

Задача А. Почти $a + b$

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

На клетчатом поле задан прямоугольник размером $a \times b$ клеток, где a — высота прямоугольника, а b — ширина. Требуется определить, сколькими различными способами можно разместить внутри него квадрат со стороной x клеток.

Квадрат должен полностью находиться внутри прямоугольника, его стороны должны проходить по границам клеток и быть параллельны сторонам прямоугольника. Квадрат должен занимать ровно $x \times x$ клеток.

Два способа размещения считаются различными, если они отличаются положением хотя бы одной клетки квадрата. Другими словами, каждое возможное положение верхнего левого угла квадрата считается отдельным способом.

Формат входных данных

В единственной строке входных данных содержатся три целых числа a , b и x ($1 \leq a, b \leq 10^9$, $1 \leq x \leq \min(a, b)$) — высота и ширина прямоугольника, сторона квадрата.

Формат выходных данных

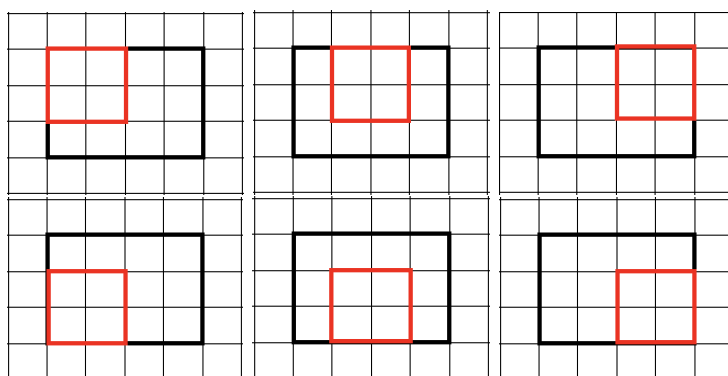
Выведите одно целое число — количество способов разместить квадрат $x \times x$ внутри прямоугольника $a \times b$.

Примеры

стандартный ввод	стандартный вывод
3 4 2	6
1000000000 1000000000 1	1000000000000000000
5 5 5	1

Замечание

Ниже можно увидеть все возможные варианты размещения квадрата внутри прямоугольника в первом примере:



Задача В. Какой сегодня день?

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

На планете Чиллалекс времяисчисление устроено иначе, чем на Земле:

- Месяц состоит из четырёх недель длительностью a , b , c и d дней соответственно.
- Дни на первой неделе месяца нумеруются целыми числами от 1 до a .
- Дни на второй неделе месяца нумеруются целыми числами от $a + 1$ до $a + b$.
- Дни на третьей неделе месяца нумеруются целыми числами от $a + b + 1$ до $a + b + c$.
- Дни на четвёртой неделе месяца нумеруются целыми числами от $a + b + c + 1$ до $a + b + c + d$.

На календаре сегодня день x текущего месяца. Так как **номер** недели достаточно важен для построения жизненных планов, вас очень интересует, какой **он** будет через y дней.

Обратите внимание, что за y дней может поменяться (и не один раз) как номер недели, так и номер месяца.

Формат входных данных

В единственной строке входных данных содержится шесть целых чисел a , b , c , d , x и y — количество дней в первой, второй, третьей и четвёртой неделях соответственно; текущий номер дня в месяце и количество дней, по прошествии которых нужно узнать номер недели в месяце.

Гарантируется, что в этой задаче $1 \leq a, b, c, d \leq 10^5$, $1 \leq x \leq a + b + c + d$, $0 \leq y \leq 10^5$.

Формат выходных данных

Выведите одно целое число от 1 до 4 — номер недели дня, который будет по прошествии y дней, начиная с сегодняшнего дня с номером x .

Примеры

стандартный ввод	стандартный вывод
2 3 2 3 1 5	3
3 3 3 3 8 10	2
1 2 3 4 1 19	4
4 3 2 1 10 31	1

Замечание

Разберём первые три примера из условия:

1. В первом примере текущий день является первым в месяце. Через пять дней будет шестой день месяца, который является частью третьей недели.
2. Во втором примере текущий день является восьмым в месяце. Через десять дней будет шестой день следующего месяца, который является последним днём второй недели.
3. В третьем примере текущий день является первым в месяце. Через девятнадцать дней будет десятый день следующего месяца, который является последним днём четвёртой недели.

