Задача A. Разминочные лягушачьи игры

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

В тридесятом царстве в новогодние праздники все лягушки собираются на самом большом болоте, чтобы поиграть в замечательную игру. Всего в этом царстве живет N зеленых лягушек и M коричневых. Для игры они выбирают на болоте $N + M + 1$ кочку, на первые N кочек слева садятся зеленые лягушки, а на последние M — коричневые (т. е. между ними находится одна кочка, на которой никто не сидит). Зеленые лягушки садятся лицом к коричневым лягушкам, а коричневые — к зеленым. Кочки настолько маленькие, что развернуться на них, не свалившись в болото, совершенно не возможно. Поэтому лягушки могут двигаться только вперед и не могут разворачиваться.

На каждом ходе игры одна из лягушек перепрыгивает с той кочки, где она сидит, на свободную кочку. При этом лягушка может прыгнуть на соседнюю кочку вперед, либо перепрыгнуть через одну кочку, если соседняя занята.

Чтобы праздник удался, зеленые лягушки должны оказаться на последних кочках, а коричневые — на первых. Порядок, в котором лягушки окажутся на кочках, не важен. Так как на праздник каждый раз приходит разное количество лягушек, то им каждый год приходится придумывать очередность прыжков. Напишите программу, которая поможет лягушкам составить план прыжков.

Формат входных данных

Во входном файле записаны два числа N и M ($1 \leqslant N, M \leqslant 1000$) — количество зеленых и коричневых лягушек соответственно.

Формат выходных данных

Выведите последовательность прыжков лягушек для достижения поставленной цели. Каждый прыжок можно задать одним числом — номером прыгающей лягушки (поскольку свободная кочка всегда ровно одна). Пронумеруем всех лягушек в соответствии с их начальным положением. Зеленые лягушки будут пронумерованы числами от 1 до N, а коричневые — с $N + 1$ до $N + M$ в порядке слева направо.

В первую строку выходного файла выведите число K — количество прыжков. K не должно превышать 10^7. Далее выведите K чисел — номера лягушек.

Если же достичь требуемой рассадки лягушек нельзя, выведите одно число -1.

Пример

стандартный ввод	стандартный вывод
`2 1`	`5` `2 3 2 1 3`

Задача B. Прямоугольнирование квадрата

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

Вам дан квадрат размера $N \times N$, который разделён сеткой на N^2 одинаковых маленьких квадратиков размера 1×1.

Любители Квадратных Шаблонов просят вас разбить этот квадрат на несколько различных (больше одного) прямоугольников так, чтобы стороны полученных прямоугольников проходили по сетке исходного квадрата, а разность площадей любых двух прямоугольников не превосходила $1.5 \times N$. Два прямоугольника считаются равными, если один из них можно превратить в другой с помощью перемещений и поворотов.

Формат входных данных

В единственной строке ввода дано одно целое число N ($1 \leqslant N \leqslant 10^6$).

Формат выходных данных

В первой строке выведите S – количество прямоугольников в разбиении.

В последующих S строках выведите описания найденных прямоугольников. Каждая из этих строк должна содержать 4 числа x, y, wx, wy – координаты левого нижнего угла, ширина и высота прямоугольника соответственно.

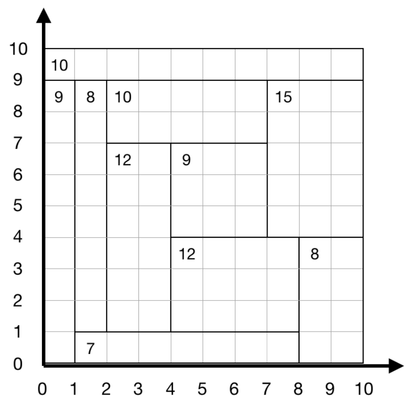
Если искомого разбиения не существует, в единственной строке файла выведите число -1.

Пример

стандартный ввод	стандартный вывод
10	10 0 0 1 9 1 0 7 1 8 0 2 4 1 1 1 8 2 1 2 6 4 1 4 3 4 4 3 3 7 4 3 5 2 7 5 2 0 9 10 1

Замечание

Пояснение к первому примеру:



В первом примере максимальная попарная разность между площадями прямоугольников равна 8.

