

## Задача А. Наибольшая возрастающая подпоследовательность

Имя входного файла: `lis.in`  
Имя выходного файла: `lis.out`

Дана последовательность, требуется найти её наибольшую возрастающую подпоследовательность.

### Формат входного файла

В первой строке входных данных задано целое число  $N$  — длина последовательности ( $1 \leq N \leq 5000$ ). Во второй строке задается сама последовательность. Числа разделяются пробелом. Элементы последовательности — целые числа, не превосходящие  $10^9$  по модулю.

### Формат выходного файла

В первой строке выведите длину наибольшей возрастающей подпоследовательности, а во второй строке выведите через пробел саму наибольшую возрастающую подпоследовательность данной последовательности. Если ответов несколько — выведите любой.

### Пример

<code>lis.in</code>	<code>lis.out</code>
6	3
3 29 5 5 28 6	3 5 28

## Задача В. Наибольшая общая подпоследовательность

Имя входного файла: `lcs.in`  
Имя выходного файла: `lcs.out`

Даны две последовательности, требуется найти и вывести их наибольшую общую подпоследовательность.

### Формат входного файла

В первой строке входных данных содержится целое число  $N$  — длина первой последовательности ( $1 \leq N \leq 2000$ ). Во второй строке заданы члены первой последовательности (через пробел) — целые числа, не превосходящие  $10^9$  по модулю. В третьей строке записано целое число  $M$  — длина второй последовательности ( $1 \leq M \leq 2000$ ). В четвертой строке задаются члены второй последовательности (через пробел) — целые числа, не превосходящие  $10^9$  по модулю.

### Формат выходного файла

В первой строке выведите длину наибольшей общей подпоследовательности, а во второй строке выведите через пробел саму наибольшую общую подпоследовательность данных последовательностей. Если ответов несколько — выведите любой.

### Пример

<code>lcs.in</code>	<code>lcs.out</code>
3	2
1 2 3	2 3
4	
2 3 1 5	

## Задача С. Рюкзак

Имя входного файла: `knapsack.in`  
Имя выходного файла: `knapsack.out`

Дано  $n$  предметов массой  $m_1, \dots, m_n$  и стоимостью  $c_1, \dots, c_n$  соответственно.

Ими наполняют рюкзак, который выдерживает вес не более  $m$ . Определите набор предметов, который можно унести в рюкзаке, имеющий наибольшую стоимость.

### Формат входного файла

В первой строке вводится натуральное число  $n$ , не превышающее 1000 и натуральное число  $m$ , не превышающее 10000.

Во второй строке вводятся  $n$  натуральных чисел  $m_i$ , не превышающих 100.

Во третьей строке вводятся  $n$  натуральных чисел  $c_i$ , не превышающих 100.

### Формат выходного файла

В первой строке выведите количество предметов, которые нужно взять. Во второй строке выведите номера предметов (числа от 1 до  $n$ ), которые войдут в рюкзак наибольшей стоимости.

### Пример

<code>knapsack.in</code>	<code>knapsack.out</code>
4 6	3
2 4 1 2	1 3 4
7 2 5 1	

## Задача D. Расстояние Левенштейна

Имя входного файла: `levenshtein.in`  
Имя выходного файла: `levenshtein.out`

Дана текстовая строка. С ней можно выполнять следующие операции:

- Заменить один символ строки на другой символ.
- Удалить один произвольный символ.
- Вставить произвольный символ в произвольное место строки.

Например, при помощи первой операции из строки «СОК» можно получить строку «СУК», при помощи второй операции — строку «ОК», при помощи третьей операции — строку «СТОК». Минимальное количество таких операций, при помощи которых можно из одной строки получить другую, называется стоимостью редактирования или расстоянием Левенштейна. Определите расстояние Левенштейна для двух данных строк.

### Формат входного файла

Программа получает на вход две строки, длина каждой из которых не превосходит 5000 символов, строки состоят только из заглавных латинских букв.

### Формат выходного файла

Требуется вывести одно число — расстояние Левенштейна для данных строк.

### Пример

<code>levenshtein.in</code>	<code>levenshtein.out</code>
ABCDEFGH ACDEXGHI	3

## Задача E. Умножение матриц

Имя входного файла: `matrix.in`  
Имя выходного файла: `matrix.out`

В произведении последовательности матриц полностью расставлены скобки, если выполняется один из следующих пунктов:

- Произведение состоит из одной матрицы.
- Оно является заключенным в скобки произведением двух произведений с полностью расставленными скобками.

Полная расстановка скобок называется оптимальной, если количество операций, требуемых для вычисления произведения, минимально.

