

УСЛОВИЯ ЗАДАЧ ОСНОВНОГО ТУРА

Задача А. Числовые промежутки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Сегодня Вася узнал про числовые промежутки. Каждый из них задаётся парой чисел — своими началом и концом, и информацией о том, включается ли в него каждый из концов. Таким образом, существует четыре типа промежутков:

- **Интервал.** Обозначается (x, y) , включает в себя все числа z : $x < z < y$.
- **Полуинтервалы.** Обозначаются $[x, y)$ и $(x, y]$, включают в себя все такие z , что $x \leq z < y$ и $x < z \leq y$ соответственно.
- **Отрезок.** Обозначается $[x, y]$ и включает в себя все числа z : $x \leq z \leq y$.

Васе захотелось посчитать количество целых чисел в каждом из заданных промежутков. Поскольку он не знает вещественных чисел, x и y — рациональные: $x = \frac{a}{b}$, $y = \frac{c}{d}$ (a и c целые, b и d целые положительные).

Рассмотрим пример: $[\frac{3}{2}, 4)$. В данном случае $d = 1$, поэтому вместо $\frac{4}{1}$ пишут просто 4. В этом множестве содержится два целых числа: 2 и 3, а число 4 не содержится.

Помогите Васе — напишите программу, которая по заданному числовому промежутку посчитает количество целых чисел, содержащихся в нём.

Формат входного файла

Первым символом идёт открывающаяся квадратная или круглая скобка. Далее записано число x в формате a/b либо a , где $|a| \leq 10^9$, $0 < b \leq 10^9$. После следует запятая и пробел. Потом — число y в таком же формате. Далее — закрывающаяся квадратная или круглая скобка. После неё идёт перевод строки.

Гарантируется, что данный числовой промежуток не является пустым (то есть содержит в себе хотя бы одно число, не обязательно целое).

Формат выходного файла

По заданному числовому промежутку выведите единственное число — количество целых чисел в нём.

Примеры

стандартный ввод	стандартный вывод
$[3/2, 4)$	2
$[-2/4, 5/3]$	2
$[-1000, 1000]$	2001
$[-2, 4/3]$	4

Задача В. Делёж яблок

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Этим летом у бабушки был большой урожай яблок. Она собрала яблоки в корзину и отдала своим K внукам.

Первый внук взял из корзины половину всех яблок и еще a_1 яблоко (если количество яблок не делилось на два, то результат деления на два он мог округлить как в большую сторону, так и в меньшую). К примеру, если в корзине было 7 яблок и $a_1 = 1$, то он мог взять либо 4, либо 5, а если было 6 яблок и $a_1 = 1$, то он взял ровно 4.

Второй внук взял половину от всех оставшихся яблок и ещё a_2 (если яблок было нечетное количество, то он также мог округлить половину как в большую, так и в меньшую сторону). И так далее, K -ый внук взял половину яблок, оставшихся после $K - 1$ внука, и ещё a_k . В итоге в корзине ничего не осталось.

Теперь они задумались, насколько же большой урожай был у бабушки. Ни один из них не помнит, делилось ли количество яблок на 2 нацело при его выборе, а если нет, то в какую сторону он округлил половину яблок. Внуков интересует минимальное и максимальное изначальное количество яблок в корзине, при которых могли произойти описанные события.

Формат входного файла

В первой строке входного файла содержится целое положительное число K ($1 \leq K \leq 1000$). Во второй строке записано K целых неотрицательных чисел $a_1, \dots, a_K$ ($0 \leq a_i \leq 1000$).

Формат выходного файла

Выведите два неотрицательных целых числа без ведущих нулей, каждое в новой строке — минимальное и максимальное возможное количество яблок в корзине соответственно.

Примеры

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

Задача С. Вася и его друзья

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В новой игре “Closed Loops 7” игрокам предлагается клетчатая таблица N на M клеток. Ход состоит в том, что очередной игрок рисует *цикл* — замкнутую линию без самопересечений, идущую только по сторонам клеток. Каждый цикл можно нарисовать только один раз за всю игру (при этом, конечно, не запрещается рисовать циклы, пересекающиеся с уже нарисованными). Игроки ходят по очереди. Выигрывает тот, кто рисует последний возможный цикл. К примеру, если $N = 2$, $M = 1$, то циклов всего три и игрок, делающий третий ход, выигрывает:



Вася позвал $K - 1$ друзей поиграть с ним. Чтобы произвести впечатление, он непременно хочет выиграть. Для этого ему нужно узнать, каким по счету игроком он должен быть, чтобы гарантированно одержать победу. Вася наслышан о ваших успехах в программировании, и за помощью он обратился именно к вам.