Задача C. Игра со стеком

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

У Вани есть натуральное число n. Он придумал следующую игру. У него есть стек, в котором изначально лежит единственное число 1. Он может выполнять следующую операцию: пусть сейчас в стеке лежит s целых чисел $a_1, a_2, \ldots, a_s$, в порядке от самого первого элемента стека к самому последнему. Ваня может выбрать два целых числа $1 \leqslant l \leqslant r \leqslant s$ и положить число $a_l + a_{l+1} + \ldots + a_r$ в конец стека.

Ваня хочет сделать последнее число в стеке равным n. Определите минимальное количество операций, которое для этого требуется и постройте любой возможный пример с этим количеством операций.

Формат входных данных

В первой строке находится единственное целое число t ($1 \leqslant t \leqslant 1000$) — количество тестовых случаев.

В следующих t строках находится по одному целому числу n ($1 \leqslant n \leqslant 10^{18}$) — число Вани.

Формат выходных данных

Выведите ответы на тестовые случаи в порядке их следования во входных данных. Для тестового случая выведите ответ в следующем формате:

В первой строке выведите единственное целое число q ($0 \leqslant q \leqslant 1000$) — минимальное количество операций со стеком, которое нужно сделать, чтобы последнее число стало равно n. Гарантируется, что за $\leqslant 1000$ операций возможно сделать последнее число равным n. В следующих строках выведите по два целых числа — описания операций. В i-й строке выведите два целых числа l_i, r_i ($1 \leqslant l_i \leqslant r_i \leqslant i$). Во время i-й операции в конец стека кладется число $a_{l_i} + a_{l_i+1} + \ldots + a_{r_i}$. После выполнения предоставленных операций последнее число стека должно быть равно n.

Все тестовые случаи независимы.

Пример

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

Задача D. Баланс соседей

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

Уже совсем скоро начнется отбор в олимпиадную школу, а еще столько всего предстоит сделать! Организаторы долго разбирались с тем, как расселить всех участников, поэтому не успели придумать легенду к этой задаче...

От вас требуется построить связный неориентированный граф с n вершинами, пронумерованными целыми числами от 1 до n. Обозначим за s_v сумму номеров вершин, смежных с вершиной v. Граф должен обладать следующим свойством: значения s_v для всех v должны быть одинаковыми.

Граф не должен содержать петли и кратные ребра.

Формат входных данных

Единственная строка содержит одно целое число n ($3 \leqslant n \leqslant 100$).

Формат выходных данных

В первой строке выведите число ребер m в построенном графе.

В каждой из следующих m строк выведите по два числа — номера вершин, соединенных очередным ребром.

Если решений несколько, выведите любое. Можно показать, что ответ всегда существует.

Пример

стандартный ввод	стандартный вывод
3	2 1 3 2 3

Замечание

В примере из условия вершина 1 соединена только с вершиной 3 (сумма равна 3), вершина 2 соединена только с вершиной 3 (сумма равна 3), а вершина 3 соединена с вершинами 1 и 2 (сумма также равна 3).

Задача E. Точки на плоскости

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

На плоскости расположены n точек (x_i, y_i) с целочисленными координатами от 0 до 10^6. Расстоянием между двумя точками с номерами a и b назовем следующую величину:

$$\mathrm{dist}(a, b) = |x_a - x_b| + |y_a - y_b|,$$

(расстояние, вычисляемое по такой формуле, называется **манхэттенским расстоянием**).

Назовем **гамильтоновым путем** некоторую перестановку p_i от 1 до n. Назовем длиной этого пути величину:

$$\sum_{i=1}^{n-1} \mathrm{dist}(p_i, p_{i+1}).$$

Найдите какой-нибудь гамильтонов путь с длиной не более, чем 25×10^8. Обратите внимание, минимизировать длину гамильтонова пути не требуется.

Формат входных данных

В первой строке дано число n $(1 \leqslant n \leqslant 10^6)$.

В $i + 1$-й строке даны координаты точки с номером i: x_i и y_i $(0 \leqslant x_i, y_i \leqslant 10^6)$.

Гарантируется, что никакие две точки не совпадают.

