

## Задача А. Кузнечик собирает монеты

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Кузнечик прыгает по столбикам, расположенным на одной линии на равных расстояниях друг от друга. Столбики имеют порядковые номера от 1 до  $N$ . В начале Кузнечик сидит на столбике с номером 1. Он может прыгнуть вперед на расстояние от 1 до  $K$  столбиков, считая от текущего.

На каждом столбике Кузнечик может получить или потерять несколько золотых монет (для каждого столбика это число известно). Определите, как нужно прыгать Кузнечику, чтобы собрать наибольшее количество золотых монет. Учитывайте, что Кузнечик не может прыгать назад.

### Формат входных данных

В первой строке вводятся два натуральных числа:  $N$  и  $K$  ( $2 \leq N, K \leq 10000$ ), разделённые пробелом. Во второй строке записаны через пробел  $N - 2$  целых числа – количество монет, которое Кузнечик получает на каждом столбике, от 2-го до  $N - 1$ -го. Если это число отрицательное, Кузнечик теряет монеты. Гарантируется, что все числа по модулю не превосходят 10 000.

### Формат выходных данных

В первой строке программа должна вывести наибольшее количество монет, которое может собрать Кузнечик. Во второй строке выводится число прыжков Кузнечика, а в третьей строке – номера всех столбиков, которые посетил Кузнечик (через пробел в порядке возрастания).

Если правильных ответов несколько, выведите любой из них.

### Пример

input.txt	output.txt
5 3	7
2 -3 5	3
	1 2 4 5

## Задача В. Наибольшая возрастающая подпоследовательность

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Пусть  $a_1, a_2, \dots, a_n$  — числовая последовательность. Длина последовательности — это количество элементов этой последовательности. Последовательность  $a_{i_1}, a_{i_2}, \dots, a_{i_k}$  называется подпоследовательностью последовательности  $a$ , если  $1 \leq i_1 < i_2 < \dots < i_k \leq n$ . Последовательность  $a$  называется возрастающей, если  $a_1 < a_2 < \dots < a_n$ .

Вам дана последовательность, содержащая  $n$  целых чисел. Найдите ее самую длинную возрастающую подпоследовательность.

### Система оценки

Подзадача 1 (60 баллов)  $n \leq 300$   
Подзадача 2 (40 баллов)  $n \leq 2000$

### Формат входных данных

В первой строке задано одно число  $n$  ( $1 \leq n \leq 2000$ ) — длина подпоследовательности. В следующей строке задано  $n$  целых чисел  $a_i$  ( $-10^9 \leq a_i \leq 10^9$ ) — элементы последовательности.

### Формат выходных данных

В первой строке выведите число  $k$  — длину наибольшей возрастающей подпоследовательности. В следующей строке выведите  $k$  чисел — саму подпоследовательность.

### Примеры

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

## Задача С. Ход конем

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

Шахматная ассоциация решила оснастить всех своих сотрудников такими телефонными номерами, которые бы набирались на кнопочном телефоне ходом коня. Например, ходом коня набирается телефон 340-49-27. При этом телефонный номер не может начинаться ни с цифры 0, ни с цифры 8.

1	2	3
4	5	6
7	8	9
	0	

Напишите программу, определяющую количество телефонных номеров длины  $N$ , набираемых ходом коня. Поскольку таких номеров может быть очень много, выведите ответ по модулю  $10^9$ .

### Формат входных данных

Во входном файле записано целое число  $N$  ( $1 \leq N \leq 100$ ).

### Формат выходных данных

Выведите в выходной файл искомое количество телефонных номеров по модулю  $10^9$ .

### Примеры

стандартный ввод	стандартный вывод
1	8
2	16

## Задача D. Расстояние по Левенштейну

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дана текстовая строка. С ней можно выполнять следующие операции:

1. Заменить один символ строки на другой символ.
2. Удалить один произвольный символ.
3. 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 входных данных

Программа получает на вход две строки, длина каждой из которых не превосходит 1000 символов, строки состоят только из заглавных латинских букв.

### Формат выходных данных

Требуется вывести одно число – расстояние Левенштейна для данных строк.

