

## Задача А. ЗОРП 2

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

Перед вами лежат  $n$  котиков. Каждый котик характеризуется своим весом  $w_i$  и своей мимимишностью  $c_i$ . Вы хотите выбрать некоторое число котиков суммарным весом не более чем  $S$  так, чтобы их суммарная мимимишность была максимально возможной.

### Формат входных данных

В первой строке содержатся два целых числа  $n$  и  $S$  — число котиков и максимальный допустимый суммарный вес ( $1 \leq n \leq 100$ ,  $1 \leq S \leq 10^9$ ). Следующие  $n$  строк содержат по два целых числа  $w_i$  и  $c_i$  — вес и мимимишность  $i$ -го котика ( $1 \leq w_i \leq 10^7$ ,  $0 \leq c_i \leq 10^4$ ). Гарантируется, что сумма всех  $c_i$  не превосходит  $10^4$ .

### Формат выходных данных

В первой строке выведите суммарную мимимишность выбранных котиков. Во вторую строку выведите целое число  $k$  — количество выбранных котиков. В третьей строке выведите  $k$  чисел — номера выбранных котиков. Если оптимальных ответов несколько, то разрешается вывести любой из них.

### Примеры

knapsack-2.in	knapsack-2.out
3 10	11
1 2	2
4 3	3 1
8 9	

## Задача В. Мостостроение

Имя входного файла: bridge.in  
Имя выходного файла: bridge.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Давным давно, в 2009-м году...

---

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

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

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

Главный инженер хочет построить мост максимальной возможной ширины. Ширина моста определяется по минимальной ширине ряда брёвен в нём.

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

### Формат входных данных

Ввод состоит из одного или нескольких тестовых случаев. Каждый тестовый случай состоит из пяти целых положительных чисел  $x$ ,  $a$ ,  $y$ ,  $b$  и  $l$ . Каждое число не превосходит 500. Общее количество брёвен в каждом тестовом случае не меньше  $l$ .

Обозначим  $d = \max(x, a, y, b, l)$ . Гарантируется, что сумма  $d$  по всем тестам не превосходит 5000.

### Формат выходных данных

#### Примеры

bridge.in	bridge.out
6 3 10 2 7	5
10 7 20 9 25	9
106 126 135 28 137	112

## Задача С. Редукция дерева

Имя входного файла: `tree.in`  
Имя выходного файла: `tree.out`  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Задано неориентированное дерево, содержащее  $n$  вершин. Можно выбрать некоторое ребро и удалить его, при этом инцидентные ему вершины не удаляются. Таким образом можно удалить из дерева некоторый набор рёбер. В результате дерево распадается на некоторое количество меньших деревьев. Требуется, удалив наименьшее количество рёбер, получить в качестве хотя бы одной из компонент связности дерево, содержащее ровно  $p$  вершин.

### Формат входных данных

Первая строка входного файла содержит пару натуральных чисел  $n$  и  $p$  ( $1 \leq p \leq n \leq 1000$ ). Далее в  $n - 1$  строке содержатся описания рёбер дерева. Каждое описание состоит из пары натуральных чисел  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n$ ) — номеров соединяемых ребром вершин.

### Формат выходных данных

В первую строку выведите наименьшее количество рёбер  $q$  в искомом наборе.

### Примеры

<code>tree.in</code>	<code>tree.out</code>
11 6 1 2 1 3 1 4 2 6 2 7 1 5 2 8 4 9 4 10 4 11	2

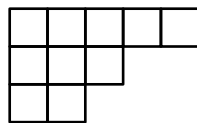
## Задача D. Увидеть Юнга и умереть

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

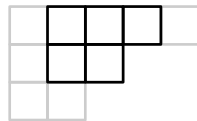
Диаграммы Юнга используются для того, чтобы изобразить разбиение числа на слагаемые. Разбиение числа  $n$  на слагаемые представляет собой сумму вида  $n = m_1 + m_2 + \dots + m_k$ , где  $m_1 \geq m_2 \geq \dots \geq m_k$ .

Диаграмма состоит из  $n$  квадратиков, организованных в виде  $k$  рядов, где  $k$  количество слагаемых в разбиении. Ряд, соответствующий числу  $m_i$ , содержит  $m_i$  квадратиков. Все ряды выровнены по левому краю и упорядочены от более длинного к более короткому.

Например, диаграмма Юнга, приведенная на рисунке, соответствует разбиению  $10 = 5 + 3 + 2$ .



Иногда можно вписать одну диаграмму Юнга в другую. Диаграмму  $X$  можно вписать в диаграмму  $Y$ , если можно удалить некоторые квадратики из диаграммы  $Y$  так, чтобы получилась диаграмма  $X$ . Отметим, что разрешается только удалять некоторые квадратики, вращать или отражать диаграмму не разрешается. Например, диаграмма для разбиения  $5 = 3 + 2$  может быть вписана в диаграмму для разбиения  $10 = 5 + 3 + 2$ , как показано на рисунке.



С другой стороны, диаграмму для разбиения  $8 = 4 + 4$  нельзя вписать в диаграмму для разбиения  $10 = 5 + 3 + 2$ .

Для заданного  $n$  найдите такое разбиение  $n$  на слагаемые, что в соответствующую ему диаграмму Юнга можно вписать максимальное количество различных диаграмм.

Например, в диаграмму для разбиения  $10 = 5 + 3 + 2$  можно вписать 36 различных диаграмм. Однако это не максимальное значение. В диаграмму для разбиения  $10 = 4 + 2 + 2 + 1 + 1$  можно вписать 41 диаграмму Юнга.

### Формат входных данных

Входной файл содержит целое число  $n$  ( $1 \leq n \leq 100$ ).

### Формат выходных данных

На первой строке выходного файла выведите максимальное число диаграмм Юнга, которые можно вписать в некоторую диаграмму, соответствующую разбиению на слагаемые числа  $n$ .

На второй строке выведите одно или более целых чисел — количество квадратиков в каждом из рядов оптимальной диаграммы.

### Примеры

<code>young.in</code>	<code>young.out</code>
10	41 4 3 2 1

## Задача Е. Транзисторы над Пекином

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

Всемирно известный профессор В.В. Адимов продолжает свои разнообразные исследование устойчивости транзисторов. Теперь в голову ему пришла следующая задача: в доме  $N$  этажей, профессор хочет выяснить номер максимального этажа, падение с которого оставляет транзистор целым. Поскольку профессор исследует сферические транзисторы в вакууме, то можете считать что разбившись при падении с этажа  $f$  транзистор обязательно разобьется при падении с этажа  $f + 1$ . Дополнительно поставлено условие, что разрешено проведение не более чем  $K$  испытаний.

Эта задача была поручена именно вам, как самому успешному аспиранту профессора Адимова. Поскольку транзисторы нынче в цене, но наука все-таки дороже, то необходимо выяснить, какое минимальное количество транзисторов необходимо закупить, чтобы успешно провести эксперимент даже если вам будет катастрофически не везти.

### Формат входных данных

В первой и единственной строке входного файла содержатся два целых положительных числа  $N$  и  $K$  не превосходящих 2000.

### Формат выходных данных

Выведите единственное число - ответ на поставленную задачу. Если для данных  $N$  и  $K$  возможна ситуация, при которой мы не сможем получить ответ на вопрос даже имея неограниченный запас бесплатных транзисторов выведите  $-1$ .

### Примеры

transistor.in	transistor.out
4 2	-1
4 3	2