Требуется найти оптимальную расстановку скобок в произведении последовательности матриц.

### Формат входного файла

В первой строке входных данных содержится целое число  $n$  — количество матриц ( $1 \leq n \leq 400$ ). В  $n$  следующих строк содержится по два целых числа  $a_i$  и  $b_i$  — количество строк и столбцов в  $i$ -ой матрице соответственно ( $1 \leq a_i, b_i \leq 100$ ). Гарантируется, что  $b_i = a_{i+1}$  для любого  $1 \leq i \leq n - 1$ .

### Формат выходного файла

В выходной файл выведите оптимальную расстановку скобок. Если таких расстановок несколько, выведите любую.

### Пример

<code>matrix.in</code>	<code>matrix.out</code>
3 10 50 50 90 90 20	$((AA)A)$

### Пояснение

В данном примере возможно две расстановки скобок:  $((AA)A)$  и  $(A(AA))$ . При первой количество операций будет равно  $10 \cdot 50 \cdot 90 + 10 \cdot 90 \cdot 20 = 63000$ , а при второй —  $10 \cdot 50 \cdot 20 + 50 \cdot 90 \cdot 20 = 100000$ .

## Задача F. Максимальный подпалиндром

Имя входного файла: `palindrome.in`  
Имя выходного файла: `palindrome.out`

Палиндромом называется строка, которая одинаково читается как слева направо, так и справа налево. Подпалиндромом данной строки называется последовательность символов из данной строки, не обязательно идущих подряд, являющаяся палиндромом. Например, «HELOLEH» является подпалиндромом строки «HTEOLFEOLEH». Напишите программу, находящую в данной строке подпалиндром максимальной длины.

### Формат входного файла

Во входном файле находится строка длиной не более 2000 символов, состоящая из заглавных букв латинского алфавита.

### Формат выходного файла

Выведите на первой строке выходного файла длину максимального подпалиндрома, а на второй строке сам максимальный подпалиндром. Если таких подпалиндромов несколько, то ваша программа должна вывести любой из них.

### Пример

<code>palindrome.in</code>	<code>palindrome.out</code>
HTEOLFEOLEH	7 HEOLOEH

## Задача G. Концерт

Имя входного файла: `concert.in`  
Имя выходного файла: `concert.out`

Петя — известный гитарист, и скоро у него концерт. Он не хочет играть все песни с одинаковой громкостью, поэтому он решил менять уровень громкости гитары перед каждой новой песней. Перед началом концерта, он сделал список из чисел  $d_i$ . Перед  $i$ -ой песней он может либо добавить  $d_i$  к текущей громкости, либо вычесть. В начале концерта громкость гитары равна  $b$ . Громкость гитары всегда должна быть в диапазоне от 0 до  $m$ , включительно. Чтобы произвести больший эффект, Петя хочет сыграть последнюю песню концерта с максимально возможной громкостью. Найдите максимальную громкость, с которой можно исполнить последнюю песню. Если нет никакого способа сыграть все песни так чтобы громкость не превысила  $m$  и не опустилась ниже 0, выведите -1.

### Формат входного файла

Первая строка содержит число песен  $n$  ( $1 \leq n \leq 50$ ), вторая строка содержит  $n$  чисел  $d_i$  ( $1 \leq d_i \leq m$ ), третья строка содержит число  $b$  ( $0 \leq b \leq m$ ), четвертая строка содержит число  $m$  ( $1 \leq m \leq 1000$ ).

### Формат выходного файла

Выведите одно число — максимально возможную громкость последней песни или  $-1$ , если выполнить условия нельзя.

### Пример

<code>concert.in</code>	<code>concert.out</code>
3 5 3 7 5 10	10
4 15 2 9 10 8 20	-1
14 74 39 127 95 63 140 99 96 154 18 137 162 14 88 40 243	238

## Задача Н. Тапкодер

Имя входного файла: `tapcoder.in`  
Имя выходного файла: `tapcoder.out`

Тапкодёр — соревнование по программированию, похожее на TopCoder. В нем у каждого кодера также есть рейтинг. Есть два типа кодеров в Тапкодере: коричневые и бирюзовые. Коричневый кодер — это кодер, чей рейтинг больше или равен 2200. Бирюзовый кодер — кодер, чей рейтинг меньше 2200.

Кодер Вася участвует в Тапкодере. В настоящее время он бирюзовый кодер. Его рейтинг в начале сезона равен  $x$ .

В сезоне пройдет  $n$  контестов. В каждом из них, Вася либо может либо напрячься и затащить, либо слить. Для каждого контеста известен вес  $d_i$ . Если Вася затащит в  $i$ -ом контесте, его рейтинг вырастет на  $d_i$ , а если он его сольет, то его рейтинг снизится на  $d_i$ , но не ниже нуля (то есть, если рейтинг Васи был меньше  $d_i$ , то он становится равным 0).

