

### у2020-1-3. Динамическое программирование

#### А. Кузнечик собирает монеты

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

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 и хочет добраться до столбика с номером  $n$ . Он может прыгнуть вперед на расстояния 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 Кузнечнику, чтобы собрать наибольшее количество золотых монет. Учтите, что Кузнечик не может прыгать назад.

**Входные данные**  
В первой строке вводятся два натуральных числа:  $n$  и  $k$  ( $3 \leq n \leq 10\,000$ ,  $1 \leq k \leq 10\,000$ ), 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.

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

Если правильных ответов несколько, выведите любой из них.

<b>Примеры</b>	
<b>входные данные</b>	<a href="#">Скопировать</a>
5 3 2 -3 5	
<b>выходные данные</b>	<a href="#">Скопировать</a>
7 3 1 2 4 5	
<b>входные данные</b>	
10 3 -13 -2 -14 -124 -9 -6 -5 -7	
<b>выходные данные</b>	<a href="#">Скопировать</a>
-16 4 1 3 6 8 10	
<b>входные данные</b>	
12 5 -5 -4 -3 -2 -1 1 2 3 4 5	<a href="#">Скопировать</a>
<b>выходные данные</b>	<a href="#">Скопировать</a>
14 7 1 6 7 8 9 10 11 12	

#### В. Черепаха и монеты

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Черепаха хочет переползти из левого верхнего угла поля размером  $n$  на  $m$  клеток ( $2 \leq n, m \leq 1000$ ) в правый нижний. За один шаг она может переместиться на соседнюю клетку вправо или на соседнюю клетку вниз. Кроме того, проходя через каждую клетку, Черепаха получает (или теряет) несколько золотых монет (это число известно для каждой клетки).

Определите, какое максимальное количество монет может собрать Черепаха по пути и как ей нужно идти для этого.

**Входные данные**  
В первой строке вводятся два натуральных числа:  $n$  и  $m$  ( $2 \leq n, m \leq 1000$ ), разделённые пробелом. В каждой из следующих  $n$  строк записаны через пробел  $m - 1$  целых чисел  $a_{ij}$  ( $|a_{ij}| \leq 10$ ), которые обозначают количество монет, получаемых Черепахой при проходе через каждую клетку. Если это число отрицательное, Черепашка теряет монеты.

**Выходные данные**  
В первой строке программа должна вывести наибольшее количество монет, которое может собрать Черепаха. Во второй строке без пробелов выводятся команды, которые нужно выполнить Черепахе: буква 'R' (от слова right) обозначает шаг вправо, а буква 'D' (от слова down) — шаг вниз.

<b>Примеры</b>	
<b>входные данные</b>	<a href="#">Скопировать</a>
3 3 0 2 -3 2 -5 7 1 2 0	
<b>выходные данные</b>	<a href="#">Скопировать</a>
6 RRDD	
<b>входные данные</b>	
4 5 4 5 3 2 9 4 6 7 5 9 5 2 5 -3 -10 3 5 2 9 3	
<b>выходные данные</b>	<a href="#">Скопировать</a>
41 RRDDRRR	

#### С. Наибольшая возрастающая подпоследовательность

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

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.

**Входные данные**  
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.

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

<b>Примеры</b>	
<b>входные данные</b>	<a href="#">Скопировать</a>
8 1 4 1 5 3 3 4 2	
<b>выходные данные</b>	<a href="#">Скопировать</a>
3 1 4 5	
<b>входные данные</b>	
3 1 2 3	<a href="#">Скопировать</a>
<b>выходные данные</b>	<a href="#">Скопировать</a>
3 1 2 3	

#### Д. Ход конем

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

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  $10^9$ .

**Входные данные**  
Во входном файле записано целое число  $n$  ( $1 \leq n \leq 100$ ).

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

<b>Примеры</b>	
<b>входные данные</b>	<a href="#">Скопировать</a>
1	
<b>выходные данные</b>	<a href="#">Скопировать</a>
8	
<b>входные данные</b>	
2	<a href="#">Скопировать</a>
<b>выходные данные</b>	<a href="#">Скопировать</a>
16	

#### Е. Расстояние по Левенштейну

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

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.

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

**Выходные данные**  
Требуется вывести одно число — расстояние Левенштейна для данных строк.

<b>Пример</b>	
<b>входные данные</b>	<a href="#">Скопировать</a>
ABCEDEFGH ACDEKXGIN	
<b>выходные данные</b>	<a href="#">Скопировать</a>
3	

#### Ф. Кафе

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 64 мегабайта  
ввод: стандартный ввод  
вывод: стандартный вывод

Около Петинго 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 прейскурант 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.

