

## Задача А. Вечеринка в Нью-Йорке

Имя входного файла: `grid.in`  
Имя выходного файла: `grid.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Скоро у Лизы день рождения! Родители долго думали, чем порадовать свою любимую дочку, и решили, что лучше всего подарить ей путешествие в Нью-Йорк. Ведь это — такой оригинальный подарок! Так как Лиза — уже самостоятельная девочка, она вполне может сама выбрать, где ей поселиться в Нью-Йорке. Однако родители очень беспокоятся за нее, поэтому они разрешили ей выбирать только из  *$N$  безопасных* пятизвездочных отелей.

Лиза — общительная девочка, и у неё очень много друзей. Узнав, что она будет отмечать день рождения в Нью-Йорке, друзья немедленно отправились туда и поселились в  $C$  хостелах.

Лиза по случаю дня рождения устраивает маленькую вечеринку. К сожалению, никто из её друзей не хочет пропустить очередную серию нового реалити-шоу «*Home-2*», но как только серия закончится, все они одновременно выйдут из своих хостелов и поедут поздравлять Лизу. Она — девочка нетерпеливая, поэтому хочет выбрать отель так, чтобы последний друг приехал к ней как можно раньше, т.е. находился как можно ближе. Так как она тоже не хочет отвлекаться от просмотра высокоинтеллектуального шоу, она попросила вас помочь ей с выбором отеля. Конечно, за это она тоже пригласит вас на вечеринку!

Как известно, Нью-Йорк представляет собой идеальную прямоугольную сетку улиц: с севера на юг тянутся  $N$  авеню, а с запада на восток —  $M$  улиц. Наличием Бродвея мы в этой задаче пренебрежем. Можно считать, что каждый объект находится в узле этой сетки (и, таким образом, однозначно задается номером авеню и номером улицы, возле пересечения которых он расположен), а два соседних узла сетки находятся на расстоянии одного километра.

### Формат входных данных

В первой строке входных данных содержится два числа  $N$  и  $M$  — размеры города ( $1 \leq N, M \leq 10^9$ ). В следующей строке содержится единственное число  $C$  — количество хостелов ( $1 \leq C \leq 10^5$ ). Далее в  $C$  строках содержатся описания хостелов, каждый из них задается двумя координатами  $x$  и  $y$  ( $1 \leq x \leq N, 1 \leq y \leq M$ ). В следующей строке содержится одно число  $H$  — количество отелей ( $1 \leq H \leq 10^5$ ). В следующих строках содержатся описания отелей, в том же формате, что и хостелы.

Отель и хостел могут располагаться возле одного и того же перекрестка.

### Формат выходных данных

В первой строке выходных данных выведите одно число — искомое оптимальное расстояние. В следующей строке выведите номер любого из отелей, гарантирующих данное расстояние.

### Примеры

| <code>grid.in</code> | <code>grid.out</code> |
|----------------------|-----------------------|
| 10 10                | 6                     |
| 2                    | 2                     |
| 1 1                  |                       |
| 3 3                  |                       |
| 2                    |                       |
| 1 10                 |                       |
| 4 4                  |                       |

## Задача В. Корень из перестановки

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

Девочка Маша написала на доске последовательность из  $N$  различных натуральных чисел из диапазона от 1 до  $N$ . Затем к доске подошел хулиган Вася и написал свою последовательность: если на  $i$ -м месте в Машинной последовательности стоит число  $j$ , а на  $j$  — число  $k$ , то на  $i$ -ое место в своей последовательности Вася пишет число  $k$ . Затем он безжалостно стер Машину последовательность. Помогите бедной девушке восстановить утерянную последовательность!

### Формат входных данных

В первой строке входного файла записано число  $N$  ( $1 \leq N \leq 1000$ ). Во второй строке идут  $N$  чисел — последовательность, написанная Васей.

### Формат выходных данных

В выходной файл выведите последовательность, написанную Машей. Если существует несколько вариантов — выведите любой.

Гарантируется, что ответ существует.

### Примеры

| <code>root.in</code> | <code>root.out</code> |
|----------------------|-----------------------|
| 3<br>2 3 1           | 3 1 2                 |

## Задача С. Интеллектуальный отпуск

Имя входного файла: e.in  
Имя выходного файла: e.out  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Туристическая отрасль в этом сезоне столкнулась с серьёзными сложностями. Добросовестные туроператоры ищут новые рекламные ходы для продажи своих туров. Как известно, наиболее благоприятная для отдыха погода меняется плавно, причём не только от одного дня к другому, но и в течение суток.