Задача С. Чередующаяся чётность

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

Вам дано целое число x . Ваша задача проверить, что в числе x чередуются чётные и нечётные цифры.

Например, в числах 32, 121, 298, 12345 и 85436 чередуются чётные и нечётные цифры, а в числах 11, 1235 и 2334 нет.

Формат входных данных

В единственной строке входных данных содержится одно целое число x ($1 \leq x \leq 10^9$).

Формат выходных данных

Для каждого теста выведите строку «YES», если в числе x чередуются чётные и нечётные цифры, и «NO» иначе.

Вы можете выводить «YES» и «NO» в любом регистре (например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительный ответ).

Примеры

стандартный ввод	стандартный вывод
123456789	YES
23445	NO
74	YES
1	YES
1000000000	NO
232	YES
2120	NO

Задача D. Олимпиада КоРОЧе

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

На олимпиаде КоРОЧе (Командная Региональная Олимпиада Челябинской области) в 2004 году участвовали n команд. Команда с номером i решила a_i задач.

Команда жюри была настолько занята отслеживанием корректности работы тестирующей системы, что не успела распределить места между командами. И вот в 2025-м году пришло время распределить места на той давно забытой олимпиаде.

Ваша задача — помочь жюри и определить, какое место заняла каждая из команд. Место на олимпиаде определяется как количество команд, набравших строго больше задач, плюс 1. Обратите внимание, что команды с одинаковым количеством задач занимают одинаковое место.

Формат входных данных

В первой строке входных данных содержится одно целое число n ($1 \leq n \leq 100$) — количество команд-участниц.

Во второй строке входных данных содержится n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 100$) — количество решённых задач каждой из команд.

Формат выходных данных

В единственной строке выходных данных выведите n целых чисел p_i , разделённых пробелом, где p_i означает место, которое заняла команда с номером i .

Примеры

стандартный ввод	стандартный вывод
6 5 5 4 3 3 2	1 1 3 4 4 6
5 1 5 4 2 3	5 1 2 4 3

Замечание

Ответом на первый пример из условия будет являться последовательность $p = [1, 1, 3, 4, 4, 6]$. Первые две команды решили больше всех задач, поэтому разделили между собой первое место. Третья команда заняла третье место, потому что строго больше задач, чем она, решили только первая и вторая команды.

Задача Е. Волшебное зеркало

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

Строка, состоящая только из цифр, называется *красивой*, если она не меняется при отражении в волшебном зеркале.

Волшебное зеркало изменяет строку по такому правилу:

- Строка разворачивается.
- Цифры 0, 1, 8 не изменяются.
- Цифры 2 и 5 меняются друг на друга.
- Цифры 6 и 9 меняются друг на друга.
- Цифры 3, 4, 7 становятся некорректными.

Например, строка 5692 *красивая*, а строка 1122 в отражении становится строкой 5511, поэтому *красивой* не является.

Вам дана строка s , состоящая из цифр. Необходимо проверить, является ли она *красивой*.

Формат входных данных

Единственная строка входных данных содержит строку s ($1 \leq |s| \leq 10^5$), состоящую из цифр.

Формат выходных данных

Для каждого теста выведите строку «YES», если строка s является *красивой*, и «NO» иначе.

Вы можете выводить «YES» и «NO» в любом регистре (например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительный ответ).

Примеры

стандартный ввод	стандартный вывод
5692	YES
1122	NO
333	NO
848	NO

Задача F. Размещение людей

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

В вагон метро, представляющий из себя единственный ряд из n сидений, заходят по очереди n человек. Так как каждый из пассажиров не любит сидеть рядом с другими людьми, заходящие пассажиры выбирают себе пустое место, расстояние от которого до ближайшего занятого места **максимально**. В случае, если таких мест несколько, пассажир сядет на ближайшее к входу в вагон место.

Пронумеруем заходящих в вагон людей от 1 до n в порядке их входа, а итоговую рассадку в вагоне представим массивом чисел, где вход в вагон будет перед первым элементом массива, а на i -й позиции будет номер человека, который сел на i -е место.

Определите, как в итоге рассядутся люди, заходящие в вагон с n пустыми местами.

Формат входных данных

В единственной строке входных данных содержится одно целое число n ($1 \leq n \leq 100$) — количество сидений в вагоне и человек, которые зайдут в него по очереди.

