

Задача А. Гирьки

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

Дан набор гирек массой $m_1, \dots, m_n$. Можно ли их разложить на две чаши весов таким образом, чтобы они оказались в равновесии?

Формат входных данных

Первая строка входных данных содержит натуральное число n , не превышающее 100. Далее идет n натуральных чисел m_i , не превышающих 100.

Формат выходных данных

Программа должна вывести «yes», если гирьки можно разложить на две кучки равной массы, или «no» в противном случае.

Пример

стандартный ввод	стандартный вывод
4 4 2 3 1	YES

Задача В. ЗОРП 2

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
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.

Пример

стандартный ввод	стандартный вывод
3 10	11
1 2	2
4 3	3 1
8 9	

Задача С. Рюкзак максимальной стоимости

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

Дано n предметов массой $m_1, \dots, m_n$ и стоимостью $c_1, \dots, c_n$ соответственно. Ими наполняют рюкзак, который выдерживает вес не более w . Определите набор предметов, который можно унести в рюкзак, имеющий наибольшую стоимость.

Формат входных данных

В первой строке вводится натуральное число n , не превышающее 100, и натуральное число w , не превышающее 10000.

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

В третьей строке вводятся n натуральных чисел c_i , не превышающих 100.

Формат выходных данных

В первой строке выведите число k — количество элементов в рюкзаке наибольшей стоимости. Во второй строке выведите k различных чисел от 1 до n — номера предметов.

Пример

стандартный ввод	стандартный вывод
4 6 2 4 1 2 7 2 5 1	3 4 3 1

Задача Е. Большой рюкзак

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

У вас есть N предметов с весами $w_1, w_2, \dots, w_N$. Требуется проверить, можно ли выбрать некоторые предметы, суммарный вес которых равен W .

Формат входных данных

В первой строке записаны два целых числа N и W ($1 \leq N \leq 2\,500$, $1 \leq W \leq 6\,250\,000$).

Во второй строке через пробел записаны N целых чисел $w_1, w_2, \dots, w_N$ ($1 \leq w_i \leq 2\,500$) — веса предметов.

Формат выходных данных

В единственной строке выведите «YES» (без кавычек), если, используя данные предметы, можно набрать вес W , либо «NO» в противном случае.

Примеры

стандартный ввод	стандартный вывод
5 10 1 2 3 4 5	YES
2 10 4 5	NO

Задача F. Двусторонний рюкзак

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

Даны n предметов с **целыми** положительными весами и **вещественными** положительными стоимостями. И рюкзак веса W . Выберите, какие предметы положить в рюкзак, чтобы их суммарный вес не превосходил W , а суммарная стоимость была максимальна?

Но есть загвоздка, один из предметов брать запрещено. Какой именно мы не знаем, поэтому хотим найти ответ для всех возможных номеров i запрещённого предмета.

Формат входных данных

На первой строке два целых числа n ($2 \leq n \leq 2000$) и W ($1 \leq W \leq 1000$). Следующие n строк содержат пары чисел — вес w_i ($1 \leq w_i \leq 10$) и стоимость $cost_i$ ($1 \leq cost_i \leq 10$).

Формат выходных данных

Выведите n чисел. i -е число — суммарная стоимость предметов, которые можно набрать в рюкзак, если i -й предмет брать запрещено.

Пример

стандартный ввод	стандартный вывод
4 10	13.100000000
5 5.5	11.600000000
4 7	12.500000000
1 2	14.500000000
2 4.1	

Замечание

- 13.1 — запрещено брать (5, 5.5), оптимальный ответ $7 + 4.1$.
- 11.6 — запрещено брать (4, 7), оптимальный ответ $5.5 + 4.1 + 2$.
- 12.5 — запрещено брать (1, 2), оптимальный ответ $5.5 + 7$.
- 14.5 — запрещено брать (2, 4.1), оптимальный ответ $7 + 5.5 + 2$.

Задача G. Гирьки: кучки одного размера

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

Дан набор гирек массой $m_1, \dots, m_N$. Разделите этот набор на две кучки равной массы, содержащие равное число гирек.

Формат входных данных

Первая строка входных данных содержит натуральное число N , не превышающее 100.

В следующей строке через пробел записаны N натуральных чисел m_i , не превышающих 100.

Формат выходных данных

Необходимо вывести в первой строчке номера гирек (числа от 1 до N), входящие в первую кучку, во второй строчке — номера гирек во второй кучке.

Если задача не имеет решения, выведите одно число -1.

Пример

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

Задача К. Распределение задач

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 512 мегабайт

На работе разработчики разделяют часть проекта на задачи. У каждой задачи есть целая, возможно отрицательная, характеристика h_i — количество удовольствия, которое принесёт эта задача разработчику, который будет ее выполнять.

Задачи внутри проекта могут зависеть от других задач, при этом циклических зависимостей нет. *Готовый проект* — финальная задача проекта, которая напрямую или через другие задачи зависит от всех остальных задач проекта. Иными словами, готовый проект — задача, от которой **не зависит** никакая другая задача. Для всех остальных задач верно, что напрямую от них зависит только одна единственная задача.

