



## Nice, Nice2. Симпатичные узоры — 1, 2

Имя входного файла: nice.in, nice2.in  
Имя выходного файла: nice.out, nice2.out

Компания «BrokenTiles» планирует заняться выкладыванием во дворах у состоятельных клиентов узоров из черных и белых плиток, каждая из которых имеет размер  $1 \times 1$  метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодняшний день 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, конечно же, 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  $m$ .

### Варианты

Nice:  $1 \leq n \cdot m \leq 30$ .

Nice2:  $1 \leq n \cdot m \leq 250$ , ответ выводить по модулю  $10^9$ .

### 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

nice.in, nice2.in	nice.out, nice2.out
2 2	14
3 3	322

## Numbers. Числа

Имя входного файла: numbers.in  
Имя выходного файла: numbers.out

Дана последовательность чисел  $a_1, a_2, \dots, a_N$ . За одну операцию разрешается удалить любое (кроме крайних) число, заплатив за это штраф, равный произведению этого числа и суммы соседних. Требуется удалить все числа, кроме крайних, с минимальным суммарным штрафом.

Например:

Начальная последовательность:

1 50 51 50 1

удаляем четвертое число, штраф —  $50 \cdot (1 + 51) = 2600$ , получаем

1 50 51 1

удаляем третье число, штраф —  $51 \cdot (50 + 1) = 2601$ , получаем

1 50 1

удаляем второе число, штраф —  $50 \cdot (1 + 1) = 100$ .

Итого штраф — 5301.

### Формат входного файла

В первой строке входного файла расположено одно число  $N$  ( $1 \leq N \leq 100$ ) — количество чисел в последовательности.

Во второй строке находятся  $N$  целых чисел  $a_1, a_2, \dots, a_N$ ; никакое из чисел не превосходит по модулю 100.

### Формат выходного файла

Выведите в выходной файл одно число — минимальный суммарный штраф.

### Пример

numbers.in	numbers.out
5	5301
1 50 51 50 1	



## Buratino. Буратино

Имя входного файла: buratino.in  
 Имя выходного файла: buratino.out

«Эх, дубинушка, ухнем!»

Папа Карло сменил работу: теперь он работает в мастерской, и целый рабочий день занимается тем, что забивает гвоздики. Чтобы ему было не скучно, у него в мастерской стоит постоянно работающий телевизор. К сожалению, производительность папы Карло напрямую зависит от его настроения, а оно, в свою очередь, — от того, что в данный момент показывают по телевизору. Правда, пока папа Карло забивает гвоздик, он не обращает ни малейшего внимания на телевизор, и поэтому скорость его работы зависит только от того, что показывали по телевизору в тот момент, когда он только начал забивать этот гвоздик. Забив очередной гвоздик, он обязательно мельком смотрит в телевизор (его настроение, естественно, меняется), и после этого он может либо сразу начать забивать следующий гвоздик, либо отдохнуть несколько секунд или даже минут, смотря телевизор.

Папа Карло начинает работу ровно в 9 часов. С 13 часов у него начинается обеденный перерыв. При этом если он незадолго до обеда хочет начать вбивать гвоздик, но понимает, что до перерыва он не закончит эту работу, то он и не начинает ее. Аналогично в 14 часов он вновь приступает к работе, а в 18 уходит домой. Это значит, что в 9:00:00 (аналогично, как и в 14:00:00) он уже может начать забивать гвоздик. Если, например, в 12:59:59 (аналогично, в 17:59:59) он хочет начать вбивать гвоздик, и на это у него уйдет 1 секунда, то он успевает вбить гвоздик до обеда (до окончания работы соответственно), а если 2 — то уже нет.

Известна программа телевизионных передач и то, как они влияют на папу Карло. Требуется составить график работы и маленьких перерывчиков папы Карло так, чтобы за рабочий день он вбил максимально возможное количество гвоздей.

### Формат входного файла

Во входном файле записано расписание телевизионных передач с 9:00:00 до 18:00:00 в следующем формате. В первой строке число  $N$  — количество телевизионных передач в этот период ( $1 \leq N \leq 32400$ ). В каждой из последующих  $N$  строк записано описание одной передачи: сначала время ее начала в формате ЧЧ:ММ:СС (ЧЧ — две цифры, задающие часы, ММ — две цифры, задающие минуты начала, СС — две цифры, задающие секунды начала). А затем через один или несколько пробелов число  $T_i$  — время в секундах, которое папа Карло будет тратить на забивание одного гвоздика, если он перед этим увидит по телевизору эту передачу ( $1 \leq T_i \leq 32400$ ).