Формат входного файла

В единственной строке входного файла находятся три целых числа: N, M — размеры таблицы ($1 \leq N \leq 100, 1 \leq M \leq 8$) и K — количество игроков ($1 < K \leq 10^9$).

Формат выходного файла

Выведите одно число от 1 до K — каким по счету игроком должен быть Вася, чтобы выиграть.

Примеры

стандартный ввод	стандартный вывод
2 1 2	1
1 8 8064	36

Задача D. Числа

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Саша и Катя учатся в начальной школе. Для изучения арифметики при этом используются карточки, на которых написаны цифры (на каждой карточке написана ровно одна цифра). Однажды они пришли на урок математики, и Саша, используя все свои карточки, показал число A , а Катя показала число B . Учитель тогда захотел дать им такую задачу, чтобы ответ на нее смогли показать и Саша, и Катя, каждый используя только свои карточки. При этом учитель хочет, чтобы искомое число было максимально возможным.

Формат входного файла

Во входном файле записано два целых неотрицательных числа A и B (каждое число в одной строке). Длина каждого из чисел не превосходит 100 000 цифр.

Формат выходного файла

Выведите одно число — максимальное целое число, которое можно составить используя как цифры первого числа, так и цифры второго числа. Если же ни одного такого числа составить нельзя, выведите -1.

Примеры

стандартный ввод	стандартный вывод
280138 798081	8810
123 456	-1

Задача Е. Питон

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Витя изучает новый язык программирования Питон. Пока он только успел изучить арифметические операции и условную инструкцию `if`, но он уже полюбил этот язык за красоту и лаконичность синтаксиса.

Отличительной особенностью языка Питон является то, что блоки после инструкций `if` и `else` (а также в циклах `for` и `while`, но Витя еще не успел изучить циклы) выделяются не ключевыми словами (например, в языке Паскаль используются слова `begin` и `end`) и не скобками (например, в языке С используются фигурные скобки), а величиной отступа от начала строки, то есть количеством пробелов, которые идут в начале строки. Например, в такой программе:

```
if a < 0:
    print("Число а - отрицательное")
    a = -a
print("Теперь а - положительное")
```

если условие `a < 0` будет истинно, то выполнятся две следующие строки `print("Число а - отрицательное")` и `a = -a`, а вот следующая строка `print("Теперь а - положительное")` уже находится вне блока условной инструкции `if` и будет выполнена после этой условной инструкции независимо от истинности проверенного условия.

Более формально правила расстановки пробелов в программе такие. Первая строка программы, а также все инструкции в программе, если они не находятся внутри блоков условных инструкций, не содержат отступа, то есть пробелов в начале строки. Если в программе встречается условная инструкция `if`, то блок после этой инструкции пишется с отступом. Величина отступа может быть произвольной (1, 2, 3 и более пробелов), но для всех инструкций внутри блока отступ должен быть одинаковым. Если после инструкции `if` идет инструкция `else`, то она должна иметь такой же отступ, что и соответствующая ей инструкция `if`, после инструкции `else` идет блок из одной и более инструкций с дополнительным отступом. При этом отступ у блока `if` и блока соответствующего ему `else` может быть различным (смотрите примеры верных программ ниже), но внутри одного блока отступ должен быть одинаковым.

Каждой инструкции `if` может соответствовать не более одной инструкции `else`. Не допускаются инструкции `else`, перед которыми нет инструкции `if`. После каждой инструкции `if` и `else` обязательно следует хотя бы одна инструкция с отступом.

Правильно (разрешается разный отступ в блоках <code>if</code> и <code>else</code>)	Правильно (вложенные условные инструкции)	Неправильно (разный отступ в одном блоке)	Неправильно (нет блока с отступом после инструкции <code>if</code>)
<pre>if x > 0: print("x > 0") print(x) else: print("x < 0") print(-x) print("Bye")</pre>	<pre>if a > b: if a > c: print(a) else: print(c) else: if b > c: print(b) else: print(c)</pre>	<pre>if x > 0: print(x) print("x > 0")</pre>	<pre>if x > 0: else: print(x)</pre>

Также допускаются вложенные условные инструкции, у блоков вложенных условных инструкций отступ должен быть большим, чем у объемлющей инструкции, но при этом может быть произвольным.

Витя хочет написать компилятор языка Питон, и для начала он решил реализовать анализатор корректности расстановки отступов в условных инструкциях. Помогите ему в решении этой задачи.

Формат входного файла

Во входном файле записан некоторый текст, содержащий не более 100 строк. Длина каждой строки не превосходит 100 символов. Каждая строка состоит из символов, ASCII-коды которых не менее 32 и не более 126.