Формат выходных данных

Выведите перестановку чисел p_i от 1 до n — искомый гамильтонов путь. Перестановка должна удовлетворять неравенству:

$$\sum_{i=1}^{n-1} \mathrm{dist}(p_i, p_{i+1}) \leqslant 25 \times 10^8.$$

Если возможных ответов несколько, разрешается вывести любой.

Гарантируется, что существует подходящая перестановка.

Пример

стандартный ввод	стандартный вывод
5 0 7 8 10 3 4 5 0 9 12	4 3 1 2 5

Задача F. Чудеса Техноереси

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

Служителю культа Адептус Механикус поручили создать новую СШК (стандартную шаблонную конструкцию) для улучшения Имперских бэйнблейдов на основе новых технологий, украденных у ксеносов. СШК, требуемая от механикума, представляет собой n двигателей, каждая пара которых соединена проводами разных типов от 0 до 9. Чтобы СШК работала нормально, нужно выбрать типы проводов так, чтобы не существовало никаких 3 или 5 двигателей, соединенных проводами одного типа(цикл из 3 или 5 двигателей). Бедный механикум умеет только чинить тостеры, поэтому такое поручение для него слишком сложное. Помогите ему!

Формат входных данных

На вход подаётся одно целое число n — количество двигателей в СШК ($2 \leqslant n \leqslant 1000$).

Формат выходных данных

Если не существует подходящей разметки, выведите -1. В противном случае выведите $n-1$ строк, где i-я строка содержит $(n-i)$ символов. j-й символ в i-й строке соответствует типу провода $(i, i+j)$. Если существует несколько вариантов, выведите любой из них.

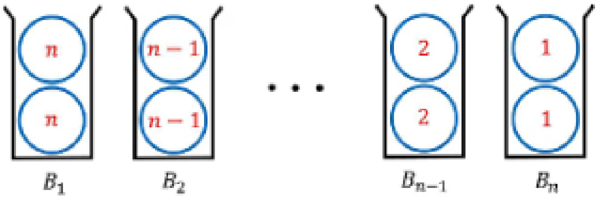
Пример

стандартный ввод	стандартный вывод
4	010 01 0

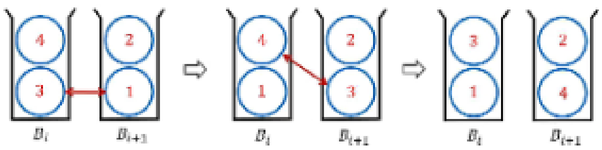
Задача G. Имперское поле экспериментов

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мегабайта

Одному имперскому коммисару надоело расстреливать еретиков и дезертиров, и он решил найти себе новое интересное занятие. Он выстроил солдат Имперской армии в n. Ряды помечены целыми числами: для каждого ряда i от 1 до n есть ровно два солдата с номером i. Ряды обозначены как $B_1, \ldots, B_n$ слева направо. Каждый ряд может содержать не более двух человек. Изначально ряд B_i содержит два солдата с номером $n + 1 - i$.



Далее коммисару захотелось отсортировать солдат так, чтобы ряд B_i содержал оба солдата с номером i. Однако, он может только выбирать любых двух солдат из двух соседних рядов и менять их местами.



Коммисару очень хочется выполнить не более $0.7n^2$ операций, однако думать он не хочет, поэтому он повесил эту задачу на одного из новоприбывших огринов. Огрин слишком занят решением примера 1+1, поэтому задачу придется решать вам.

Формат входных данных

Одна строка с целым числом n ($3 \leqslant n \leqslant 100$).

Формат выходных данных

Первая строка — количество операций S. Следующие S строк содержат тройки j, a, b, где:

- j — номер ряда B_j (соседний с B_{j+1}),
- a — номер солдата из B_j,
- b — номер солдата из B_{j+1}.

Примеры

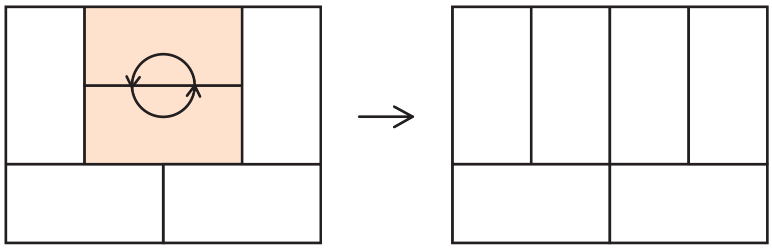
стандартный ввод	стандартный вывод
3	5 1 3 2 2 3 1 1 3 1 2 3 1 1 2 1
4	10 1 4 3 2 4 2 3 4 1 1 4 2 2 4 1 3 4 1 1 3 1 2 3 1 2 3 2 1 2 1

Задача H. Parquet Re-laying

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Петя решил положить паркет в комнате размером $n \times m$, паркет состоит из досок размером 1×2. Когда строители уложили паркет, выяснилось, что его рисунок выглядит совсем не так, как нравится Пете, и строителям придется его переложить.