Васе нравится бирюзовый цвет. Поэтому он старается никогда не становиться коричневым кодером два раза подряд. То есть, всякий раз, когда Вася становится коричневым кодером, то следующий контест (если остались) он должен слить и стать бирюзовым снова.

В Тапкодере есть награда «Хамелеон года» для кодера, цвет которого изменился больше всего раз в течение года.

Найдите максимальное число изменений цвета, которые может сделать Вася в этом сезоне.

### Формат входного файла

Первая строка содержит число контестов  $n$  ( $1 \leq n \leq 50$ ), вторая строка содержит  $n$  чисел  $d_i$  ( $1 \leq d_i \leq 10^9$ ), третья строка содержит число  $x$  ( $0 \leq x < 2200$ ).

### Формат выходного файла

Выведите одно число — максимальное число изменений цвета, которые может сделать Вася.

### Пример

<code>tapcoder.in</code>	<code>tapcoder.out</code>
5 100 200 100 1 1 2000	3
5 0 0 0 0 0 2199	0
9 90000 80000 70000 60000 50000 40000 30000 20000 10000 0	1

## Задача I. Монстры

Имя входного файла: `monsters.in`  
Имя выходного файла: `monsters.out`

Манао идет через долину, населенную монстрами. Во время своего путешествия, он столкнется с несколькими монстрами по одному. Ужасность  $i$ -го монстра — положительное целое число  $d_i$ .

Манао не собирается бороться с монстрами. Вместо этого, он собирается подкупить некоторых из них, чтобы они присоединились к нему. Для подкупа  $i$ -го монстра, Манао необходимо  $c_i$  золотых монет. Монстры не слишком жадные, поэтому каждое значение  $c_i$  будет либо 1, либо 2.

В начале Манао путешествует один. Каждый раз, когда он встречает монстра, он может подкупить его, если же он этого не сделает, то монстр может на него напасть. Монстр атакует Манао если его ужасность строго больше, чем общая ужасность всех монстров в команде Манао. Иными словами, всякий раз, когда Манао встречает монстра, который напал бы на него, он обязательно должен подкупить его. Если же он сталкивается с монстром, который на него не нападает, то он может либо подкупить его, либо просто пройти мимо. Найдите минимальное число монет, которые нужно потратить Манао, чтобы пройти через долину.



Рассмотрим следующий пример: Манао идет через долину, в которой живут Дракон, Гидра и Кролик-убийца. Когда он встречает дракона, он не имеет никакого выбора, кроме как подкупить его, потратив 1 золотую монету (Манао всегда должен подкупить первого монстра, которого он встречает, потому что, когда он путешествует в одиночку, общая ужасность монстров в его команде равна нулю). Когда они встречают Гидру, Манао может либо пройти мимо, либо подкупить ее. В конце он должен пройти Кролика-убийцу. Если Манао подкупил Гидру, то общая ужасность его команды превышает ужасность Кролика, поэтому они могут пройти. В противном случае, придется подкупить Кролика за две золотые монеты. Таким образом, оптимальным выбором является подкупить Гидру, а затем пройти мимо Кролика-убийцы. Общая сумма потраченных денег таким образом, будет 2 золотые монеты.

### Формат входного файла

Первая строка содержит число контесов  $n$  ( $1 \leq n \leq 50$ ), вторая строка содержит  $n$  чисел  $d_i$  ( $1 \leq d_i \leq 10^{12}$ ), третья строка содержит  $n$  чисел  $c_i$  ( $1 \leq c_i \leq 2$ ).

### Формат выходного файла

Выведите одно число — минимальное число монет, которое нужно потратить, чтобы пройти через долину

## Пример

monsters.in	monsters.out
3 8 5 10 1 1 2	2
4 1 2 4 1000000000 1 1 1 2	5
6 200 107 105 206 307 400 1 2 1 1 1 2	2
10 5216 12512 613 1256 66 17202 30000 23512 2125 33333 2 2 1 1 1 1 2 1 2 1	5

## Задача J. Симпатичный узор

Имя входного файла: `nice.in`  
Имя выходного файла: `nice.out`

Компания *Broken Tiles* планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер  $1 \times 1$  метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника  $n \times m$  метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезные проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным. Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата  $2 \times 2$  метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

### Формат входного файла

На первой строке входного файла находятся два натуральных числа  $n$  и  $m$ .  $1 \leq n \cdot m \leq 30$ .

