Memory limit:        256 мегабайт

Это интерактивная задача.

Игра «Битовая рулетка» проходит по следующим правилам.

- У игрока изначально есть 160 фишек.
- За одну ставку игрок может поставить любое целое положительное количество  $k$  фишек, не большее, чем количество фишек у игрока в настоящий момент.
- Генерируется целое число от 0 до  $2^{20} - 1$ . Если сумма его бит нечётна, то игрок выигрывает (возвращает ставку и получает ещё  $k$  фишек). В противном случае игрок теряет ставку. При этом само число игроку *не показывается*.
- Если фишек у игрока не осталось, он считается проигравшим.
- Если игрок сделал более 200 ставок, он считается проигравшим.
- Если у игрока в какой-то момент времени оказалось 200 или более фишек, он считается выигравшим

На сайте AlgoLeaks Вы прочитали, что каждое последующее число на барабанах вычисляется по формуле  $x_i = 487237 * x_{i-1} + 1011807 \bmod 2^{20}$ . Откуда берётся первое число, на сайте не написано.

Сможете ли Вы гарантированно выиграть?

### Input

Программа жюри выдаёт Вашей программе на вход одно целое число — количество фишек, которое у Вас есть в наличии перед очередной ставкой, или  $-1$  в случае, если Ваша программа должна немедленно завершиться.

### Output

Если полученное на вход число равно  $-1$ , немедленно завершите работу программы (в противном случае корректный код ошибки не гарантируется). Если же на вход получено число  $T > 0$ , выведите свою ставку — целое число в интервале от 1 до  $T$ . Не забудьте очищать буфер вывода командой `flush`.

### Example

стандартный ввод	стандартный вывод
160	5
155	10
165	15
180	20
-1	

## Problem K. Подземные ходы

Input file:            стандартный ввод  
Output file:         стандартный вывод  
Time limit:          6 секунд  
Memory limit:       256 мегабайт

Чешский крот Крóтек живет на двумерной плоскости, расположенной на глубине  $\varepsilon$  под поверхностью земли.

Недавно Крóтек прорыл  $N$  подземных ходов для своего нового дома. Каждый ход представляет собой прямой отрезок, заданный координатами начала и конца. Известно, что эти отрезки не пересекаются и никакие три концевые точки отрезков не лежат на одной прямой.

Теперь Крóтек хочет объединить все ходы в единую сеть, прорыв дополнительные ходы (которые тоже представляют собой отрезки прямой). Новые ходы должны начинаться и заканчиваться в концах уже существующих ходов. Кроме того, новые ходы не должны пересекаться друг с другом, а также пересекать старые ходы.

Крóтек хочет узнать, какое минимальное суммарное расстояние ему придется дополнительно прорыть, чтобы осуществить задуманное.

### Input

В первой строке входного файла дано целое число  $N$  ( $1 \leq N \leq 1000$ ). В последующих  $N$  строках содержатся четверки чисел  $(x_1, y_1, x_2, y_2)$  — координаты выходов для каждого уже прорытого хода. Координаты представляют собой целые числа, не превышающие по модулю  $10^4$ .

### Output

Выведите одно число — суммарную длину дополнительных подземных ходов в итоговой сети (не учитывая ранее прорытые) с абсолютной или относительной погрешностью не хуже  $10^{-6}$ .

### Example

стандартный ввод	стандартный вывод
2 0 0 3 0 0 1 1 2	1.0000000000

## Problem L. Игра со строкой

Input file:            стандартный ввод  
Output file:          стандартный вывод  
Time limit:           2 секунды  
Memory limit:        256 мегабайт

Маленькая Элизабет любит играть со строками. У нее есть строка длины  $n$ , полностью состоящая из букв «а». Она делает со строкой следующие действия:

- Если первая буква в строке «а», то Элизабет дописывает в конец строки «bc», после чего удаляет из строки первые 2 буквы.
- Если первая буква в строке «b», то тогда Элизабет дописывает в конец строки «a», после чего также удаляет из строки первые 2 буквы.
- Если же первая буква в строке «с», то Элизабет дописывает в конец строки «aaa», после чего опять удаляет из строки первые 2 буквы.

Элизабет останавливается после того, как у нее получится строка, состоящая из одной буквы. Например, при  $n = 4$ , для достижения цели ей требуется 6 шагов:

$$aaaa \rightarrow aabc \rightarrow bc bc \rightarrow bca \rightarrow aa \rightarrow bc \rightarrow a$$

Элизабет обнаружила, что для некоторых  $n$  нужно делать слишком много операций, и не понятно, завершится ли процесс вообще. Поэтому она попросила Вас написать для этого программу.

### Input

В единственной строке дано одно целое число  $n$  ( $2 \leq n \leq 10^6$ ).

### Output

Выведите одно число — количество требуемых шагов. Если процесс будет продолжаться бесконечно — выведите  $-1$ .

### Examples

стандартный ввод	стандартный вывод
4	6
3	24

## Problem N. Numbers

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 Mebibytes

На доске в ряд выписаны целые числа от 1 до  $n$ . За одно действие вы можете заменить два произвольных числа  $a$  и  $b$ , стоящих на различных позициях (значения чисел могут совпадать), числом  $|a - b|$ . Можно ли данными операциями сделать так, чтобы на доске осталось единственное целое число  $x$ ?

### Input

Первая строка входных данных содержит число  $T$  — количество тестовых примеров, для которых необходимо решить поставленную задачу ( $1 \leq T \leq 10^5$ ).

В следующих  $T$  строках содержатся по два целых числа  $n_i, x_i$ , ( $1 \leq n_i \leq 10^9, 0 \leq x_i \leq 10^9$ ).