Однако строители решили, что снимать паркет целиком и потом раскладывать его заново очень сложно, поэтому каждый час они будут делать такую операцию: вынуть какие-то две доски, образующие квадрат 2×2, повернуть их на 90 градусов и положить на то же место.



При этом у них нет никаких идей, как с помощью таких операций получить требуемое расположение, и возможно ли это вообще.

Помогите Пете составить план для рабочих или скажите, что это невозможно. План должен содержать не более 100 000 команд.

Формат входных данных

Первая строка входного файла содержит числа n и m — размер комнаты ($1 \leqslant n, m \leqslant 50$).

Следующие n строк содержат по m символов — описание текущего положения досок паркета. Каждый символ обозначает положение половинки доски. Символы '`L`', '`R`', '`U`' и '`D`' соответствуют левой, правой, верхней и нижней половинкам соответственно.

Следующие n строк содержат по m символов содержат описание требуемой конфигурации в том же формате.

Формат выходных данных

В первой строке выведите k — число операций в плане для рабочих. В следующих k строках выведите описания операций. Операция задается координатами (строка и столбец) левой верхней половинки доски, над которыми проводится операция.

Если решения нет, выведите в первой строке `-1`.

Система оценки

Подзадача	Баллы	Ограничения	Оценка	Необх. подзадачи
1	13	$n, m \leqslant 4$	подзадача	—
2	28	$n, m \leqslant 20$	подзадача	1
3	59	$n, m \leqslant 50$	подзадача	1, 2

В первой и второй подзадачах сообщаются результаты проверки на каждом тесте подзадачи.

В третьей подзадаче сообщаются баллы за подзадачу и результат проверки первого не пройденного теста.

Пример

стандартный ввод	стандартный вывод
2 3 ULR DLR LRU LRD	2 1 2 1 1

Задача I. Минимальная уникальная подстрока

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

Пусть у вас есть строка s из символов "0" и "1". Будем называть строку t подстрокой строки s, если существует $1 \leqslant l \leqslant |s| - |t| + 1$, такое что $t = s_l s_{l+1} \ldots s_{l+|t|-1}$. Будем называть подстроку t строки s уникальной, если существует единственное такое l.

Например, пусть $s =$"1010111". Тогда $t =$"010" является уникальной подстрокой s, так как существует единственное подходящее $l = 2$. Заметим, что $t =$"10" не является уникальной подстрокой s, так как подходят $l = 1$ и $l = 3$. А, например, $t =$"00" вообще не является подстрокой строки s, так как не существует подходящих l.

Сегодня Вася на уроке информатики решал такую задачу: дана строка из символов "0" и "1", надо найти длину её кратчайшей уникальной подстроки. Написав решение к этой задаче, он решил его протестировать. Он просит помощи у вас.

Вам даны 2 таких целых положительных числа n и k, что $(n \bmod 2) = (k \bmod 2)$, где $(x \bmod 2)$ — это операция взятия остатка числа x при делении на 2. Найдите любую строку s состоящую из n символов "0" и "1", такую что наименьшая длина её уникальной подстроки равна k.

Формат входных данных

В первой строке даны два целых числа n и k, разделённые пробелом ($1 \leqslant k \leqslant n \leqslant 100\,000$, $(k \bmod 2) = (n \bmod 2)$).

Формат выходных данных

Выведите строку s длины n, состоящую из символов "0" и "1". Минимальная длина уникальной подстроки s должна равняться k. Среди таких строк разрешается вывести **любую**. Гарантируется, что хотя бы одна подходящая строка существует.