### Формат выходного файла

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера  $n \times m$ . Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

### Пример

<code>nice.in</code>	<code>nice.out</code>
2 2	14
3 3	322

## Задача К. Логическое дерево

Имя входного файла: `boolean.in`  
Имя выходного файла: `boolean.out`

Рассмотрим разновидность двоичного дерева которую мы назовем логическим деревом. В этом дереве каждый уровень полностью заполнен, за исключением, возможно, последнего (самого глубокого) уровня. При этом, все вершины последнего уровня находятся максимально слева. Дополнительно, каждая вершина дерева имеет ноль или двоих детей.

Каждая вершина дерева имеет связанное с ней логическое значение (1 или 0). Кроме этого, каждая *внутренняя* вершина имеет связанную с ней логическую операцию (“И” или “ИЛИ”). Значение вершины с операцией “И” — это логическое “И” значений ее детей. Аналогично, значение вершины с операцией “ИЛИ” — это логическое “ИЛИ” значений ее детей. Значения всех листьев задаются во входном файле, поэтому значения всех вершин дерева могут быть найдены.

Наибольший интерес для нас представляет корень дерева. Мы хотим, чтобы он имел заданное логическое значение  $v$ , которое может отличаться от текущего. К счастью, мы можем изменять логические операции некоторых внутренних вершин (заменить “И” на “ИЛИ” и наоборот).

Дано описание логического дерева и набор вершин, операции в которых могут быть изменены. Найдите наименьшее количество вершин, которые следует изменить, чтобы корень дерева принял заданное значение  $v$ . Если это невозможно, то выведите строку “IMPOSSIBLE” (без кавычек).

### Формат входного файла

В первой строке входного файла находятся два числа  $n$  и  $v$  ( $1 \leq n \leq 10000, 0 \leq v \leq 1$ ) — количество вершин в дереве и требуемое значение в корне соответственно. Поскольку все вершины имеют ноль или двоих детей, то  $n$  — нечетное. Следующие  $n$  строк описывают вершины дерева. Вершины нумеруются с 1 до  $n$ .

Первые  $(n-1)/2$  строк описывают внутренние вершины. Каждая из них содержит два числа —  $g$  и  $c$ , которые принимают значение либо 0, либо 1. Если  $g = 1$ , то вершина представляет логическую операцию “И”, иначе она представляет логическую операцию “ИЛИ”. Если  $c = 1$ , то операция в вершине может быть изменена, иначе нет. Внутренняя вершина с номером  $i$  имеет детей  $2i$  и  $2i+1$ .

Следующие  $(n+1)/2$  строк описывают листья. Каждая строка содержит одно число 0 или 1 — значение листа.

### Формат выходного файла

В выходной файл выведите ответ на задачу.

### Пример

<code>boolean.in</code>	<code>boolean.out</code>
9 1 1 0 1 1 1 1 0 0 1 0 1 0 1	1
5 0 1 1 0 0 1 1 0	IMPOSSIBLE

## Задача L. Задача коммивояжера

Имя входного файла: `salesman.in`  
Имя выходного файла: `salesman.out`

Вам дан неориентированный взвешенный граф без петель и кратных ребер. Необходимо найти в нем путь наименьшего веса, который проходит по всем вершинам ровно один раз.

### Формат входного файла

В первой строке находятся два целых числа  $n$  и  $m$  — количество вершин и ребер в графе ( $1 \leq n \leq 18$ ,  $0 \leq m \leq \frac{n \cdot (n-1)}{2}$ ). Следующие  $m$  строк содержат описания ребер: три целых числа  $a_i, b_i, w_i$ , обозначающих соответственно пару вершин и вес ребра, соединяющего эти вершины ( $1 \leq a_i, b_i \leq n$ ,  $1 \leq w_i \leq 10^8$ ).

### Формат выходного файла

Выведите в выходной файл одно число — вес искомого пути. Если такого пути не существует, выведите  $-1$ .

### Примеры

<code>salesman.in</code>	<code>salesman.out</code>
4 6 1 2 20 1 3 42 1 4 35 2 3 30 2 4 34 3 4 12	62
4 3 1 2 1 1 3 1 1 4 1	-1

## Задача М. Халява

Имя входного файла: `free.in`  
Имя выходного файла: `free.out`

Гриша очень любит газировку PupsiCola. Однажды он узнал, что, собрав несколько крышек со звездочками, можно получить футболку. Гриша нашел  $a$  крышек с одной звездочкой,  $b$  крышек с двумя звездочками и  $c$  крышек с тремя звездочками. На футболку можно обменять набор крышек, общее количество звездочек на которых не меньше  $k$ . Помогите Грише узнать, сколько футболок он может получить.