### Output

Выведите  $T$  строк. В  $i$ -ой строке выведите «YES» (без кавычек), если из чисел  $1, 2, \dots, n_i$  приведёнными в условии задачи действиями можно получить единственное число  $x_i$ , и «NO» (без кавычек) иначе.

### Examples

standard input	standard output
5	YES
3 0	NO
3 1	YES
3 2	NO
3 3	NO
3 5	

## Problem O. Ones as the Difference

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 секунды  
Memory limit: 256 мегабайт

Задано число  $n$ . Найдите  $n$ -е число, в десятичной записи которого разность двух любых соседних цифр равна единице.

### Input

Первая строка входного файла содержит одно целое число  $n$ , ( $1 \leq n \leq 512$ ).

### Output

В выходной файл выведите одно число — ответ к задаче.

### Examples

standard input	standard output
1	1
2	2
10	10

## Problem P. Places at the Olympics

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 512 mebibytes

Для сравнения результатов стран на Олимпийских играх используется два способа подсчёта медалей. Первый — «по количеству» учитывает общее количество медалей без учёта их достоинства. Второй — «по качеству» учитывает сначала количество золотых медалей, при их равенстве — количество серебряных и при их равенстве — количество бронзовых.

Вам задано количество золотых, серебряных и бронзовых медалей, завоёванных соответственно Берляндией и Байтландией. Определите, в каком из зачётов Берляндия победила.

### Input

Первая строка входа содержит одно целое число  $T$  ( $1 \leq T \leq 20$ ) — количество тестовых примеров.

Каждый тестовый пример состоит из шести целых чисел от 0 до 500 включительно. Первые три числа задают количество золотых, серебряных и бронзовых медалей соответственно, завоёванных Берляндией, следующие три — количество медалей, завоёванных Байтландией (в аналогичном формате).

### Output

:

Для каждого тестового примера выведите “Win”, если Берляндия победила в обоих зачётах, “Count”, если она выиграла только по количеству, “Quality”, если выиграла только по качеству и “None” во всех оставшихся случаях.

### Example

standard input	standard output
5	Win
20 10 30 20 2 0	Count
15 10 20 15 11 15	Quality
13 6 11 6 21 31	None
14 4 15 14 9 30	None
1 2 3 1 2 3	

## Problem Q. Quaqua the Frog

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Лягушка Кваква решила перебраться в новое болото. Кваква может прыгать на целое число клеток, перепрыгивая при этом не более чем через  $j$  клеток. Оба болота расположены на полосе  $1 \times c$  клеток. Старое болото находится в первой клетке полосы, а новое — в последней.

Некоторые клетки полосы содержат ямы с кислотой, попадать в которые для лягушек небезопасно. Так что Кваква собирается перепрыгивать эти ямы. Дополнительная проблема заключается в том, что сегодня дует сильный ветер в направлении от старого болота к новому, и Кваква не может прыгать против ветра, то есть прыгать лягушка может только в направлении от старого болота к новому.

Выясните, сможет ли Кваква сегодня добраться до нового болота и, если сможет, выведите минимальное количество прыжков, которое ей для этого необходимо.

### Input

Первая строка входа содержит одно целое число  $T$  ( $1 \leq T \leq 35$ ) — количество тестовых примеров.

Далее следуют  $n$  тестовых примеров. Каждый тестовый пример состоит из двух строк. Первая строка содержит два целых числа  $c$  ( $2 \leq c \leq 50$ ) и  $j$  ( $0 \leq j \leq 50$ ) — количество клеток на пути между старым и новым болотами (включая оба болота) и максимальное количество клеток, через которое может перепрыгнуть Кваква, соответственно. Вторая строка содержит  $c$  символов,  $i$ -й из которых равен 'A', если эта клетка содержит яму с кислотой, и '.', если соответствующая клетка безопасна. Гарантируется, что первый и последний символы строки равны '.'.

### Output

Для каждого тестового примера выведите одно целое число — наименьшее количество прыжков, необходимое для того, чтобы Кваква добралась от старого болота до нового. Если же это сделать невозможно, выведите  $-1$ .

### Example

standard input	standard output
4	2
8 3	1
.AA.A.A.	-1
3 50	3
...	
8 1	
...AA...	
10 4	
..AAAA.AA.	

## Problem R. Road in Mountains

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 секунда  
Memory limit: 256 мегабайт

Извилистые горные дороги, иначе называемые серпантинами, могут быть представлены как линии, составленные из дуг окружностей. Большая Берляндская Горная дорога составлена из  $N$  полуокружностей.  $i$ -й участок имеет радиус  $r_i$  и рекомендованная скорость движения на этом участке равна  $v_i$ . Считается, что скорость на каждом участке постоянна и в момент перемещения к следующему участку она мгновенно изменяется на  $v_{i+1}$ .

Берляндское агентство путешествий поручило Вам вычислить среднюю скорость на Большой Берляндской дороге при движении с рекомендованной скоростью на каждом её участке.

### Input

В первой строке входного файла содержится одно число  $N$  — количество участков дороги.  $1 \leq N \leq 300$ . Каждая из  $N$  следующих строк содержит два числа:  $r_i$ ,  $v_i$  — радиус соответствующего участка дороги и рекомендованную скорость ( $0 \leq r_i \leq 10^6$ ,  $0 < v_i \leq 10^6$ ).

### Output

Выведите среднее значение скорости на всём пути следования с абсолютной или относительной погрешностью не хуже  $10^{-6}$ .

### Examples

standard input	standard output
2 1 1 1 1	1.0