Для большинства туристических направлений есть многолетние посекундные результаты измерений различных климатических параметров, например, температуры или влажности. У каждого человека своё понимание того, насколько различными могут быть подобные значения во время отпуска, но всех интересуют непрерывные туры как можно большей продолжительности.

Пусть мы зафиксировали туристическое направление и некоторый климатический параметр. Будем называть изменчивостью тура разницу между максимальным и минимальным значением выбранного параметра за всё время поездки. Для каждого туриста известно максимальное приемлемое значение изменчивости  $k_i$ .

Даны результаты измерений некоторого климатического параметра на одном из курортов и значения  $k_i$  для нескольких туристов. Требуется для каждого из них определить максимальный диапазон, подходящий для отпуска.

### Формат входных данных

В первой строке входного файла находится целое число  $N$  ( $1 \leq N \leq 600\,000$ ) — количество сделанных измерений. Во второй строке —  $N$  целых чисел, по модулю не превосходящих  $10^9$  — данные посекундных измерений.

В третьей строке входного файла находится число  $M$  ( $1 \leq M \leq 100$ ) — количество туристов, для которых необходимо найти оптимальный диапазон. В четвёртой строке —  $M$  целых чисел  $k_1, k_2, \dots, k_M$  ( $0 \leq k_i \leq 10^9$ ) — максимальная возможная разница между выбранным климатическим параметром в непрерывном диапазоне дней для каждого из туристов.

### Формат выходных данных

В выходной файл для каждого из  $M$  запросов в отдельной строке выведите два числа: номер первого измерения диапазона и номер последнего измерения, входящего в диапазон. Нумерация измерений ведётся с единицы. Если для некоторого туриста существует несколько подходящих диапазонов максимальной длины, выведите границы любого из них.

### Примеры

| e.in               | e.out |
|--------------------|-------|
| 7                  | 3 5   |
| 10 1 10 12 11 1 11 | 4 5   |
| 2                  |       |
| 2 1                |       |
| 9                  | 3 4   |
| 1 5 2 3 6 4 7 8 9  | 1 9   |
| 6                  | 7 9   |
| 1 10 2 4 5 0       | 2 6   |
|                    | 1 6   |
|                    | 1 1   |

## Задача D. Need for Speed

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Недавно какой-то школьник сломал стрелку спидометра на велосипеде Дениса Кириенко. Школьник, конечно, приклеил новую, но он мог приклеить её под неправильным углом. Таким образом, если спидометр показывает скорость  $s$ , на самом деле скорость равна  $s + c$ , где  $c$  — неизвестное вещественное число (возможно, отрицательное).

Денис Кириенко хочет узнать значение  $c$ . Он только что совершил поездку, которая состояла из  $n$  участков. На  $i$ -м участке он проехал  $d_i$  километров, а спидометр показывал  $s_i$ . Вся поездка заняла  $t$  часов.

Помогите ему определить значение  $c$ . Обратите внимание, что показания спидометра могли быть отрицательными, но настоящая скорость всегда была больше нуля.

### Формат входных данных

В первой строке заданы целые числа  $n$  и  $t$  — количество участков и общее время поездки ( $1 \leq n \leq 1000$ ,  $1 \leq t \leq 10^6$ ).

Следующие  $n$  строк описывают участки пути. На  $i$ -й строке записаны два целых числа  $d_i$  и  $s_i$  — длина  $i$ -го участка и показания спидометра на нём ( $1 \leq d_i \leq 1000$ ,  $|s_i| \leq 1000$ ).

### Формат выходных данных

Выведите  $c$  с абсолютной или относительной погрешностью не более  $10^{-6}$ .

### Примеры

| стандартный ввод                 | стандартный вывод |
|----------------------------------|-------------------|
| 3 5<br>4 -1<br>4 0<br>10 3       | 3.0000000         |
| 4 10<br>5 3<br>2 2<br>3 6<br>3 1 | -0.5086534        |

## Задача Е. Окна

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

На экране расположены прямоугольные окна, каким-то образом перекрывающиеся (со сторонами, параллельными осям координат). Вам необходимо найти точку, которая покрыта наибольшим числом из них.