Формат выходных данных

В единственной строке выходных данных выведите n чисел через пробел: i -е число должно быть равно номеру человека, выбравшего i -е место.

Примеры

стандартный ввод	стандартный вывод
5	1 4 3 5 2
6	1 4 3 5 6 2
11	1 6 4 7 8 3 9 5 10 11 2
12	1 6 5 7 8 3 9 10 4 11 12 2

Замечание

В первом примере из условия пассажиры выбирают места следующим образом:

1. Первый пассажир выберет первое место, так как оно самое левое (и ближайшее к входу).
2. Второй пассажир займет последнее место, так как оно находится дальше всего от занятого места.
3. Третий пассажир сядет на третье место, так как оно максимально удалено от первого и последнего места.
4. Четвертый пассажир займет второе место, так как второе и четвертое места одинаково удалены от занятых мест, но второе место находится левее.
5. Пятый пассажир займет единственное свободное место.

Задача G. Просто считаем квадраты

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

Вам даны два целых числа l и r . Ваша задача посчитать количество таких неотрицательных целых чисел x , что $l \leq x^2 \leq r$.

Формат входных данных

В единственной строке входных данных содержится два целых числа l и r ($0 \leq l \leq r \leq 10^{18}$).

Формат выходных данных

Вывести единственное целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
2 16	3
0 1000000000000000000	1000000001
123456789 987654321987654321	993796879

Замечание

В первом примере из условия подойдут следующие значения x :

- $x = 2$, так как $2 \leq 4 \leq 16$.
- $x = 3$, так как $2 \leq 9 \leq 16$.
- $x = 4$, так как $2 \leq 16 \leq 16$.

Задача Н. Число и его квадрат

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

Вам дан массив целых чисел $a_1, a_2, \dots, a_n$. Ваша задача посчитать количество таких неотрицательных целых чисел x , что x и x^2 встречаются в массиве a .

Формат входных данных

В первой строке входных данных содержится одно целое число n ($1 \leq n \leq 10^5$) — длина массива.
Во второй строке входных данных содержится n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
5 1 4 9 2 16	3
5 2 4 4 2 2	1
13 2 2 1 1 1 4 15 17 15 225 225 16 16	4

Замечание

В первом примере из условия подходят числа 1, 2 и 4, так как в массиве встречаются эти числа и 1, 4 и 16 соответственно.

Во втором примере из условия подходит только число 2, так как в массиве встречаются числа 2 и 4.

В третьем примере из условия подходят числа 1, 2, 4 и 15, так как в массиве встречаются эти числа и 1, 4, 16 и 225 соответственно.

Задача I. Червячки

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

Вам даны два массива целых чисел $a_0, a_1, \dots, a_{n-1}$ и $b_0, b_1, \dots, b_{n-1}$. Гарантируется, что массив b — это переупорядоченный массив a .

Вы можете выполнять следующую операцию не более n раз (возможно, ноль):

- Поменять любые два элемента массива b местами. Другими словами, вы можете выбрать два индекса $0 \leq i < j < n$ и поменять b_i и b_j местами.

Выведите последовательность таких операций, чтобы после их применения массив b был равен массиву a . Обратите внимание, что вам **не нужно** минимизировать количество операций.

Можно показать, что всегда возможно сделать массив b равным массиву a не более чем за n операций при заданных ограничениях.

Формат входных данных

В первой строке входных данных содержится одно целое число n ($2 \leq n \leq 100$) — размер массивов a и b .

Во второй строке входных данных содержится n целых чисел $a_0, a_1, \dots, a_{n-1}$ ($1 \leq a_i \leq 10^9$).

В третьей строке входных данных содержится n целых чисел $b_0, b_1, \dots, b_{n-1}$ ($1 \leq b_i \leq 10^9$).

Гарантируется, что массив b — это переупорядоченный массив a .

Формат выходных данных

В первой строке выходных данных выведете количество операций, которое вам потребуется сделать, чтобы сделать массив b равным массиву a .

В последующих строках выведите сами операции в виде двух чисел i и j через пробел, где i и j — индексы переставляемых элементов. Обращаем ваше внимание, что $0 \leq i < j < n$.

Если существует несколько вариантов ответа, вы можете вывести любой.