Строка считается инструкцией `if`, если первыми непобельными символами строки является слово `if`, после которого идет пробел, а затем — любое число любых символов. Строка считается инструкцией `else`, если она содержит только одно слово `else:` (с двоеточием после него) и, возможно, отступ в начале строки.

Любая строка содержит хотя бы один непобельный символ. Последняя строка программы обязательно содержит ровно одно слово `exit()` без пробелов, завершающееся символом конца строки.

Формат выходного файла

Если отступы в этой программе расставлены правильно, то программа должна вывести одно число 0. Если отступы расставлены неправильно, то нужно вывести минимальный номер строки, в которой нарушаются правила расстановки отступов.

Примеры

стандартный ввод	стандартный вывод
<pre>a, b, c = map(int, input().split()) if a > b: if a > c: print(a) else: print(c) else: if b > c: print(b) else: print(c) exit()</pre>	0
<pre>x = int(input()) if x < 0: print("Negative") x = -x else: print("Positive") exit()</pre>	4

Задача F. Как стать призером

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

К тренеру на занятия по подготовке к олимпиаде ходит N участников. Для каждого из них заданы два параметра: начальный условный опыт A_i и условный интеллект B_i .

Каждое занятие устроено так: тренер подходит к какому-нибудь участнику и обсуждает с ним возникшие вопросы и проблемы. В результате такого обсуждения условный опыт этого участника возрастает на B_i (то есть чем выше условный интеллект участника, тем больше он может взять из общения с тренером).

За все время подготовки к олимпиаде тренер может подойти ко всем участникам суммарно не более C раз (он может подходить к разным участникам, может несколько раз подходить к одному и тому же). Для того, чтобы участник стал призером олимпиады, к началу олимпиады его условный опыт должен быть не меньше, чем K .

Напишите программу, которая вычислит максимальное количество призеров олимпиады, которое сможет подготовить тренер.

Формат входного файла

Сначала вводятся натуральные числа N , C , K , задающие количество участников, количество подходов, которые может сделать тренер, и условный опыт, необходимый, чтобы стать призером олимпиады, соответственно ($1 \leq N \leq 10^6$, $1 \leq C \leq 10^9$, $1 \leq K \leq 10^9$). Далее идут N пар целых неотрицательных чисел A_i , B_i , задающих начальный условный опыт и условный интеллект каждого участника. Каждое из чисел A_i и B_i не превышает 10^9 .

Формат выходного файла

Выведите одно число — наибольшее количество призеров олимпиады, которое успеет подготовить тренер.

Примеры

стандартный ввод	стандартный вывод
3 5 6 1 1 2 1 4 2	2
4 10 3 0 1 0 1 0 2 2 0	3

Задача G. Промежуточный контроль

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Группа студентов ИГЭУ написала промежуточный контроль. В результате ровно $A\%$ студентов получили отлично, ровно $B\%$ — хорошо, ровно $C\%$ — удовлетворительно, а остальные $D\%$ написали его на неудовлетворительно. Какое минимальное количество студентов должно быть в группе для того, чтобы могли получиться такие результаты?

Формат входного файла

Вводятся 4 целых числа от 0 до 100 — A, B, C, D ($A + B + C + D = 100$).

Формат выходного файла

Выведите единственное целое положительное число — минимальное возможное количество студентов в группе.

Пример

стандартный ввод	стандартный вывод
40 50 5 5	20

Задача Н. К-квест

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

В одной из компьютерных игр-квестов есть следующее задание. На карте игрового мира размещены N персонажей, с каждым из которых может встретиться игрок. От общения с i -м персонажем карма игрока меняется на величину a_i , которая может быть как положительной, так отрицательной или даже нулем.

Изначально карма игрока равна нулю. Для того чтобы пройти на следующий уровень, нужно чтобы карма была в точности равна значению K , при этом карма также может принимать как положительные, так и отрицательные значения.

Комнаты, в которых находятся персонажи, соединены односторонними магическими порталами, поэтому игроку придется встречать персонажей в определенной последовательности: после персонажа номер i он попадает к персонажу номер $i + 1$, затем к персонажу номер $i + 2$, и т.д. В комнате последнего персонажа с номером N портала к другому персонажу нет.

Для перемещения между персонажами можно использовать еще и заклинания телепортации, но к сожалению у героя осталось всего лишь два свитка с заклинаниями. Поэтому один из этих свитков придется использовать для того, чтобы телепортироваться к любому из персонажей, а второй свиток — чтобы покинуть игровой мир, после того, как карма героя станет равна K .

Помогите игроку определить, в какую комнату надо телепортироваться в начале и из какой комнаты нужно покинуть игровой мир, чтобы достичь кармы K или сообщите, что это невозможно.