### Формат входных данных

В первой строке входного файла записано число окон  $n$  ( $1 \leq n \leq 50\,000$ ). Следующие  $n$  строк содержат координаты окон  $x_{(1,i)}$   $y_{(1,i)}$   $x_{(2,i)}$   $y_{(2,i)}$ , где  $(x_{(1,i)}, y_{(1,i)})$  — координаты левого верхнего угла  $i$ -го окна, а  $(x_{(2,i)}, y_{(2,i)})$  — правого нижнего (на экране компьютера  $y$  растет сверху вниз, а  $x$  — слева направо). Все координаты — целые числа, по модулю не превосходящие  $10^6$ .

### Формат выходных данных

В первой строке выходного файла выведите максимальное число окон, покрывающих какую-либо из точек в данной конфигурации. Во второй строке выведите два целых числа, разделенных пробелом — координаты точки, покрытой максимальным числом окон. Окна считаются замкнутыми, т. е. покрывающими свои граничные точки.

### Примеры

| windows.in   | windows.out |
|--|-------------|
| 2<br>0 0 3 3<br>1 1 4 4                                  | 2<br>1 3    |
| 1<br>0 0 1 1   | 1<br>0 1    |
| 4<br>0 0 1 1<br>0 1 1 2<br>1 0 2 1<br>1 1 2 2            | 4<br>1 1    |
| 5<br>0 0 1 1<br>0 1 1 2<br>0 0 2 2<br>1 0 2 1<br>1 1 2 2 | 5<br>1 1    |

## Задача F. Разверни их. Полностью!

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Определим операцию  $\text{rev}(K, N)$  над числом  $N$  от 0 до  $2^K - 1$ , которая разворачивает биты в двоичной записи числа  $N$ . Например,  $\text{rev}(4, 5) = 10$ . Вам необходимо отвечать на запросы

$$\text{get}(A, B) = \left( \sum_{N=A}^B \text{rev}(k, N) \right) \bmod (1\,000\,000\,001).$$

### Формат входных данных

В первой строке входного файла заданы через пробел два целых числа  $K$  и  $T$  ( $1 \leq K \leq 31$ ,  $1 \leq T \leq 500\,000$ ). В следующих  $T$  строках идут запросы, состоящие из двух целых чисел  $A$  и  $B$ , разделённых пробелом  $0 \leq A \leq B < 2^K$ .

### Формат выходных данных

Для каждого теста выведите на отдельной строчке ответ.

### Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 25 2             | 252706816         |
| 1 16             | 44924975          |
| 17 16777000      |                   |
| 3 1              | 6                 |
| 3 3              |                   |

## Задача G. Кто в лес, кто по дрова?

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Лагерь был разбит на опушке леса. У команды из  $N$  человек оставалось две сложные задачи: отправиться в лес на разведку, а также подготовить дрова.

Команда ещё не до конца сформировалась как единое целое, поэтому было решено на каждое из заданий отправить набор людей, все из которых были бы попарно знакомы.

Так как бездельничать в команде не принято, каждый должен заняться ровно одним из данных дел: кто не в лес, тот по дрова, и наоборот.

### Формат входных данных

В первой строке записаны целые числа  $N$  и  $M$  — количество человек в команде и количество пар знакомых ( $1 \leq N, M \leq 100\,000$ ).

В следующих  $M$  строках записаны пары целых чисел  $a_i, b_i$ , означающие, что люди  $a_i$  и  $b_i$  знакомы между собой ( $1 \leq a_i, b_i \leq N, a_i \neq b_i$ ). Гарантируется, что все пары  $(a_i, b_i)$  различны.

### Формат выходных данных

Выведите  $N$  чисел: для каждого из  $N$  человек 0, если следует отправить его в лес, и 1, если по дрова.

Если распределение невозможно, выведите единственное число  $-1$ . Если же существует больше одного возможного ответа, выведите любой из них.

