

## 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 29 5 5 28 6	3 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 ACDEXGIH	3

## Задача Е. 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. 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

bookshelf.in	bookshelf.out
5 10 5 7 9 2 8 5 13 2 3 8	21

### 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 Н. 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

salesman.in	salesman.out
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 I. Подготовка

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

Артур хочет научиться решать задачи на динамическое программирование. Для этого он решает задачи с одного известного сайта с автоматической системой проверки задач.

Сайт устроен следующим образом: у каждого пользователя есть свой рейтинг  $R$ . У каждой задачи на этом сайте есть сложность  $s_i$ . Задача доступна для сдачи только тем пользователям, у которых рейтинг не меньше сложности этой задачи ( $R \geq s_i$ ). Это нужно для того, чтобы новичок не принялся решать сложную задачу. После решения задачи рейтинг участника увеличивается на  $p_i$ . Каждую из задач можно только один раз.

Артур изучил все задачи и для каждой оценил число дней  $t_i$ , которое ему потребуется для ее решения. Теперь Артур хочет спланировать оставшееся до олимпиады время таким образом, чтобы достигнуть максимально возможного рейтинга. Сейчас у Артура рейтинг  $R_0$ . Какой максимальный рейтинг он может получить через  $T$  дней?

### Формат входного файла

В первой строке входных данных задано три целых числа:  $n$ ,  $T$  и  $R_0$  — количество задач на сайте, количество дней и рейтинг Артура на текущий момент.

В следующих  $n$  строках задано по три целых числа  $s_i$ ,  $p_i$  и  $t_i$  — сложность задачи, увеличение рейтинга за решение задачи и количество дней, которое требуется на её решение.

Все числа во входных данных положительные.

### Формат выходного файла

В первой строке выходного файла выведите максимальный рейтинг, который может иметь Артур через  $T$  дней.

Во второй строке выведите номера задач, разделенные пробелами, в том порядке, в котором необходимо решать задачи, чтобы получить максимальный рейтинг. Задачи нумеруются от 1 до  $n$  в том порядке, в котором они заданы во входных данных.

### Примеры

practice.in	practice.out
4 10 1 10 10 1 1 5 5 7 3 1 2 4 4	20 2 4 1
4 10 1 11 10 1 1 5 5 7 3 1 2 4 4	13 2 4 3
3 4 3 3 3 2 3 3 2 3 5 3	9 1 2
3 5 3 3 3 2 3 3 2 3 5 4	9 1 2

## Задача J. Шаблоны

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод

Многие операционные системы используют шаблоны для ссылки на группы объектов: файлов, пользователей, и т. д. Ваша задача — реализовать простейший алгоритм проверки шаблонов для имен файлов.

В этой задаче алфавит состоит из маленьких букв английского алфавита и точки ('.'). Шаблоны могут содержать произвольные символы алфавита, а также два специальных символа: '?' и '\*'. Знак вопроса ('?') соответствует ровно одному произвольному символу. Звездочка '\*' соответствует подстроке произвольной длины (возможно, нулевой). Символы алфавита, встречающиеся в шаблоне, отображаются на ровно один такой же символ в проверяемой строке. Строка считается подходящей под шаблон, если символы шаблона можно последовательно отобразить на символы строки таким образом, как описано выше. Например, строки "ab", "aab" и "beda." подходят под шаблон "\*a?", а строки "bebe", "a" и "ba" — нет.

### Формат входного файла

Первая строка входного файла определяет шаблон  $P$ . Вторая строка  $S$  состоит только из символов алфавита. Ее необходимо проверить на соответствие шаблону. Длины обеих строк не превосходят 10 000. Строки могут быть пустыми — будьте внимательны!

### Формат выходного файла

Если данная строка подходит под шаблон, выведите YES. Иначе выведите NO.

### Примеры

стандартный ввод	стандартный вывод
k?t*n kitten	YES
k?t?n kitten	NO

## Задача К. Горные лыжи

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

Суровая зима в Санкт-Петербурге длится  $n$  дней. Таня очень любит кататься на лыжах и часто выезжает на близлежащий горнолыжный курорт. Так, про некоторые дни последней зимы Таня помнит, что была в этот день на горнолыжном курорте. Про другие дни никакой информации нет.

Известно, что Таня всегда ездит на курорт одинаково: она выезжает утром некоторого дня, проводит на курорте ровно  $k$  дней и возвращается вечером  $k$ -го дня поездки. При этом могло оказаться, что Таня снова поехала на курорт на следующий день после окончания предыдущей поездки. Те, дни, когда Таня не ездила на горнолыжный курорт, она провела в городе.

Зима закончилась, и подруги говорят Тане, что она слишком много катается на лыжах. Чтобы понять, так ли это, Таня хочет выяснить, каким могло быть максимальное количество зимних дней, которые провела в городе.