### Пример

<code>input.txt</code>	<code>output.txt</code>
ABCDEFGH ACDEXGHI	3

## Задача Е. Кафе

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

Около Петинского университета недавно открылось новое кафе, в котором действует следующая система скидок: при каждой покупке более чем на 100 рублей покупатель получает купон, дающий право на один бесплатный обед (при покупке на сумму 100 рублей и меньше такой купон покупатель не получает).

Однажды Пете на глаза попался преискуронт на ближайшие  $n$  дней. Внимательно его изучив, он решил, что будет обедать в этом кафе все  $n$  дней, причем каждый день он будет покупать в кафе ровно один обед. Однако стипендия у Пети небольшая, и поэтому он хочет по максимуму использовать предоставляемую систему скидок так, чтобы его суммарные затраты были минимальны. Требуется найти минимально возможную суммарную стоимость обедов и номера дней, в которые Пете следует воспользоваться купонами.

### Формат входных данных

В первой строке входного файла записано целое число  $n$  ( $0 \leq n \leq 100$ ). В каждой из последующих  $n$  строк записано одно целое число, обозначающее стоимость обеда в рублях на соответствующий день. Стоимость — неотрицательное целое число, не превосходящее 300.

### Формат выходных данных

В первой строке выдайте минимальную возможную суммарную стоимость обедов. Во второй строке выдайте два числа  $k_1$  и  $k_2$  — количество купонов, которые останутся неиспользованными у Пети после этих  $n$  дней и количество использованных им купонов соответственно.

В последующих  $k_2$  строках выдайте в возрастающем порядке номера дней, когда Пете следует воспользоваться купонами. Если существует несколько решений с минимальной суммарной стоимостью, то выдайте то из них, в котором значение  $k_1$  максимально (на случай, если Петя когда-нибудь ещё решит заглянуть в это кафе). Если таких решений несколько, выведите любое из них.

### Примеры

стандартный ввод	стандартный вывод
5 110 40 120 110 60	260 0 2 3 5
3 110 110 110	220 1 1 2

## Задача F. Удаление скобок 2.0

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дана строка, составленная из круглых, квадратных и фигурных скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность.

### Формат входных данных

Во входном файле записана строка из круглых, квадратных и фигурных скобок. Длина строки не превосходит 100 символов.

### Формат выходных данных

Выведите максимальную длину строки, являющуюся правильной скобочной последовательностью, которую можно получить из исходной строки удалением некоторых символов.

### Пример

стандартный ввод	стандартный вывод
([)]	2

## Задача G. Удаление скобок 2.0

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дана строка, составленная из круглых, квадратных и фигурных скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность.

### Формат входных данных

Во входном файле записана строка из круглых, квадратных и фигурных скобок. Длина строки не превосходит 100 символов.

### Формат выходных данных

Выведите строку максимальной длины, являющейся правильной скобочной последовательностью, которую можно получить из исходной строки удалением некоторых символов.

### Пример

стандартный ввод	стандартный вывод
([)]	[]

## Задача Н. Выбор вершин дерева

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дан граф, являющийся деревом. Множество вершин графа называется *допустимым*, если никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа. Для каждого такого множества посчитаем количество вершин в нём. Каково максимальное из этих количеств?

### Формат входных данных

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 записано целое число  $p_i$  — номер вершины-предка  $i$ -й вершины. Для корня дерева  $p_i = 0$ ; для всех остальных вершин  $1 \leq p_i \leq n$ .

Гарантируется, что заданный во входном файле граф является деревом.

### Формат выходных данных

В первой строке выходного файла выведите одно число — максимальное количество вершин в допустимом множестве.

### Примеры

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



## Задача I. Паросочетание

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Двудольным графом называется неориентированный граф  $(V, E)$ ,  $E \subseteq V \times V$  такой, что его множество вершин  $V$  можно разбить на два множества  $A$  и  $B$ , для которых  $\forall (e_1, e_2) \in E$   $e_1 \in A$ ,  $e_2 \in B$  и  $A \cup B = V$ ,  $A \cap B = \emptyset$ .

Паросочетанием в двудольном графе называется любой набор его несмежных рёбер, то есть такой набор  $S \subseteq E$ , что для любых двух рёбер  $e_1 = (u_1, v_1)$ ,  $e_2 = (u_2, v_2)$  из  $S$   $u_1 \neq u_2$  и  $v_1 \neq v_2$ .