### Примеры

| стандартный ввод                       | стандартный вывод |
|--|-------------------|
| 3 1<br>1 2                             | 0 0 1             |
| 5 5<br>1 2<br>2 3<br>3 4<br>4 5<br>5 1 | -1                |
| 5 5<br>1 2<br>2 3<br>3 1<br>2 4<br>5 4 | 0 0 0 1 1         |

## Задача Н. Скобки

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 2 секунды         |
| Ограничение по памяти:  | 256 мегабайт      |

Правильной скобочной последовательностью называется такая последовательность круглых и квадратных скобок, которая могла бы встречаться в каком-нибудь арифметическом выражении. Например, последовательности  $() []$  и  $([])$  являются правильными скобочными последовательностями, а последовательности  $[] []$  и  $)() ($  — нет. Считается, что круглые и квадратные скобки равноправны, но круглой открывающей скобке обязательно должна соответствовать круглая закрывающая, а квадратной — квадратная.

Скобки считаются упорядоченными в таком порядке:  $(', ')$ ,  $[' , ']$ . Например,  $' )'$  больше чем  $' ('$ , но меньше чем  $' ['$ , а максимальной из всех скобок считается  $' ]'$ .

Говорят, что одна скобочная последовательность лексикографически меньше другой, если существует такое число  $k$ , что первые  $k$  скобок в них совпадают, а  $k + 1$ -ая скобка в первой меньше, чем во второй.

Программист Вася выписал на бумажке все правильные скобочные последовательности длины  $2n$  таким образом, что каждая написанная скобочная последовательность лексикографически меньше следующей за ней. Он очень обрадовался, что теперь может по лексикографическому номеру скобочной последовательности найти саму последовательность (последовательности, записанные на бумажке, нумеруются, начиная с 1). Однако бумажка потерялась... А повторять эту огромную работу Васе лень... Помогите бедному Васе написать программу, которая бы выдавала правильную скобочную последовательность по ее лексикографическому номеру!

### Формат входных данных

Во входном файле задано число  $n$  ( $1 \leq n \leq 20$ ) и произвольное натуральное число  $A$  — лексикографический номер искомой последовательности.

### Формат выходных данных

В первой строке выходного файла должна содержаться искомая скобочная последовательность.

### Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 2 1              | $(( ))$           |
| 2 3              | $() []$           |



## Задача I. Праздничная иллюминация

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 2 секунды         |
| Ограничение по памяти:  | 256 мегабайт      |

Для того, чтобы устроить праздничную иллюминацию по поводу проведения зимних сборов, у нас есть  $N$  разноцветных лампочек, занумерованных от 1 до  $N$ . Лампочки управляются четырьмя кнопками.

Кнопка 1. При нажатии на эту кнопку, все лампочки меняют состояние. Те, которые были включены, выключаются, а те, которые выключены — включаются.

Кнопка 2. При нажатии на эту кнопку меняют свое состояние лампочки с нечетными номерами.

Кнопка 3. При нажатии на эту кнопку меняют свое состояние лампочки с четными номерами.

Кнопка 4. При нажатии на эту кнопку меняют свое состояние все лампочки с номерами  $3K + 1 (K \geq 0)$ , т.е. 1,4,7...

Есть счетчик  $C$ , который записывает суммарное количество нажатий на кнопки. В начале праздника все лампочки включены, и счетчик  $C$  сброшен в 0.

По значению счетчика  $C$  и конечному состоянию некоторых лампочек необходимо найти все возможные конечные конфигурации лампочек.

### Формат входных данных

Первая строка входного файла содержит число  $N$  ( $1 \leq N \leq 100$ ). Во второй находится значение счетчика  $C$  ( $1 \leq C \leq 10\,000$ ). В третьей строке записаны номера лампочек, которые включены. Строка заканчивается числом  $-1$ . В четвертой строке в таком же формате заданы лампочки, которые выключены.

### Формат выходных данных

В выходной файл выведите всевозможные конечные состояния лампочек, по одному в строке. Каждая строка содержит  $N$  символов — 0, если соответствующая лампочка выключена, и 1 в противном случае. Конфигурации нужно выводить в лексикографическом порядке.

### Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 10               | 0000000000        |
| 1                | 0101010101        |
| -1               | 0110110110        |
| 7 -1             |                   |

## Задача J. Прямоугольники

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

Вася очень любит головоломки, в которых необходимо посчитать количество прямоугольников на большой картинке с горизонтальными и вертикальными отрезками. Сегодня он нарисовал прямоугольник  $m \times n$  на прямоугольной решетке на своем ноутбуке, несколько таких отрезков внутри, и предложил своему брату Саше решить ее. Но Саша не хочет проводить время, считая прямоугольники, которых так много, поэтому он решил использовать свой компьютер для этого. Он попросил Вас написать программу, которая найдет количество прямоугольников на прямоугольной решетке с несколькими вертикальными и горизонтальными отрезками на ней.