Примеры

стандартный ввод	стандартный вывод
2 1 2 2 1	1 0 1
5 1 2 3 4 5 5 4 3 2 1	2 0 4 1 3
5 1 2 3 4 5 1 2 3 4 5	0
7 1 2 1 2 1 2 1 2 2 2 1 1 1 1	3 0 3 2 4 4 5

Замечание

Рассмотрим второй пример из условия:

- Из массива b можно получить массив a за две операции, сначала поменяв местами элементы с индексами 0 и 4, а потом 1 и 3. После первой операции массив b станет равен $[1, 4, 3, 2, 5]$, а после второй $[1, 2, 3, 4, 5]$, что равно массиву a .

Задача J. Количество делителей

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

Вам даны два взаимно простых числа x и y . Выведите количество делителей числа $x \cdot y$.
Два числа являются взаимно простыми, если у них нет общих делителей, кроме единицы.

Формат входных данных

В единственной строке входных данных содержится два целых числа x и y ($1 \leq x, y \leq 10^9$).
Гарантируется, что числа x и y взаимно просты.

Формат выходных данных

Выведите одно целое число — количество делителей числа $x \cdot y$.

Примеры

стандартный ввод	стандартный вывод
2 3	4
5 12	12
999999999 1000000000	2000

Замечание

В первом примере из условия $2 \cdot 3 = 6$. У числа 6 есть четыре делителя: 1, 2, 3 и 6.

Во втором примере из условия $5 \cdot 12 = 60$. У числа 60 есть двенадцать делителей: 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30 и 60.

Задача К. Системы счисления

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

Есть два числа a и b . Число a вам дано в системе счисления с основанием c , а число b в системе счисления с основанием d . В этой задаче мы рассматриваем только системы счисления с основанием 2, 4, 8 или 16.

Вам также дана система счисления e . Вам нужно вывести сумму чисел a и b в системе счисления с основанием e и произведение чисел a и b в системе счисления с основанием e .

Напомним, что:

- В системе счисления с основанием 2 используются только цифры 0 и 1.
- В системе счисления с основанием 4 используются только цифры от 0 до 3.
- В системе счисления с основанием 8 используются только цифры от 0 до 7.
- В системе счисления с основанием 16 используются только цифры от 0 до 9, а также символы A, B, C, D, E и F, которые соответственно равны цифрам от 10 до 15.

Формат входных данных

В первой строке входных данных содержится строка a' и целое число c . Строка a' представляет собой корректную запись числа a в системе счисления с основанием c .

Во второй строке входных данных содержится строка b' и целое число d . Строка b' представляет собой корректную запись числа b в системе счисления с основанием d .

В третьей строке входных данных содержится единственное целое число e . Гарантируется, что системы счисления c , d и e являются либо двоичными, либо четверичными, либо восьмиричными, либо шестнадцатеричными.

Гарантируется, что длина строк a' и b' не больше пяти. Обратите внимание, что запись чисел может содержать лидирующие нули.

Формат выходных данных

В первой строке выходных данных выведите сумму чисел a и b в системе счисления с основанием e , а во второй — произведение чисел a и b в системе счисления с основанием e .

Обратите внимание, что вам **нельзя** выводить лидирующие нули.

Примеры

стандартный ввод	стандартный вывод
10 2 11 2 4	11 12
31 4 1ADF 16 8	15354 256523
FFFFF 16 EFEFE 16 16	1EFEFD EFEFD10102
00000 4 00000 8 16	0 0
ABCDE 16 FEDC 16 2	10111011101110111010 101010110000100111101001001011001000

Замечание

Рассмотрим первый пример из условия:

- $a = 10_2 = 1 \cdot 2^1 + 0 \cdot 2^0 = 2$.
- $b = 11_2 = 1 \cdot 2^1 + 1 \cdot 2^0 = 3$.
- $a + b = 2 + 3 = 5$ и $a \cdot b = 6$. В четверичной системе счисления $5 = 11_4$ и $6 = 12_4$.

Рассмотрим сумму чисел a и b из второго примера из условия:

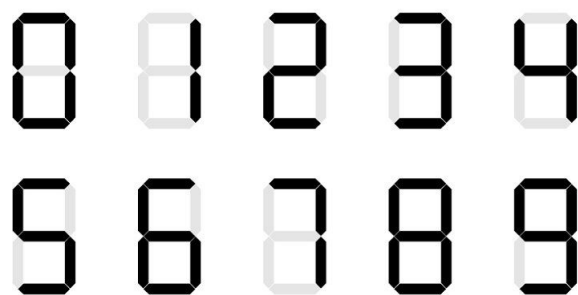
- $a = 31_4 = 3 \cdot 4^1 + 1 \cdot 4^0 = 13$.
- $b = 1ADF_{16} = 1 \cdot 16^3 + 10 \cdot 16^2 + 13 \cdot 16^1 + 15 \cdot 16^0 = 6879$.
- $a + b = 13 + 6879 = 6892$. В восьмеричной системе счисления $6892 = 15354_8$.