Примеры

стандартный ввод	стандартный вывод
4 4	1111
5 3	01010
7 3	1011011

Замечание

В первом тесте легко видеть, что единственной уникальной подстрокой строки $s =$"1111" является вся строка s, длина которой 4.

Во втором тесте у строки $s =$"01010" минимальной по длине уникальной подстрокой является строка $t =$"101" длина которой 3.

Во третьем тесте у строки $s =$"1011011" минимальной по длине уникальной подстрокой является строка $t =$"110" длина которой 3.

Задача J. Головоломка из фишек

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

Егор придумал новую головоломку из фишек, в которую предлагает сыграть вам.

Головоломка имеет вид таблицы из n строк и m столбцов, в каждой клетке которой могут располагаться несколько черных и белых фишек, положенных в ряд. Таким образом, состояние в клетке можно описать строкой, состоящей из символов «`0`» (белая фишка) и «`1`» (черная фишка), возможно пустой, а вся головоломка — это таблица, в каждой клетке которой стоит строка из нулей и единиц. Задача состоит в том, чтобы из одного состояния таблицы получить другое.

Для этого можно использовать следующую операцию:

- выбрать две различные клетки (x_1, y_1) и (x_2, y_2), при этом клетки должны находиться в одной строке или одном столбце таблицы, и строка в клетке (x_1, y_1) должна быть непустой;
- за операцию можно переместить последний символ строки в клетке (x_1, y_1) в начало строки в клетке (x_2, y_2).

Для вас Егор загадал 2 состояния таблицы — начальное и конечное. Гарантируется, что количества нулей и единиц в таблицах совпадают. Ваша задача — с помощью нескольких операций получить из начального состояния конечное. Конечно, Егор не хочет, чтобы количество операций было очень большим. Обозначим за s количество символов в каждой из таблиц (в таблицах поровну символов). Тогда вы должны использовать не более $4 \cdot s$ операций.

Формат входных данных

В первой строке записаны два целых числа n и m ($2 \leqslant n, m \leqslant 300$) — количества строк и столбцов у таблиц в головоломке, соответственно.

В следующих n строках находится описание начального состояния таблицы в следующем формате: в каждой строке находится m непустых строк, состоящих из нулей и единиц. В i-й из этих строк, j-я строка описывает строку, записанную в клетке (i, j). Строки нумеруются от 1 до n, столбцы нумеруются от 1 до m.

В следующих n строках находится описание конечного состояния таблицы в аналогичном формате.

Обозначим суммарную длину строк в начальном состоянии за s. Гарантируется, что $s \leqslant 100\,000$. Также гарантируется, что количества нулей и единиц совпадают в начальном и конечном состояниях.

Формат выходных данных

В первой строке выведите одно целое число q — количество использованных операций. Необходимо найти решение, для которого $0 \leqslant q \leqslant 4 \cdot s$.

В следующих q строках выведите по 4 целых числа x_1, y_1, x_2, y_2. i-я из этих строк должна описывать i-ю операцию. Необходимо, чтобы было выполнено $1 \leqslant x_1, x_2 \leqslant n$, $1 \leqslant y_1, y_2 \leqslant m$, $(x_1, y_1) \neq (x_2, y_2)$, $x_1 = x_2$ или $y_1 = y_2$. При этом строка в клетке (x_1, y_1) должна быть непустой. Данная последовательность операций должна переводить таблицу из начального состояния в конечное.

Можно показать, что ответ существует. Если есть несколько возможных решений, выведите любое.

Примеры

стандартный ввод	стандартный вывод
2 2 00 10 01 11 10 01 10 01	4 2 1 1 1 1 1 1 2 1 2 2 2 2 2 2 1
2 3 0 0 0 011 1 0 0 0 1 011 0 0	4 2 2 1 2 1 2 2 2 1 2 1 3 1 3 1 2

Замечание

Рассмотрим первый пример.

- Текущее состояние таблицы:

  ```
  00 10
  01 11
  ```

  Первая операция. В клетке $(2, 1)$ записана строка 01. Применяя операцию к двум клеткам $(2, 1)$ и $(1, 1)$, мы переносим 1 из конца строки 01 в начало строки 00, получая строку 100.

- Текущее состояние таблицы:

  ```
  100 10
  0   11
  ```

  Вторая операция. В клетке $(1, 1)$ записана строка 100. Применяя операцию к двум клеткам $(1, 1)$ и $(1, 2)$, мы переносим 0 из конца строки 100 в начало строки 10, получая строку 010.