### Формат входных данных

Входной файл состоит из нескольких наборов тестов. Каждый тест начинается с со значений  $m$  и  $n$ . В следующей строке находится единственное число  $k$  — количество нарисованных отрезков. Далее идут описания отрезков. Каждый отрезок описывается четырьмя целыми числами  $x_1 y_1 x_2 y_2$ , где либо  $x_1 = x_2$ , либо  $y_1 = y_2$ . Все  $x$  из диапазона от 0 до  $m$ , а  $y$  — от 0 до  $n$ . Значения  $n$  и  $m$  не превосходят 100.

Входной файл заканчивается строкой, состоящей из двух 0.

Саша довольно неаккуратный. Он мог поместить один и тот же отрезок несколько раз во входной файл. Так же он мог разбить отрезок на несколько, возможно перекрывающихся, частей.

### Формат выходных данных

Для каждого теста из входного набора необходимо вывести количество прямоугольников на решетке, в соответствии с форматом, приведенным в примере.

### Примеры

| <code>rect.in</code>           | <code>rect.out</code>                     |
|--------------------------------|---|
| 2 2<br>2                       | Grid 1: The number of rectangles is<br>5. |
| 0 1 2 1<br>1 1 1 2<br>2 2<br>1 | Grid 2: The number of rectangles is<br>1. |
| 0 1 1 1<br>0 0                 |   |

## Задача К. Штурман

|                         |                            |
|-------------------------|----------------------------|
| Имя входного файла:     | <code>navigator.in</code>  |
| Имя выходного файла:    | <code>navigator.out</code> |
| Ограничение по времени: | 2 секунды                  |
| Ограничение по памяти:  | 64 мегабайта               |

Андрей Сергеевич — водитель со стажем. Но даже ему, порой, приходится нелегко на наших дорогах. Ремонтные работы, тупики и неожиданные развязки. Все это сопровождается знаками ограничения скорости или обгона. Без помощника в такой ситуации не обойтись.

После одной из поездок в летнюю смену ЛКШ, у Андрея Сергеевича остались записи его общения со штурманом. Записи могли иметь следующий вид:

1. `Begin of X` — начало населенного пункта  $X$
2. `End of X` — конец населенного пункта  $X$
3. `Speed limit y` — максимальная допустимая скорость на текущем участке —  $y$  км/ч
4. `No speed limit` — отмена текущего ограничения на скорость
5. `No overtaking` — на текущем участке запрещен обгон
6. `Overtaking` — на текущем участке разрешен обгон
7. `No restriction` — отмена ограничений на скорость и обгон
8. `Info` — после этой фразы штурман выдаёт все текущие ограничения

Если не установлены дополнительные ограничения, обгон разрешен, а максимальная скорость внутри населенного пункта составляет 60 км/ч. Вне населенного пункта — 90 км/ч. Вложенных населенных пунктов не бывает. Если сообщение «`No restriction`» появилось внутри населенного пункта, ограничение на скорость 60 км/ч продолжает действовать. Записи отмены ограничений могут присутствовать, даже если для них нет записи, задающей этого ограничения. Если подряд встречается несколько ограничений на скорость, то действует последнее из них, до тех пор, пока оно не будет отменено.

Андрей Сергеевич попросил вас написать программу, которая на каждый запрос последнего типа выдавала бы все текущие ограничения.

### Формат входных данных

В первой строке входного файла находится число  $N$  ( $1 \leq N \leq 100$ ) — количество записей. Следующие  $N$  строк содержат по одной записи штурмана каждая. Названия всех населенных пунктов не длинее 100 символов, ограничения на скорость являются натуральными числами, не превосходящими 90.

### Формат выходных данных

На каждую фразу последнего типа необходимо вывести три строки — текущее местонахождение, ограничение на скорость и разрешен ли обгон. Блоки необходимо разделять переводами строк. Придерживайтесь формата выходных данных, заданного в примере.

