

Программа может читать данные с клавиатуры (стандартного ввода) и выводить результат на экран (стандартный вывод), либо программа может использовать файловый ввод-вывод (имена файлов указаны в условии задачи).

Во время тура каждая задача тестируется на части тестов, так называемая online-группа тестов. На тестах online-группы программа может набрать от 40 до 60 баллов в зависимости от задачи (смотрите условия). Результат проверки решения на online-группе будет доступен во время тура.

После окончания тура последнее сданное решение по каждой задаче будет проверено на тестах offline-группы. Суммарное количество баллов за тесты online-группы и offline-группы равно 100 по каждой задаче.

Всего за тур вы можете отправлять решения на проверку не более 60 раз (суммарно по всем задачам).

## Задача А. Кола

Имя входного файла: `ain.txt`  
Имя выходного файла: `aout.txt`  
Ограничение по времени: 0,2 секунды  
Ограничение по памяти: 64 мегабайта

Завод по производству колы изготавливает ее не только для магазинов, но и для всемирно известной сети ресторанов быстрого питания.

Ежедневно завод отгружает один и тот же объем колы в литрах. Служба доставки сети ресторанов обычно использует для транспортировки колы емкости объемом или только 50 литров, или только 70 литров. Если доставка осуществляется с помощью емкостей в 50 литров, то для перевозки имеющегося объема колы необходимо  $A$  емкостей. А если с помощью емкостей в 70 литров, то необходимо  $B$  емкостей. При этом в каждом из случаев одна из емкостей может быть заполнена не полностью.

Недавно сеть ресторанов решила утвердить новый объем емкостей для доставки колы — 60 литров. Сколько емкостей теперь может понадобиться для доставки того же самого объема колы?

### Формат входного файла

Входные данные содержат 2 числа  $A$  и  $B$ , расположенных каждое в отдельной строке ( $1 \leq A, B \leq 10\,000\,000$ ).

### Формат выходного файла

Выведите все возможные значения для количества емкостей по 60 литров, которые окажутся заполненными (в том числе одна возможно частично), в порядке возрастания или число  $-1$ , если значения  $A$  и  $B$  противоречат друг другу, то есть они были записаны неверно.

### Примеры

<code>ain.txt</code>	<code>aout.txt</code>
3 2	2 3
1 2	-1

В первом примере колы могло быть, например, 115 литров, в этом случае понадобится две емкости в 60 литров, а могло быть — 135 литров, в этом случае понадобятся уже три емкости по 60 литров. Четыре емкости не могут понадобиться никогда.

### Система оценивания

Online-группа тестов оценивается в 60 баллов, в этой группе  $1 \leq A, B \leq 1\,000$ .

Offline-группа тестов оценивается в 40 баллов.

## Задача В. Числа

Имя входного файла: `bin.txt`  
Имя выходного файла: `bout.txt`  
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

<code>bin.txt</code>	<code>bout.txt</code>
280138 798081	8810
123 456	-1

### Система оценивания

Online-группа тестов оценивается в 60 баллов, в этой группе числа  $A$  и  $B$  содержат не более 1000 цифр каждое. При этом решения, правильно работающие для случая, когда  $A$  и  $B$  содержат не более 6 цифр, будут оценены не менее, чем в 20 баллов. Решения, правильно работающие для случая, когда  $A$  и  $B$  содержат не более 9 цифр, будут оценены не менее, чем в 40 баллов.

Offline-группа тестов оценивается в 40 баллов.

## Задача С. Расписание турнира

Имя входного файла: `cin.txt`  
Имя выходного файла: `cout.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Как известно, командные спортивные соревнования часто проводятся по круговой системе, когда любые две команды должны сыграть между собой ровно один матч. Круговой турнир проводится в несколько туров, в одном туре каждая команда может сыграть не более одного матча. Например, если в турнире участвуют 4 команды, то турнир можно провести в три тура: в первом туре команда 1 играет с командой 2, а команда 3 играет с командой 4, во втором туре 1 играет с 3, а 2 играет с 4, в третьем туре — 1 играет с 4, а 2 играет с 3.