Формат входного файла

В первой строке входных данных записаны два числа: количество персонажей N и необходимый уровень кармы K ($|K| \leq 10^9$, $K \neq 0$). Во второй строке через пробел записаны N целых чисел $a_1, a_2, \dots, a_N$ — величины, на которые меняется карма героя после общения с персонажами с номерами $1, 2, \dots, N$ соответственно.

Формат выходного файла

Выведите номер комнаты, в которую надо войти игроку и номер комнаты, из которой надо выйти, чтобы набрать карму K . Если возможных вариантов несколько, то необходимо вывести самый короткий путь, а если и таких несколько, то путь, начинающийся в комнате с как можно большим номером. Если достичь кармы K последовательно общаясь с персонажами невозможно, то выведите одно число -1 .

Примеры

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

Задача I. Магия числа 23

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

По заданному числу N найдите натуральное число K , такое что:

- число $\overline{KK}$ (повторённая два раза десятичная запись K) является точным квадратом некоторого натурального числа (см. примеры),
- K при записи в десятичной системе счисления имеет длину от N до $N + 23$ (включительно).

Так, для $N = 1$ условию удовлетворяет, например, число $K = 13223140496$, т.к. оно имеет длину 11, что укладывается в диапазон от 1 до 24, а также число 1322314049613223140496 является точным квадратом натурального числа.

Формат входного файла

Вводится одно натуральное число N ($1 \leq N \leq 2323$).

Формат выходного файла

Выведите искомое число K . Если чисел, удовлетворяющих условию, несколько, выведите любое из них. Если таких чисел не существует, выведите 0.

Примеры

стандартный ввод	стандартный вывод
1	13223140496
11	13223140496
10	29752066116
39	715976331360946745562130177514792899409

Задача J. Доставка на дом

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Фирма по грузоперевозкам привезла к воротам загородного дома заказанный домашний кинотеатр в очень большой кубической коробке размерами $1 \times 1 \times 1$ метр. Так как машину на территорию участка не пустили, коробка была сброшена у ворот. Одна из ее сторон (имеется в виду грань куба) помечена как хрупкая — та, рядом с которой расположен экран. Коробка выгружена так, что хрупкая сторона не находится на земле.

Из-за огромных размеров коробки по участку её можно передвигать только перекатывая через ребра. При этом хрупкая сторона не должна оказаться на земле, иначе экран немедленно сломается.

Участок имеет форму прямоугольника размером N на M метров. План участка нарисован на клетчатой бумаге, размер клетки которой соответствует 1 метру. На плане введена система координат так, что левая нижняя клетка плана имеет координату $(1, 1)$, правая нижняя — $(1, M)$, правая верхняя — (N, M) .

Изначально коробка расположена рядом с воротами, в клетке, которая на плане имеет координаты $(1, b)$ (эта клетка расположена у нижней стороны плана участка), а переместить ее надо к двери — на другую клетку с координатами (c, d) . Задано, с какой стороны исходно находится хрупкая сторона. С какой стороны она будет после перекатываний — не важно (важно лишь, чтобы она не оказалась на земле). Участок окружён по периметру забором, поэтому коробку не получится выкатить за пределы участка.

Ваша задача — помочь грузчикам перекатить коробку от ворот до двери дома, не поломав экрана.

Формат входного файла

В первой строке вводятся целые числа N, M, b, c, d ($1 \leq N \leq 10\,000$, $1 \leq M \leq 10\,000$, $1 \leq b \leq M$, $1 \leq c \leq N$, $1 \leq d \leq M$). Во второй строке содержится одна из букв L, R, T, F, B, описывающая начальное положение хрупкой стороны коробки (слева, справа, сверху, спереди и сзади соответственно). Считается, что задняя сторона коробки повернута в сторону ворот. Ворота и дверь на плане изображаются разными клетками.

Формат выходного файла

Выведите последовательность перекатываний, которая позволит грузчикам выполнить поставленную задачу. Перекатывания обозначаются буквами

- L (перекатывание влево — на единицу уменьшается вторая координата),
- R (перекатывание вправо — на единицу увеличивается вторая координата),
- F (перекатывание вперед — на единицу увеличивается первая координата),
- B (перекатывание назад — на единицу уменьшается первая координата).

Общее количество перекатываний не должно превышать $4(M + N)$ — иначе грузчики не возьмется за столь тяжелую работу.

Если это невозможно, выведите IMPOSSIBLE.

Примеры

стандартный ввод	стандартный вывод
4 3 2 3 2 T	RFFL
2 1 1 2 1 F	IMPOSSIBLE