Таня могла первый раз поехать на курорт до начала зимы и могла закончить последнюю поездку после окончания зимы.

### Формат входного файла

В первой строке находятся три целых положительных числа  $n$ ,  $k$  и  $m$  — продолжительность зимы в днях, длительность одной поездки на горнолыжный курорт в днях и количество дней, в которые Таня точно была на курорте ( $1 \leq k \leq n \leq 10^9$ ,  $1 \leq m \leq 2 \cdot 10^5$ ,  $m \leq n$ ).

Во второй строке находятся  $m$  чисел  $d_1, d_2, \dots, d_m$  — номера дней в которые Таня точно была на курорте ( $1 \leq d_i \leq n$ ). Каждый день перечислен не более одного раза.

### Формат выходного файла

Выведите единственное целое число — максимальное количество зимних дней, в которые Таня не была на горнолыжном курорте.

### Примеры

ski.in	ski.out
4 2 2 1 4	2
7 4 3 4 3 5	3
5 1 5 5 4 3 2 1	0
13 3 6 3 5 6 8 9 11	4

### Пояснение

В первом примере Таня могла быть на курорте два раза: первый раз начиная за день до зимы и заканчивая в 1-й день зимы; второй раз начиная в последний день зимы и заканчивая в 1-й день после зимы.

Таким образом, Таня могла провести в городе второй и третий дни зимы.

Во втором примере Таня могла быть на курорте один раз, например, начиная со второго дня зимы и заканчивая пятым днём зимы. Таким образом, Таня могла провести в городе три дня — в первый, в шестой и в седьмой дни зимы.

## Задача L. Выбор вершин взвешенного дерева

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

Дано дерево. В вершинах написаны целые числа. Множество вершин дерева называется *допустимым*, если никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа. Для каждого такого множества вычислим сумму чисел, написанных в его вершинах. Какова максимальная из этих сумм?

### Формат входного файла

Граф в этой задаче задан в виде *корневого дерева*. В графе выделена вершина "— *корень дерева*. Для каждой вершины  $i$ , не являющейся корнем, задан номер вершины-предка  $p_i$  в корневом дереве. Дерево, заданное таким образом, состоит из рёбер  $i$  "—  $p_i$  для всех вершин  $i$ , кроме корня.

В первой строке входного файла записано целое число  $n$  "— количество вершин в графе ( $1 \leq n \leq 100$ ). В следующих  $n$  строках задан граф. В  $i$ -й из этих строк записаны через пробел два целых числа  $p_i$  и  $q_i$ ; здесь  $p_i$  "— номер вершины-предка  $i$ -ой вершины, а  $q_i$  "— число, записанное в этой вершине. Для корня дерева  $p_i = 0$ ; для всех остальных вершин  $1 \leq p_i \leq n$ . Числа  $q_i$  не превосходят по модулю 10 000.

Гарантируется, что заданный во входном файле граф является деревом.

### Формат выходного файла

В первой строке выходного файла выведите одно число "— максимальную сумму чисел в допустимом множестве.

### Примеры

selectw.in	selectw.out
5 0 1 1 2 1 3 2 4 3 5	10
6 5 8 6 0 5 -1 1 1 0 3 1 2	8

## Задача М. Мостостроение

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

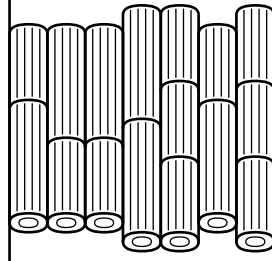
В деревне Гадюкино регулярно идут проливные дожди, в результате чего речка Вонючка, которую обычно можно просто перешагнуть, выходит из берегов. Чтобы можно было перейти разлившуюся реку, планируется построить плавучий мост из бревен, оставшихся от строительства бани бизнесмена, поселившегося неподалеку.

Все оставшиеся бревна имеют одинаковую толщину. При этом есть  $x$  бревен длины  $a$  и  $y$  бревен длины  $b$ .

Построенный мост должен состоять из  $l$  рядов, каждый из которых составлен из одного или нескольких бревен. Пилить бревна нельзя, так как последняя пила утонула при разливе Вонючки.

Главный инженер хочет построить мост максимальной возможной ширины, при этом ширина моста определяется по минимальной ширине ряда бревен.

Например, если нужно построить мост из семи рядов, и при этом есть шесть бревен длины 3 и десять бревен длины 2, то можно построить мост ширины 5.



### Формат входного файла

Входной файл содержит пять натуральных чисел:  $x$ ,  $a$ ,  $y$ ,  $b$  и  $l$ . Все числа не превышают 150. Общее количество бревен не меньше  $l$ .

### Формат выходного файла

Выведите в выходной файл одно число — максимальную возможную ширину моста.

### Примеры

<code>bridge.in</code>	<code>bridge.out</code>
6 3 10 2 7	5
10 7 20 9 25	9