Передачи записаны в хронологическом порядке. Первая передача всегда начинается в 09:00:00. Можно считать, что последняя передача заканчивается в 18:00:00.

### Формат выходного файла

В первую строку выходного файла требуется вывести максимальное количество гвоздиков, которое папа Карло успеет вбить за рабочий день.

### Пример

buratino.in	buratino.out
2 09:00:00 3600 14:00:00 3600	8
4 09:00:00 1800 12:59:31 10 13:45:23 1800 15:00:00 3600	14

### Замечание

В первом примере Папа Карло вбивает по одному гвоздику каждый час.

Во втором примере в первую половину дня он вбивает по гвоздику за полчаса, но в 12:30:00 он не начинает вбивать гвоздики, а ждет 12:59:31, и успевает до обеда вбить 2 гвоздика. С 14 до 15 часов вбиваются 2 гвоздя, а затем по одному гвоздю в час.



## Network. Сеть

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

В компьютерной сети вашей фирмы  $n$  компьютеров. В последнее время свитч, к которому они подключены, сильно барахлит, и потому не любые два компьютера могут связаться друг с другом. Кроме того, если компьютер  $a$  обменивается информацией с компьютером  $b$ , то никакие другие компьютеры не могут в это время обмениваться информацией ни с  $a$ , ни с  $b$ . Вам необходимо вычислить максимальное количество компьютеров, которые могут одновременно участвовать в процессе обмена информацией.

### Формат входного файла

В первой строке файла задано целое число  $n$  ( $1 \leq n \leq 18$ ). Далее идут  $n$  строк по  $n$  символов, причем  $j$  символ  $i$ -й строки равен 'Y', если  $i$ -й и  $j$ -й компьютеры могут обмениваться информацией, иначе он равен 'N'.  $i$ -й символ  $i$ -й строки всегда равен 'N', кроме того, матрица символов симметрична.

### Формат выходного файла

Выведите максимальное количество компьютеров, которые могут одновременно участвовать в процессе обмена информацией.

### Пример

<code>network.in</code>	<code>network.out</code>
5 NYYYY YNNNN YNNNY YNNNY YNYYN	4

## Brides. В поисках невест

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

Однажды король Флатландии решил отправить  $k$  своих сыновей на поиски невест. Всем известно, что во Флатландии  $n$  городов, некоторые из которых соединены дорогами. Король живет в столице, которая имеет номер 1, а город с номером  $n$  знаменит своими невестами.

Итак, король повелел, чтобы каждый из его сыновей добрался по дорогам из города 1 в город  $n$ . Поскольку, несмотря на обилие невест в городе  $n$ , красивых среди них не так много, сыновья опасаются друг друга. Поэтому они хотят добраться до цели таким образом, чтобы никакие два сына не проходили по одной и той же дороге (даже в разное время). Так как король любит своих сыновей, он хочет, чтобы среднее время сына в пути до города назначения было минимально.

### Формат входного файла

В первой строке входного файла находятся числа  $n$ ,  $m$  и  $k$  — количество городов и дорог во Флатландии и сыновей короля, соответственно ( $2 \leq n \leq 200$ ,  $1 \leq m \leq 2000$ ,  $1 \leq k \leq 100$ ). Следующие  $m$  строк содержат по три целых положительных числа каждая — города, которые соединяет соответствующая дорога и время, которое требуется для ее прохождения (время не превышает  $10^6$ ). По дороге можно перемещаться в любом из двух направлений, два города могут быть соединены несколькими дорогами.

### Формат выходного файла

Если выполнить повеление короля невозможно, выведите на первой строке число  $-1$ . В противном случае выведите на первой строке минимальное возможное среднее время, которое требуется сыновьям, чтобы добраться до города назначения (не менее, чем с пятью знаками после десятичной точки). В следующих  $k$  строках выведите пути сыновей: сначала число дорог в пути и затем номера дорог в пути в том порядке, в котором их следует проходить. Дороги нумеруются, начиная с единицы, в том порядке, в котором они заданы во входном файле.

### Пример

<code>brides.in</code>	<code>brides.out</code>
5 8 2	3.00000
1 2 1	3 1 5 6
1 3 1	3 2 7 8
1 4 3	
2 5 5	
2 3 1	
3 5 1	
3 4 1	
5 4 1	