Ваша задача — найти максимальное паросочетание в двудольном графе, то есть паросочетание с максимально возможным числом рёбер.

### Формат входных данных

В первой строке записаны два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 250$ ), где  $n$  — число вершин в множестве  $A$ , а  $m$  — число вершин в  $B$ .

Далее следуют  $n$  строк с описаниями рёбер —  $i$ -я вершина из  $A$  описана в  $(i + 1)$ -й строке файла. Каждая из этих строк содержит номера вершин из  $B$ , соединённых с  $i$ -й вершиной  $A$ . Гарантируется, что в графе нет кратных ребер. Вершины в  $A$  и  $B$  нумеруются независимо (с единицы). Список завершается числом 0.

### Формат выходных данных

Первая строка выходного файла должна содержать одно целое число  $l$  — количество рёбер в максимальном паросочетании. Далее следуют  $l$  строк, в каждой из которых должны быть два целых числа  $u_j$  и  $v_j$  — концы рёбер паросочетания в  $A$  и  $B$  соответственно.

### Пример

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

## Задача J. Продавец аквариумов

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

Продавец аквариумов для кошек хочет объехать  $n$  городов, посетив каждый из них ровно один раз. Помогите ему найти кратчайший путь.



### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 13$ ) — количество городов. Следующие  $n$  строк содержат по  $n$  чисел — длины путей между городами.

В  $i$ -й строке  $j$ -е число —  $a_{i,j}$  — это расстояние между городами  $i$  и  $j$  ( $0 \leq a_{i,j} \leq 10^6$ ;  $a_{i,j} = a_{j,i}$ ;  $a_{i,i} = 0$ ).

### Формат выходных данных

В первой строке выходного файла выведите длину кратчайшего пути. Во второй строке выведите  $n$  чисел — порядок, в котором нужно посетить города.

### Примеры

стандартный ввод	стандартный вывод
5 0 183 163 173 181 183 0 165 172 171 163 165 0 189 302 173 172 189 0 167 181 171 302 167 0	666 4 5 2 3 1

## Задача К. Симпатичные узоры

Имя входного файла: `nice.in`  
Имя выходного файла: `nice.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Компания BrokenTiles 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 серьезных 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 входных данных

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

### Формат выходных данных

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.

### Примеры

<code>nice.in</code>	<code>nice.out</code>
1 1	2
1 2	4

## Задача L. Cows in a Skyscraper

Имя входного файла: `skyscraper.in`  
Имя выходного файла: `skyscraper.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Коровы любят соревноваться в беге по лестницам небоскребов. А вниз потом едут на лифте.

Лифт имеет максимальную вместимость  $w$  ( $1 \leq w \leq 10^8$ ) фунтов, а корова номер  $i$  весит  $c_i$  ( $1 \leq c_i \leq w$ ) фунтов.

Помогите Бесси определить минимальное количество спусков лифта, чтобы переместить вниз все  $n$  ( $1 \leq n \leq 18$ ) коров.

Сумма весов коров в каждом спуске не должна превышать  $W$ .

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $w$  ( $1 \leq n \leq 18$ ;  $1 \leq w \leq 10^8$ ).

Следующие  $n$  строк содержат веса коров:  $i$ -я строка содержит целое число  $c_i$  ( $1 \leq c_i \leq w$ ).

### Формат выходных данных

Первая строка должна содержать число  $r$  — минимально количество спусков лифта.

Каждая из следующих  $r$  строк содержит множество коров, которые сейчас спускаются. Строка начинается с количество коров на данном спуске, далее содержатся номера коров.

### Примеры

<code>skyscraper.in</code>	<code>skyscraper.out</code>
4 10	3
5	2 1 3
6	1 2
3	1 4
7	

### Замечание

Всего 4 коровы с весами 5 6 3 и 7. Вместимость лифта — 10.

Мы можем поместить в лифт корову 3 и любую из оставшихся коров. Но все другие коровы не помещаются даже по две. В решении представленном выше, в первом спуске участвуют коровы 1 и 3, Во втором — корова 2, в третьем — корова 4. Существует несколько правильных решений для данного ввода.