- Текущее состояние таблицы:

  ```
  10 010
  0  11
  ```

  Третья операция. В клетке $(1, 2)$ записана строка 010. Применяя операцию к двум клеткам $(1, 2)$ и $(2, 2)$, мы переносим 0 из конца строки 010 в начало строки 11, получая строку 011.

- Текущее состояние таблицы:

  ```
  10 01
  0  011
  ```

  Четвертая операция. В клетке $(2, 2)$ записана строка 011. Применяя операцию к двум клеткам $(2, 2)$ и $(2, 1)$, мы переносим 1 из конца строки 011 в начало строки 0, получая строку 10.

- Текущее состояние таблицы:

  ```
  10 01
  10 01
  ```

Можно видеть, что мы получили требуемое состояние таблицы.

Задача K. Тест на терпение

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

Как вы знаете, Энакину не хватает терпения, поэтому мастер Оби-Ван решил дать много одинаковых задачек своему ученику. А именно, он дал ему n троек чисел a_i, b_i, c_i и попросил найти для каждой тройки два числа x_i и y_i таких, что побитовое И этих двух чисел равно a_i, побитовое ИЛИ этих чисел равно b_i, а их разность $x_i - y_i$ равна c_i. У Энакина нет времени на решение задач, поэтому он просит вас сделать это задание вместо него. Учтите, что Оби-Ван торопился на совет джедаев, поэтому среди троек могут быть такие, для которых решение не существует.

Формат входных данных

В первой строчке дано целое число n — количество троек ($1 \leqslant n \leqslant 10000$).

В следующих n строчках даны описания троек — три целых числа a_i, b_i, c_i ($0 \leqslant a, b, c \leqslant 10^9$).

Формат выходных данных

Для каждой тройки определите, существует ли два числа, удовлетворяющих условиям.

Если существует, то выведите в одной строчке два целых числа x_i, y_i — ответ для тройки i. Если существует несколько правильных пар чисел, являющихся ответом, разрешается вывести любую.

Если не существует, то выведите в одной строчке -1.

Пример

стандартный ввод	стандартный вывод
3 8 14 2 2 7 3 8 13 4	12 10 6 3 -1

Замечание

Побитовое И двух неотрицательных целых чисел a и b определяется так. Сначала два числа записываются в двоичной системе счисления одно над другим так, чтобы последняя цифра второго числа оказалась точно под последней цифрой первого числа. Далее, если записи чисел оказались неравной длины, то более короткое из двух чисел дополняется слева нулями, пока цифр в числах не станет поровну. Дальше под этими двумя числами записывается третье двоичное число по следующим правилам: если на i-м месте в первом и втором числе стоят `1`, то в третьем числе на i-м месте записывается `1`; в противном случае там записывается `0`. Построенное таким образом третье число и является двоичной записью побитового И данных двух чисел.

Например, рассмотрим числа 17 и 71. В двоичном виде они записываются как `10001` и `1000111`. Дополним первое число двумя нулями слева, чтобы в нём стало семь цифр, как и во втором числе: `0010001`. Напишем под парой цифр `1`, если они обе равны `1`, и `0` иначе:

```
0010001
1000111
-------
0000001
```

Если стереть лидирующие нули, получится `1` — двоичная запись числа 1. Таким образом, $17 \, \mathrm{AND} \, 71$ — побитовое И чисел 17 и 71 — равно 1.

Похожим образом определяется побитовое ИЛИ. Мы так же выписываем числа, дополняем нулями, но под двумя цифрами ставим `1` в случае, если хотя бы одна из цифр равна `1`, а иначе ставим `0`. В частности, для тех же чисел 17 и 71 имеем:

```
0010001
1000111
-------
1010111
```

Полученная двоичная запись представляет число $2^6 + 2^4 + 2^2 + 2^1 + 2^0 = 87$. Таким образом, $17 \, \mathrm{OR} \, 71$ — побитовое ИЛИ чисел 17 и 71 — равно 87.

В языке Pascal побитовые И и ИЛИ вычисляются с помощью команд a `and` b и a `or` b, а в C++, Java, Python — с помощью a `&` b и a `|` b.