Задача L. Электронные часы

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

Поликарп проснулся рано утром и, по своему обыкновению, посмотрел на свои настольные часы, чтобы узнать, сколько времени. Однако, к глубокому разочарованию нашего героя, выяснилось, что его сосед по комнате положил свои вещи на стол перед часами, тем самым частично ограничивая видимость циферблата для Поликарпа. Встать с кровати было лень, поэтому Поликарп задался вопросом — сколько различных валидных значений времени могут показывать часы в текущий момент, исходя из видимой для Поликарпа части циферблата? Вычислять это ему тоже было лень, поэтому он попросил вас помочь ему с этим.

Часы Поликарпа показывают время в формате `hh:mm`, то есть на циферблате изображено два двузначных числа, разделённых двоеточием — число часов и число минут. На каждой позиции для цифры имеется 7 независимых сегментов. Каждый из них может гореть или не гореть. Цифры отображаются с помощью некоторых комбинаций горящих сегментов. Вот как на часах отображаются цифры от 0 до 9:



В этой задаче каждый сегмент каждой цифры может либо гореть, либо не гореть, либо быть скрыт от зрения Поликарпа (то есть его состояние неизвестно).

Возможно такое, что горящий набор сегментов одной позиции, видимый Поликарпу, не соответствует ни одной цифре. Это означает, что часы сломаны, и в ответе будет 0 валидных значений времени.

Обратите внимание, что на часах Поликарпа первая цифра (цифра десятков часов) всегда пишется! То есть время «полвторого ночи» будет на часах выглядеть как 01:30. **Отсутствие на циферблате цифры десятков часов считается невалидной конфигурацией!**

Формат входных данных

Каждая позиция для цифры на циферблате во входных данных представлена матрицей размера 9×6 , состоящей из символов `.`, `1` и `X`. Вот так выглядит матрица для цифры 8:

```
.1111.  
1....1  
1....1  
1....1  
.1111.  
1....1  
1....1  
1....1  
.1111.
```

Формально, в индексации с единицы вертикальные сегменты начинаются во 2-й, а заканчиваются в 4-й, а также начинаются в 6-й, а заканчиваются в 8-й строчках 1-го и 6-го столбцов, а горизонтальные — начинаются во 2-м, а заканчиваются в 5-м столбцах 1-й, 5-й и 9-й строк.

Если какой-то сегмент не горит, то на его месте в матрице будут символы `.` вместо единиц. Например, вот как выглядит на циферблате позиция, на которой не горит ни один сегмент:

```
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....
```

Если какой-то сегмент скрыт от зрения Поликарпа, то на его месте в матрице будут символы `X` вместо единиц. Например, вот как выглядит на циферблате позиция, на которой все сегменты скрыты от зрения Поликарпа:

```
.XXXX.  
X...X  
X...X  
X...X  
.XXXX.  
X...X  
X...X  
X...X  
.XXXX.
```

Матрицы, соответствующие отдельным позициям на циферблате, во входных данных объединены в общую матрицу размера 9×29 , задающую весь циферблат. В этой общей матрице позиции, соответствующие десяткам и единицам часов, а также десяткам и единицам минут, разделены столбиком из точек. Число часов и число минут разделены тремя столбцами, содержащими точки, а также два символа `*` по центру (формально, в индексации с единицы — в 4-й и 6-й строках 15-го столбца), символизирующих двоеточие. Вот, например, как отображается время 18:08 в описанном формате:

```
.....1111.....1111...1111.  
.....1.1...1...1...1.1...1  
.....1.1...1...1...1.1...1  
.....1.1...1*.1...1.1...1  
.....1111.....1111.  
.....1.1...1*.1...1.1...1  
.....1.1...1...1...1.1...1  
.....1.1...1...1...1.1...1  
.....1111.....1111...1111.
```

Входные данные представляют собой 9 строк, в каждой по 29 столбцов, содержащие циферблат в описанном выше формате.

