

## Задача А. Банкомат

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

В некотором государстве в обращении находятся банкноты определённых номиналов. Национальный банк хочет, чтобы банкомат выдавал любую запрошенную сумму при помощи минимального числа банкнот, считая, что запас банкнот каждого номинала неограничен. Помогите Национальному банку решить эту задачу.

### Формат входных данных

Первая строка входных данных содержит натуральное число  $N$  не превосходящее 100 — количество номиналов банкнот в обращении. Вторая строка входных данных содержит  $N$  различных натуральных чисел  $x_1, x_2, \dots, x_N$ , не превосходящих  $10^4$  — номиналы банкнот. Третья строка содержит натуральное число  $S$ , не превосходящее  $10^4$  — сумму, которую необходимо выдать.

### Формат выходных данных

В первую строку выходного файла выведите минимальное число слагаемых (или -1, если такого представления не существует). Во вторую строку выведите это представление в любом порядке.

### Примеры

<code>stdin</code>	<code>stdout</code>
5 1 3 7 12 32 40	3 32 7 1

## Задача В. Рюкзак

Имя входного файла: knapsack.in  
Имя выходного файла: knapsack.out  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 64 мегабайта

Найдите максимальный вес золота, который можно унести в рюкзаке вместительностью  $S$ , если есть  $N$  золотых слитков с заданными весами.

### Формат входных данных

В первой строке входного файла записаны два целых числа —  $S$  и  $N$  ( $1 \leq S \leq 10\,000$ ,  $1 \leq N \leq 200$ ). Далее следует  $N$  неотрицательных целых чисел, не превосходящих 100 000 — веса слитков.

### Формат выходных данных

Выведите искомый максимальный вес.

### Примеры

knapsack.in	knapsack.out
10 3 1 4 8	9
20 4 5 7 12 18	19

## Задача С. Рюкзак-2

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Найдите максимальную цену слитков золота, которые можно унести в рюкзаке вместительностью  $S$ , если есть  $N$  золотых слитков с заданными весами и ценами.

### Формат входных данных

В первой строке входных данных записаны два числа —  $S$  и  $N$  ( $1 \leq S \leq 10\,000$ ,  $1 \leq N \leq 300$ ).

В двух следующих строках записано по  $N$  неотрицательных целых чисел в каждой — веса и стоимости слитков соответственно. Каждое из этих чисел не превосходит 100 000.

### Формат выходных данных

Определите набор предметов максимальной стоимости, помещающийся в данный рюкзак. В первой строке выведите стоимость предметов в набранном наборе, во второй — количество предметов в наборе. В следующей строке выведите через пробел номера этих предметов.

### Примеры

<code>stdin</code>	<code>stdout</code>
10 3	123
1 4 8	2
72 7 51	1 3

## Задача D. Сумма кубов

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

Дано натуральное число  $N$ . Необходимо представить его в виде суммы точных кубов натуральных чисел, содержащей наименьшее число слагаемых. Программа должна вывести это число слагаемых.

### Формат входных данных

Программа получает на вход натуральное число  $N$ , не превосходящее  $2^{15} - 1$ .

### Формат выходных данных

Программа должна вывести единственное натуральное число.

### Примеры

<code>cubes.in</code>	<code>cubes.out</code>
33	5
1	1

## Задача Е. Получите в сумме ноль

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 146 мегабайт

Даны целые неотрицательные числа  $a_1, a_2, \dots, a_n$ . Определите, какой знак следует поставить перед каждым из них, чтобы результат сложения данных чисел с учётом выбранных знаков равнялся нулю. Гарантируется, что способ расставить знаки таким образом существует.

### Формат входных данных

В первой строке записано целое число  $n$  ( $1 \leq n \leq 100$ ). Во второй строке через пробел записаны числа  $a_i$  ( $0 \leq a_i \leq 500$ ).

### Формат выходных данных

Выведите единственную строку из  $n$  символов.  $i$ -й символ строки должен соответствовать знаку, который требуется поставить в сумме перед  $i$ -м числом, чтобы получить в результате ноль.

### Примеры

stdin	stdout
4 2 3 0 1	-+--

## Задача F. Федя и вопросы «Своей игры»

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Как известно, в ЛКШ.2013.Июль «Свою игру» проводит Федя. Только что обнаружилось, что для предстоящего турнира ему не хватает одной темы. Тема в «Своей игре» — это набор ровно из пяти вопросов, лежащих в одной области, причем каждый следующий вопрос строго сложнее предыдущего. К счастью, у Феде уже есть пакет из  $N$  готовых вопросов из области «Исторические личности Берляндии». Для каждого из вопросов известна его сложность (натуральное число). Посчитайте, сколько у Феде способов собрать из вопросов пакета одну тему. Два способа считаются различными, если хотя бы один вопрос, присутствующий в одном из них, не присутствует во втором.

### Формат входных данных

В первой строке файла находится одно число  $N$  — количество вопросов в наборе Феде ( $5 \leq N \leq 300$ ). В следующей строке  $N$  чисел, отсортированных по неубыванию — сложности вопросов из пакета. Каждая сложность — натуральное число от 1 до 100, сложности некоторых вопросов могут совпадать.

### Формат выходных данных

Выведите одно число — количество способов сделать тему для «Своей игры».

### Примеры

<code>stdin</code>	<code>stdout</code>
7 1 1 2 3 4 5 5	4
7 1 2 3 4 5 5 6	11
5 1 1 2 3 4	0

## Задача G. Рюкзак 0-1: минимум предметов

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

Дано  $N$  предметов массой  $m_1, \dots, m_N$ . Ими наполняют рюкзак, который выдерживает вес не более  $M$ . Как набрать вес в точности  $M$ , используя как можно меньше предметов?

### Формат входных данных

В первой строке вводится натуральное число  $N$ , не превышающее 100 и натуральное число  $M$ , не превышающее 10000.

Во второй строке вводятся  $N$  натуральных чисел  $m_i$ , не превышающих 100.

### Формат выходных данных

Выведите наименьшее необходимое число предметов или 0, если набрать данный вес невозможно.

### Примеры

стандартный ввод	стандартный вывод
1 5968 18	0

## Задача Н. ЗОРП 2

Имя входного файла: knapsack-2.in  
Имя выходного файла: knapsack-2.out  
Ограничение по времени: 4 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 I. Гладиолус

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

Числовая последовательность задана рекуррентной формулой:  $a_{i+1} = (ka_i + b) \bmod m$ . Найдите её наибольшую возрастающую подпоследовательность.

### Формат входных данных

Программа получает на вход пять целых чисел: длину последовательности  $n$  ( $1 \leq n \leq 10^5$ ), начальный элемент последовательности  $a_1$ , параметры  $k$ ,  $b$ ,  $m$  для вычисления последующих членов последовательности ( $1 \leq m \leq 10^4$ ,  $0 \leq k < m$ ,  $0 \leq b < m$ ,  $0 \leq a_1 < m$ ).

### Формат выходных данных

Требуется вывести наибольшую возрастающую подпоследовательность данной последовательности, разделяя числа пробелами. Если таких последовательностей несколько, необходимо вывести одну (любую) из них.

### Примеры

<code>gladiolus.in</code>	<code>gladiolus.out</code>
5 41 2 1 100	41 67 71

### Замечание

В данном примере последовательность состоит из 5 элементов:  $a_1 = 41$ ,  $a_{i+1} = (2a_i + 1) \bmod 100$ , то есть последовательность имеет вид 41, 83, 67, 35, 71.