### Формат входного файла

Входной файл содержит целые числа  $a$ ,  $b$ ,  $c$  и  $k$  ( $0 \leq a, b, c \leq 1000$ ,  $0 \leq a \cdot b \cdot c \leq 10^7$ ,  $1 \leq k \leq 10000$ ).

### Формат выходного файла

Выведите максимальное количество футболок, которые может получить Гриша.

### Пример

<code>free.in</code>	<code>free.out</code>
2 2 2 4	3
0 0 4 4	2

## Задача N. Книжный шкаф

Имя входного файла: `bookshelf.in`  
Имя выходного файла: `bookshelf.out`

Когда Фермер Джон не доит коров, не собирает сено и не строит изгороди, он сидит и читает хорошую книгу. С годами он собрал коллекцию из  $n$  книг ( $1 \leq n \leq 2000$ ), и хочет построить для них новое множество книжных полок.

Книга  $i$  имеет ширину  $w_i$  и высоту  $h_i$ . Книги необходимо ставить на полки в определенном порядке: первая полка должна содержать книги с номерами от 1 до  $k_1$  для некоторого  $k_1$ . Вторая полка должна содержать книги от  $k_1 + 1$  до  $k_2$ , и т.д. Каждая полка имеет ширину  $l$  ( $1 \leq l \leq 10^9$ ). Высота полки равна высоте самой высокой книги на этой полке, а высота шкафа равна сумме высот всех полок.

Помогите ФД вычислить минимально возможную высоту книжного шкафа.

### Формат входного файла

- Строка 1: два разделенных пробелом целых числа:  $n$  и  $l$ .
- Строки 2 ..  $1 + n$ : Строка  $i + 1$  содержит два разделенных пробелом целых числа:  $h_i$  и  $w_i$ . ( $1 \leq h_i \leq 10^6$ ,  $1 \leq w_i \leq l$ ).

### Формат выходного файла

- Строка 1: Минимально возможная высота шкафа.

### Пример

<code>bookshelf.in</code>	<code>bookshelf.out</code>
5 10	21
5 7	
9 2	
8 5	
13 2	
3 8	

### Пояснение

Всего 5 книг. Каждая полка имеет ширину не более 10.

Всего 3 полки. Первая содержит книгу 1 (высота 5, ширина 7), вторая содержит книги 2..4 (высота 13, ширина 9), третья содержит книгу 5 (высота 3, ширина 8).

## Задача О. Зонтики

Имя входного файла: `umbrella.in`  
Имя выходного файла: `umbrella.out`

Сегодня идет дождь. Коровы Фермера Джона, пронумерованные от 1 до  $n$  ( $1 \leq n \leq 5000$ ) не хотят мокнуть, но они стоят в стойлах без крыш. Стойла расположены на прямой, корова  $i$  расположена в стойле с координатой  $x_i$  ( $1 \leq x_i \leq m$ ). Никакие две коровы не стоят в одном стойле.

Для того, чтобы защитить коров от дождя, ФД хочет купить им зонтики. Зонтик, который закрывает коров с координатами от  $x_i$  до  $x_j$ , имеет ширину  $(x_j - x_i + 1)$ . В магазине есть зонтики всех размеров от 1 до  $m$  ( $1 \leq m \leq 10^5$ ). Зонтик шириной  $w$  стоит  $c_w$  долларов ( $1 \leq c_w \leq 10^6$ ).

Помогите ФД определить минимальную сумму, которую он должен потратить на покупку зонтиков, чтобы защитить от дождя всех своих коров.

### Формат входного файла

- Строка 1: Два разделенных пробелами целых числа:  $n$  и  $m$ .
- Строки 2 ..  $n + 1$ : Строка  $i + 1$  содержит одно целое число:  $x_i$ .
- Строки  $n + 2$  ..  $n + m + 1$ : Строка  $n + j + 1$  содержит одно целое число:  $c_j$ .

### Формат выходного файла

- Строка 1: Одно целое число — минимальная стоимость покупки зонтиков для покрытия всех коров.

### Пример

<code>umbrella.in</code>	<code>umbrella.out</code>
6 12	9
1	
2	
11	
8	
4	
12	
2	
3	
4	
4	
8	
9	
15	
16	
17	
18	
19	
19	

### Пояснение

Всего есть 12 стойл, и стойла 1, 2, 4, 8, 11, 12 содержат коров. Зонтик на одно стойло стоит 2, зонтик на два стойла стоит 3 и т.д.

При покупке зонтиков с размерами 4, 1 и 2, можно покрыть всех коров и это будет стоить 9 ( $4 + 2 + 3 = 9$ )