## Примеры

| navigator.in    | navigator.out            |
|-----------------|--------------------------|
| 9               | current location: none   |
| Info            | current speed limit: 90  |
| Begin of Moscow | overtake is allowed      |
| Speed limit 40  |                          |
| No overtaking   | current location: Moscow |
| Info            | current speed limit: 40  |
| No restriction  | overtake is not allowed  |
| End of Moscow   |                          |
| Speed limit 70  | current location: none   |
| Info            | current speed limit: 70  |
|                 | overtake is allowed      |

## Задача L. Список зачисленных в ЛКШ

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

Максим никак не мог дожидаться появления списка зачисленных в ЛКШ.Зима. Он каждую минуту обновлял страничку на сайте. Наконец, счастливый миг настал — список зачисленных был вывешен, и, что самое главное, Максим нашёл в нём свое имя.

Он немедленно решил поделиться своей радостью и разослал всем друзьям ВКонтакте сообщение о том, что он зачислен в ЛКШ. Ровно через минуту каждый друг Максима увидел это сообщение и проверил своё имя на сайте. Если друг также был зачислен в зимнюю смену, то он рассылал такие же сообщения всем своим друзьям, включая Максима. Ещё через минуту ребята, в первый раз увидевшие такие сообщения, искали свои имена в списках и, в случае попадания в смену, рассылали сообщения всем своим друзьям, и так далее. В результате оказалось, что все дети, подавшие заявки, узнали о том, зачислены ли они в зимнюю смену.

Витя не знает, кто с кем дружит, но он знает для каждого ребёнка, сколько минут прошло с момента появления информации на сайте до момента, когда ребёнок проверил наличие своего имени в списке зачисленных. Вите стало интересно, какое наименьшее и наибольшее количество сообщений могли отправить друг другу ЛКШата ВКонтакте?

### Формат входных данных

В первой строке записано единственное целое число  $N$  ( $1 \leq N \leq 10^5$ ) — количество ребят, подававших заявку в ЛКШ.Зима. В следующей строке записано  $N$  целых чисел,  $i$ -е из которых означает, через сколько минут после появления информации на сайте  $i$ -й человек проверил своё имя в списке зачисленных. Все числа целые, неотрицательные и не превосходят  $10^5$ .

### Формат выходных данных

Выведите «Impossible», если информация у Вити оказалась заведомо неверной. В противном случае выведите два целых числа — наименьшее и наибольшее количество сообщений, которые могли отправить друг другу ЛКШата.

### Примеры

| <code>messages.in</code> | <code>messages.out</code> |
|--------------------------|---------------------------|
| 2<br>0 1                 | 1 2                       |
| 5<br>2 0 1 2 3           | 6 12                      |
| 5<br>2 0 2 2 3           | Impossible                |

**Примечание** В первом примере минимум достигается, когда в ЛКШ зачислен только Максим. Максимум же достигается, когда зачислены оба школьника.

## Задача М. Праздничные дни

|                         |               |
|-------------------------|---------------|
| Имя входного файла:     | holidays.in   |
| Имя выходного файла:    | holidays.out  |
| Ограничение по времени: | 2 секунды     |
| Ограничение по памяти:  | 256 мегабайта |

Жители Флатландии очень любят праздники. Они считают, что любая более-менее важная памятная дата заслуживает того, чтобы ее отметить. Поэтому во Флатландии очень много государственных праздников. Настолько много, что там нет даже обычных выходных (как суббота и воскресенье в России) — жителям вполне хватает государственных праздников.

У каждого праздника есть день, когда он должен отмечаться. При этом день многих праздников определяется по очень сложным правилам; например, День защиты Флатландских волков отмечается в последнее полнолуние года, Международный День СУНЦов — в 42-й день летних каникул, а День Флатландского метеоролога — на следующий день после первого дождливого летнего четверга. (Правда, флатландские метеорологи настолько суровы, что могут предсказать, на какой день придется их профессиональный праздник, за год до него.)

Естественно, что при такой сложной и запутанной системе праздников нередко случается так, что несколько праздников выпадают на один и тот же день. Но флатландцы — очень систематичные люди, и они не могут отмечать несколько праздников в один и тот же день, поэтому в таких случаях праздники надо переносить.