Гарантируется, что ни один сегмент не горит частично. То есть каждый сегмент либо полностью горит, либо полностью не горит, либо полностью скрыт от взора Поликарпа.

Формат выходных данных

В первой строке выведите одно целое число k — количество валидных моментов времени, которые могут показывать часы Поликарпа в заданных условиях. В следующих k строках выведите

сами валидные моменты времени в формате **hh:mm**, упорядоченные лексикографически, по одному в строке. Если часы Поликарпа сломаны, то просто выведите 0 в единственной строке.

Примеры

стандартный ввод	стандартный вывод
.1111.....1111...1111. 1.....1.....1.1....1 1.....1.....1.1....1 1.....1.*.....1.1....1 .1111.....1111...XXXX. X....X.....1.*.....1.1....1 X....X.....1.....1.1....1 X....X.....1.....1.1....1 .XXXX.....1111...1111.	2 21:30 21:38
.1111..... 1....1.1....1.....1.1....1 1....1.1....1.....1.1....1 1....1.1....1.*.....1.1....1 1111..... 1....1.....1.*.....1.X....X 1....1.....1.....1.X....X 1....1.....1.....1.X....X .1111.....XXXX.	0
.....*.....*.....	0
.1111...1111....1111...1111. 1....1.1....1...1....1.1....1 1....1.1....1...1....1.1....1 1....1.1....1.*.1....1.1....1 .1111...1111....1111...1111. 1....1.1....1.*.1....1.1....1 1....1.1....1...1....1.1....1 1....1.1....1...1....1.1....1 .1111...1111....1111...1111.	0

Замечание

В первом примере цифры единиц часов и десятков минут определены однозначно. На месте цифры единиц минут может быть как цифра 0, так и цифра 8. Комбинация известных Поликарпу горящих сегментов на позиции десятков часов подходит и для цифры 2, и для цифры 3, но цифра 3 не может стоять на этой позиции, так как в сутках всего 24 часа.

Во втором примере комбинация видимых Поликарпу сегментов единиц минут не может быть частью ни одной цифры, следовательно, часы сломаны.

Задача М. Закупаемся в магазине

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

В магазине проводится акция. Вы можете купить несколько товаров, и самый дорогой достанется вам бесплатно! Но дело в том, что это очень необычный магазин, поэтому вы можете купить только непрерывный отрезок товаров.

Формально, товары представлены как массив a длины n из элементов неотрицательной стоимости. Стоимость i -го товара равна a_i рублей. Вы можете выбрать любой непрерывный подотрезок и купить все товары на нём за сумму их стоимостей, не платя за самый дорогой.

Какое максимальное количество товаров вы можете купить, если вы можете потратить **не более** s рублей.

Формат входных данных

В первой строке входных данных содержится два целых числа n и s ($1 \leq n \leq 10^5$, $0 \leq s \leq 10^{14}$) — количество товаров в магазине и количество имеющихся у вас денег соответственно.

Во второй строке входных данных содержится n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$) — стоимости товаров.

Формат выходных данных

Выведите одно целое число — максимальное количество товаров, которое вы можете купить.

Примеры

стандартный ввод	стандартный вывод
1 9331054993288 807670057	1
6 5 2 6 1 1 4 3	4
5 0 0 5 0 0 1	4
10 20 1 10 5 6 2 4 3 8 7 9	6

Замечание

В первом примере из условия всего один товар. Один товар мы всегда можем купить бесплатно.

Во втором примере из условия можно купить четыре последних товара, потратив 5 рублей.

В третьем примере из условия можно купить первые четыре товара, потратив 0 рублей, так как самый дорогой товар нам достанется бесплатно.

Задача N. Дивергенция

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

У вас есть граф с $2^{n+1} - 1$ вершинами. Граф является полным двоичным деревом — другими словами, между вершинами с номерами i и $\lfloor \frac{i}{2} \rfloor$ для всех i от 2 до $2^{n+1} - 1$ проведено ребро. На каждом ребре лежит так называемый D-mail с мощностью 0, 1 или 2.