Организаторам олимпиады Сочи-2014 необходимо организовать несколько командных турниров по круговой системе с участием различного числа команд. График олимпиады очень плотный, поэтому каждый турнир нужно провести в минимально возможное число туров. Для составления расписания каждого турнира они решили обратиться за помощью к программистам.

### Формат входного файла

Во входном файле записано одно натуральное число  $N$  — количество команд, участвующих в турнире ( $2 \leq N \leq 100$ ).

### Формат выходного файла

В первой строке выведите минимальное количество туров  $K$ , необходимых для проведения кругового турнира из  $N$  команд. Каждая из  $K$  следующих строк содержит описание одного тура. В начале строки выведите количество игр  $n_i$ , которое необходимо сыграть в  $i$ -м туре. Далее идет  $n_i$  пар чисел — команды, которые играют в этом туре. Команды, играющие между собой, разделяются символом “-” (минус), а разные игры разделяются пробелом.

### Примеры

<code>cin.txt</code>	<code>cout.txt</code>
4	3 2 1-2 3-4 2 1-3 2-4 2 1-4 2-3
3	3 1 1-2 1 2-3 1 3-1

### Система оценивания

Online-группа тестов содержит тесты для  $N \leq 10$  и оценивается в 40 баллов.  
Offline-группа тестов оценивается в 60 баллов.

## Задача D. Питон

Имя входного файла: `din.txt`  
Имя выходного файла: `dout.txt`  
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 a - отрицательное")
    a = -a
print("Теперь a - положительное")
```

если условие `a < 0` будет истинно, то выполнятся две следующие строки `print("Число a - отрицательное")` и `a = -a`, а вот следующая строка `print("Теперь a - 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.

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

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 &gt; 0:     print("x &gt; 0")     print(x) else:     print("x &lt; 0")     print(-x) print("Bye")</pre>	<pre>if a &gt; b:     if a &gt; c:         print(a)     else:         print(c) else:     if b &gt; c:         print(b)     else:         print(c)</pre>	<pre>if x &gt; 0:     print(x)     print("x &gt; 0")</pre>	<pre>if x &gt; 0: else:     print(x)</pre>

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 непробельными 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 непробельный 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

din.txt	dout.txt
<pre>a, b, c = map(int, input().split()) if a &gt; b:     if a &gt; c:         print(a)     else:         print(c) else:     if b &gt; c:         print(b)     else:         print(c) exit()</pre>	0
<pre>x = int(input()) if x &lt; 0:     print("Negative")     x = -x else:     print("Positive") exit()</pre>	4

## Система оценивания

Online-группа тестов оценивается в 50 баллов. Тесты этой группы не содержат инструкции `else`.

Offline-группа тестов оценивается в 50 баллов.

Программа, которая выдает правильный ответ только на тех примерах, в которых ответ 0, будет оцениваться в 0 баллов (то есть для получения ненулевого числа баллов за задачу программа должна выдавать правильный ответ хотя бы на одном тесте, помимо теста из условия, в котором ответ не 0).

## Задача Е. К-квест

Имя входного файла: `ein.txt`  
Имя выходного файла: `eout.txt`  
Ограничение по времени: 0,5 секунды  
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, ...,  $N$  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

<code>ein.txt</code>	<code>eout.txt</code>
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

### Система оценивания

Тесты по этой задаче разбиты на группы. На 1-3 группах тестов проверка проводится во время тура (online), на последней группе — после окончания тура (offline).

В первой группе тестов  $1 \leq N \leq 100$ ,  $|a_i| \leq 100$ . Баллы начисляются только при прохождении всех тестов группы, группа оценивается в 20 баллов.

Во второй группе тестов  $1 \leq N \leq 2000$ ,  $|a_i| \leq 1\,000\,000$ . Баллы начисляются только при прохождении всех тестов группы, группа оценивается в 20 баллов.

В третьей группе тестов  $1 \leq N \leq 200\,000$ ,  $0 \leq a_i \leq 10^9$ . Баллы начисляются только при прохождении всех тестов группы, группа оценивается в 20 баллов.

В четвертой группе тестов  $1 \leq N \leq 200\,000$ ,  $|a_i| \leq 10^9$ . Каждый тест этой группы оценивается отдельно. Общее число баллов за тесты этой группы равно 40.