Естественно, просто так переносить праздники нельзя. Перенос праздника допускается только при условии, что все дни между правильной датой праздника и тем днем, на который этот праздник переносится, также будут праздничными. Строго говоря, праздник с дня  $i$  можно отмечать в день  $j$  при одном из двух следующих условий:

- или  $i = j$ ,
- или  $i \neq j$ , но при этом в день  $i$ , а также во все дни между  $i$  и  $j$  не включительно отмечаются какие-то другие праздники (обратите внимание, что может быть как  $i < j$ , так и  $i > j$ ).

Таким образом, весь год будет состоять из чередующихся блоков рабочих и выходных дней, и перенос праздника допустим только при условии, что он попадает в тот же блок.

Флатландцы — последовательные люди, и придерживаются принципа «работать — так работать, отдыхать — так отдыхать». Поэтому они не любят частых смен выходных и рабочих дней. Поэтому праздники должны быть перенесены так, чтобы средняя длина блоков выходных дней была наибольшей.

Сегодня во Флатландии, как и во всем мире, наступает Новый год. Через несколько часов президент Флатландии выступит с обращением перед гражданами. По традиции, он должен озвучить календарь выходных и рабочих дней на наступающий год. Это расписание должен был подготовить Министр Флатландии по Праздникам, но, к сожалению, вчера отмечался День Министров Флатландии, и потому Министр не успел подготовить календарь — и не успеет, поскольку компьютером пользоваться он не умеет, а ручную составление календаря отнимает много времени.

Поэтому подготовить календарь поручено вам. Напишите программу, которая, зная, на какой день сколько праздников приходится, перенесет их в соответствии с правилами. Учтите, что нельзя праздники отмечать раньше первого дня года, и нельзя их отмечать позже последнего дня.

### Формат входных данных

На первой строке входного файла находится одно число  $n$  — количество дней во Флатландском году ( $1 \leq n \leq 4000$ ). На второй строке находятся  $n$  неотрицательных целых чисел,  $i$ -ое из которых указывает, сколько праздников приходится на  $i$ -ый день года.

Гарантируется, что сумма всех чисел во второй строке входного файла не превосходит  $n$ .

### Формат выходных данных

Выведите в выходной файл  $n$  символов: для каждого дня года выведите, должен он быть выходным или рабочим. Выводите символ 'X' (латинская заглавная буква X) для рабочего дня и символ '\*' (звездочка) для выходного.

## Примеры

| holidays.in      | holidays.out |
|------------------|--------------|
| 6<br>0 0 2 0 1 0 | XX***X       |
| 6<br>0 0 2 0 0 1 | X**XX*       |

## Задача N. Конфеты доверчивых друзей

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

Коля очень любит сладости. Однако на Новый год ему подарили всего лишь  $K$  конфет. Тогда Коля решил нажиться на  $N$  своих друзьях. Друзья занумерованы целыми числами от 1 до  $N$ . Степень доверия  $i$ -го друга Коле равна величине  $d_i$ .

Коля может совершать следующие действия:

1. Если Коля ничего не должен  $i$ -му другу, то он может взять у него в долг не более  $d_i$  конфет.
2. Если Коля занимал у  $i$ -го друга  $t$  конфет, и сейчас у него как минимум  $t + 1$  конфета, то он может вернуть долг этому другу, отдав при этом ещё одну лишнюю конфету в качестве компенсации за неудобства. После возвращения долга  $i$ -му другу его степень доверия  $d_i$  увеличивается на единицу.
3. Коля может взять все имеющиеся у него конфеты и скрыться в неизвестном направлении. Это действие он может выполнить только один раз, в самом конце своей операции.

Изначально Коля никому ничего не должен, ему известны все степени доверия друзей. Коле интересно, какое наибольшее количество конфет он сможет набрать перед тем, как скроется в неизвестном направлении.

### Формат входных данных

В первой строке через пробел записаны целые числа  $N$  и  $K$ . В следующей строке через пробел записаны  $N$  целых чисел — степени доверия друзей Коли. Все числа неотрицательные и не превосходят 1000.

### Формат выходных данных

Выведите единственное число — наибольшее количество конфет, которое может оказаться у Коли. Если же Коля может получить сколь угодно много конфет, выведите «Infinity».

### Примеры

| <code>friends.in</code> | <code>friends.out</code> |
|-------------------------|--------------------------|
| 1 1<br>3                | 4                        |
| 2 0<br>0 0              | 0                        |