Если вкратце, то D-mail с мощностью x позволит вам переместиться во времени из $(11 - x)$ -го октября на один день назад, в $(11 - x - 1)$ -е октября. Например, имея хотя бы один D-mail с мощностью 0, вы сможете переместиться в 10-е октября, а, имея D-mail-ы с мощностями 0 и 1, можно попасть аж в 9-е октября, в то время как один D-mail с мощностью 1 сам по себе не позволит вам никуда переместиться.

Чтобы успеть подготовиться к отбору на ЧеРеКОШ, вы хотите переместиться из 11-го октября в прошлое. Для этого вы можете один раз выбрать любые две вершины в дереве, забрать все D-mail-ы, лежащие на кратчайшем пути между этими вершинами, и использовать их по своему усмотрению в любом порядке.

Вы хотите выбрать вершины таким образом, чтобы полученные D-mail-ы позволили вам переместиться как можно дальше в прошлое. Если таких способов взять вершины много, то вам нужно выбрать из них вариант, при котором расстояние между взятыми вершинами минимально. Определите, как это сделать.

Формат входных данных

В первой строке входных данных содержится одно целое число n ($1 \leq n \leq 18$) — глубина дерева.

Во второй строке входных данных содержится $(2^{n+1} - 2)$ целых чисел $a_2, a_3, a_4, \dots, a_{2^{n+1}-1}$ ($0 \leq a_i \leq 2$), где a_i — мощность D-mail-а, лежащего на ребре между вершинами i и $\lfloor \frac{i}{2} \rfloor$.

Формат выходных данных

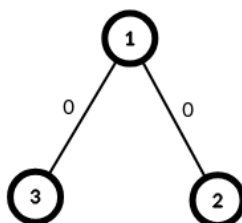
В единственной строке выходных данных выведите два целых числа через пробел — на сколько дней в прошлое вы сможете переместиться и кратчайшее расстояние между вершинами, которые вы можете для этого выбрать.

Примеры

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

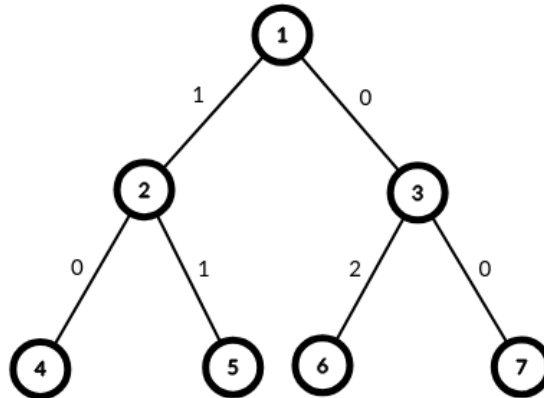
Замечание

В первом примере из условия граф выглядит следующим образом:



Чтобы переместиться в 10-е октября, можно выбрать пару вершин $(1, 2)$, $(1, 3)$ или $(2, 3)$. Для пар $(1, 2)$ и $(1, 3)$ расстояние между вершинами равно 1, для $(2, 3)$ — 2, поэтому ответ — 1 1.

Во втором примере из условия граф выглядит следующим образом:



Чтобы переместиться на три дня, можно выбрать пару вершин $(4, 6)$, $(5, 6)$ или $(2, 6)$. Кратчайшее расстояние между вершинами имеет пара $(2, 6)$.

В третьем примере из условия мы не сможем переместиться в прошлое ни на один день, как бы мы ни выбирали пару вершин. Чтобы получить кратчайшее расстояние 0, достаточно выбрать пару из двух одинаковых вершин: $(1, 1)$, $(2, 2)$ или $(3, 3)$.

Задача О. Побег из графа

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

Петя ходит по ориентированному графу, состоящему из n вершин и m рёбер, и несёт в кармане неотрицательное целое число рублей. Если до прохода по ребру у Пети x рублей, то за проход возьмётся плата в размере $\lfloor \sqrt{x} \rfloor$ рублей, где записью $\lfloor a \rfloor$ обозначается целая часть числа a .

В некоторых вершинах графа есть выходы: находясь в вершине e и потратив c_e рублей, Петя может закончить движение и покинуть граф. Для каждой из вершин графа определите минимальную сумму денег, стартовав с которой, Петя сможет добраться от этой вершины до какого-нибудь из выходов и покинуть граф.

Формат входных данных

В первой строке входных данных содержится три целых числа n , m и t ($1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$, $0 \leq t \leq n$) — количество вершин и рёбер в графе, а также количество выходов.