**Входные данные**  
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.

**Выходные данные**  
В первой строке выведите минимальную возможную суммарную стоимость обедов. Во второй строке выведите 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 выведите в возрастающем порядке номера дней, когда Пете следует воспользоваться купонами. Если существует несколько решений с минимальной суммарной стоимостью, то выведите то из них, в котором значения  $k_1$  максимально (на случай, если Пете когда-нибудь ещё решит заглянуть в это кафе). Если таких решений несколько, выведите любое из них.

<b>Примеры</b>	
<b>входные данные</b>	<a href="#">Скопировать</a>
5 110 40 120 110 60	
<b>выходные данные</b>	<a href="#">Скопировать</a>
260 0 2 3 5	
<b>входные данные</b>	
3 110 110 110	<a href="#">Скопировать</a>
<b>выходные данные</b>	<a href="#">Скопировать</a>
220 1 1 2	

#### Г. Удаление скобок 2.0

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

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.

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

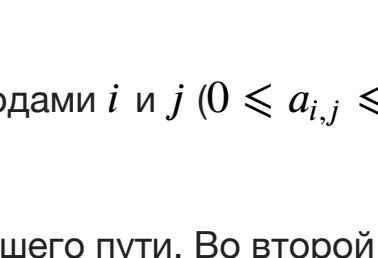
**Выходные данные**  
Выведите строку максимальной длины, являющейся правильной скобочной последовательностью, которую можно получить из исходной строки удалением некоторых символов.

<b>Пример</b>	
<b>входные данные</b>	<a href="#">Скопировать</a>
{[()]}	
<b>выходные данные</b>	<a href="#">Скопировать</a>
{}	

#### Н. Продавец аквариумов

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

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



**Входные данные**  
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  $j$  ( $0 \leq a_{i,j} \leq 10^9$ ;  $a_{i,i} = a_{j,i} = a_{i,j} = 0$ ).

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

<b>Примеры</b>	
<b>входные данные</b>	<a href="#">Скопировать</a>
5 0 183 163 173 181 183 0 165 172 171 163 165 0 189 392 173 172 189 0 167 181 171 392 167 0	
<b>выходные данные</b>	<a href="#">Скопировать</a>
666 4 5 2 3 1	

#### І. Замощение доминошками

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Вам дана прямоугольная сетка из квадратных ячеек. Символ 'X' представляет собой уже покрытую ячейку, символ '.' — это ячейка, которую еще нужно покрыть.

Вы хотите покрыть все ячейки '.', перекрывающимися доминошками  $2 \times 1$ . Найдите количество способов сделать это. Два способа считаются различными, если две ячейки покрыты одним и тем же домино в одном замощении и двумя разными домино в другом замощении.

**Входные данные**  
Первая строка содержит числа  $n$  и  $m$  ( $1 \leq n, m \leq 12$ ). Следующие  $n$  строк содержат по  $m$  символов — описание доски.

**Выходные данные**  
Выведите число замощений доминошками.

<b>Примеры</b>	
<b>входные данные</b>	<a href="#">Скопировать</a>
2 2 .. ..	
<b>выходные данные</b>	<a href="#">Скопировать</a>
2	
<b>входные данные</b>	
3 3 .. ..X ...	<a href="#">Скопировать</a>
<b>выходные данные</b>	<a href="#">Скопировать</a>
2	

<b>входные данные</b>	
3 3 ... ...X ...X	<a href="#">Скопировать</a>
<b>выходные данные</b>	<a href="#">Скопировать</a>
4	

#### Ј. Симпатичные узоры

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

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  $n$  на  $m$  метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезные проблемы: во первых, каждый 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 показалось исследованию, 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!

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

**Выходные данные**  
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.

<b>Примеры</b>	
<b>входные данные</b>	<a href="#">Скопировать</a>
1 1	
<b>выходные данные</b>	<a href="#">Скопировать</a>
2	
<b>входные данные</b>	
1 2	<a href="#">Скопировать</a>
<b>выходные данные</b>	<a href="#">Скопировать</a>
4	

#### К. Мультирюкзак

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Один рюкзак может вместить предметы с суммарным весом не более  $s$  ( $1 \leq s \leq 10^9$ ). Есть  $n$  предметов, с весами  $w_i$  ( $1 \leq w_i \leq s$ ).

Найдите способ разместить предметы в минимальное число рюкзаков.

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

Следующие  $n$  строк содержат веса предметов  $w_i$  ( $1 \leq w_i \leq s$ ).

**Выходные данные**  
Первая строка должна содержать число  $k$  — минимально число рюкзаков.

Каждая из следующих  $k$  строк описывает один рюкзак. Строка начинается с количества предметов, далее содержатся номера коров.

<b>Примеры</b>	
<b>входные данные</b>	<a href="#">Скопировать</a>
4 10 5 6 3 7	
<b>выходные данные</b>	<a href="#">Скопировать</a>
3 2 1 3 1 2 1 4	