В следующих m строках входных данных содержатся по два целых числа u и v ($1 \leq u, v \leq n$), означающих, что существует ребро из вершины u в вершину v . Гарантируется, что в графе нет петель и кратных рёбер.

В следующих t строках входных данных содержатся по два целых числа e и c_e ($1 \leq e \leq n$, $0 \leq c_e \leq 10^9$), означающих, что в вершине e находится выход, требующий платы c_e рублей. Гарантируется, что в каждой вершине находится не более одного выхода.

Формат выходных данных

Выведите n целых чисел на одной строке — минимальные суммы, нужные для покидания графа, начиная с каждой из его вершин, в порядке возрастания номеров вершин. Если из некоторой вершины покинуть граф невозможно, выведите для неё -1 .

Примеры

стандартный ввод	стандартный вывод
5 4 3 3 2 3 4 5 3 5 4 1 1 2 10 3 100	1 10 13 -1 17
4 4 2 1 2 2 3 3 1 1 4 1 0 2 1	0 0 0 -1

Замечание

В первом примере из условия:

- Вершины 1 и 2: Петя сразу же покидает граф, не переходя ни по одному ребру. Это будет стоить Пете 1 и 10 рублей соответственно.

- Вершина 3: имея 13 рублей, Петя пройдет по ребру в вершину 2 (заплатив 3 рубля), и за 10 рублей покинет граф через неё. Использовать выход, расположенный в самой вершине 3, будет невыгодно.
- Вершина 4: из неё невозможно добраться ни до каких выходов.
- Вершина 5: имея 17 рублей, Петя пройдет по ребру в вершину 3 (заплатив 4 рубля), затем пройдет в вершину 2 (заплатив 3 рубля), и за 10 рублей покинет граф через вершину 2.

Во втором примере из условия:

- Имея 0 рублей, Петя может двигаться по рёбрам без оплаты. Из вершин 2 и 3 он попадёт в вершину 1 и покинет граф бесплатно.

Задача Р. Лексикографическая минимизация

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 512 мегабайт

Вам даны два массива целых чисел $a_1, a_2, \dots, a_n$ и $b_1, b_2, \dots, b_m$.

Над массивом a выполняется m операций обновления. На j -м шаге выбирается некоторый индекс $1 \leq i \leq n$ и значение a_i заменяется на $a_i \oplus b_j$. Обратите внимание, что каждый элемент b_j используется ровно один раз, а выбор индекса i на каждом шаге может быть произвольным.

Ваша задача состоит в том, чтобы выбрать такую последовательность индексов $i_1, i_2, \dots, i_m$, чтобы после выполнения всех m операций итоговый массив a был лексикографически минимальным среди всех возможных массивов, достижимых описанным процессом.

Напомним, что массив x длины n лексикографически меньше, чем массив y длины n , если существует такой индекс $1 \leq i \leq n$, что $x_i < y_i$, и для всех индексов $1 \leq j < i$ выполнено $x_j = y_j$.

За $\oplus$ обозначена битовая операция XOR. Операция XOR двух чисел a и b — это битовая операция, которая применяется независимо к каждой паре соответствующих битов a и b . Эта операция представляет собой сложение по модулю 2. Вот её таблица истинности для пары битов:

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Формат входных данных

В первой строке входных данных содержится одно целое число n ($1 \leq n \leq 16$) — длина массива a .

Во второй строке входных данных содержится n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$), где a_i — i -й элемент массива a .

В третьей строке входных данных содержится одно целое число m ($1 \leq m \leq 16$) — длина массива b .

В четвертой строке входных данных содержится m целых чисел $b_1, b_2, \dots, b_m$ ($0 \leq b_i \leq 10^9$), где b_i — i -й элемент массива b .

Формат выходных данных

В единственной строке выходных данных выведите n целых чисел — лексикографически минимально возможный массив a .

Примеры

стандартный ввод	стандартный вывод
2 1 1 2 1 1	0 0
3 7 6 9 5 3 1 15 10 12	0 5 6

Замечание

Во втором примере из условия значение i на каждом из пяти шагов будет таким: 2, 1, 3, 1, 1. Тогда массив a будет меняться следующим образом:

$$[7, 6, 9] \rightarrow [7, 5, 9] \rightarrow [6, 5, 9] \rightarrow [6, 5, 6] \rightarrow [12, 5, 6] \rightarrow [0, 5, 6]$$