Катя — лидер команды из трёх человек, которой поручено выполнить очередной проект. Чтобы разграничить зоны ответственности было решено разделить проект на 3 части так, чтобы минимизировать количество зависимостей между задачами разных разработчиков. Для этого необходимо, чтобы существовало ровно **две** задачи таких, что разработчик занимающийся этой задачей не совпадал с разработчиком, который занимается задачей, зависящей от этой задачи. Также Катя хочет, чтобы суммарное удовольствие, полученное от выполнения задачи каждым из разработчиков было одинаковым.

Помогите команде и скажите, можно ли разделить все задачи проекта так, чтобы выполнялись все требования.

Формат входных данных

В первой строке входных данных дано число n — количество задач в проекте ($1 \leq n \leq 5 \cdot 10^5$).

Во второй строке дано n чисел — число p_i задает номер задачи, от которой зависит задача i , или 0, если задача с номером i является готовым проектом ($0 \leq p_i \leq n$). Гарантируется, что существует единственный i , у которого $p_i = 0$.

Во третьей строке дано n чисел — число с номером i задает количество удовольствия, которое будет получено при выполнении задачи ($-100 \leq h_i \leq 100$).

Формат выходных данных

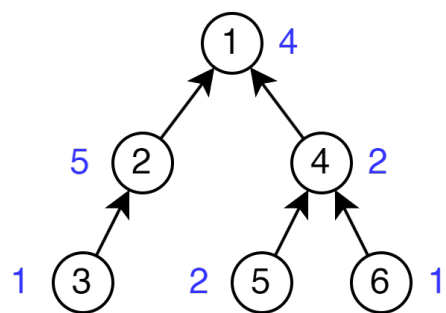
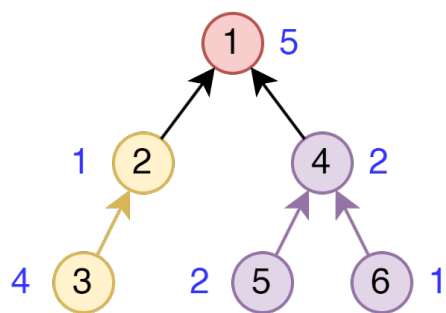
В единственной строке выведите «YES», если все задачи проекта можно разделить на 3 части так, чтобы выполнялись все требования, и «NO» иначе.

Примеры

стандартный ввод	стандартный вывод
6 0 1 2 1 4 4 5 1 4 2 2 1	YES
6 0 1 2 1 4 4 4 5 1 2 2 1	NO

Замечание

Ниже приведен пример распределения задач с зависимостями из первого и второго примеров из условий. Синим цветом рядом с каждой задачей обозначена величина h_i . Стрелка от задачи a к задаче b означает, что задача b зависит от задачи a .



Задача L. Выбор команды

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

В олимпиадной школе учатся n учеников, которые занимаются программированием. У каждого ученика есть три основных навыка — умение придумывать решение задачи, умение быстро и качественно писать код, умение тестировать написанную программу. Эти навыки характеризуются целыми неотрицательными числами x, y и z соответственно (чем больше число, тем лучше развит навык). Директор школы хочет собрать команду школьников из k человек для успешного выступления на командной олимпиаде по программированию. В выбранной команде a человек будут придумывать решение каждой задачи, b человек будут по очереди писать код, а c человек будут тестировать написанные программы.

Показатель эффективности команды — сумма трёх чисел:

1. сумма показателей x среди a школьников, которые придумывают решение;
2. сумма показателей y среди b школьников, которые пишут код;
3. сумма показателей z среди c школьников, которые тестируют программу.

Требуется собрать команду так, чтобы её показатель эффективности был максимален.

Формат входных данных

В первой строке даны два числа n и k ($3 \leq n \leq 10^5$, $3 \leq k \leq \min(n, 15)$).

Во второй строке — три числа a, b и c ($a + b + c = k$, $a \geq 1$, $b \geq 1$, $c \geq 1$).

Далее идут n строк, в которых перечисляются характеристики каждого из учеников. В $i + 2$ -й строке записаны неотрицательные числа $0 \leq x_i, y_i, z_i \leq 10^5$.

Формат выходных данных

Требуется вывести одно число — максимальный показатель эффективности.

Пример

стандартный ввод	стандартный вывод
3 3 1 1 1 5 4 10 1 3 9 2 7 16	24

Задача М. Balatro

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 6 секунд
Ограничение по памяти: 256 мегабайт

Нужно ли действительно объяснять правила *Balatro*? Просто сыграйте сами — и не забудьте остановиться до начала следующего контеста.

Для тех, кто не знаком с правилами: у вас есть n карт на руках. Каждая карта имеет два связанных значения: a_i и b_i .

Вы можете выбрать любое подмножество S ровно из k карт и сыграть их вместе, чтобы получить очки, вычисляемые по формуле:

$$\left(\sum_{i \in S} a_i\right) \cdot \left(\sum_{i \in S} b_i\right)$$

Ваша задача — определить максимальное возможное количество очков, которое можно получить за один ход, используя ровно k карт.

Известно также, что колода *сбалансирована*, то есть ни одна карта не имеет одновременно слишком больших значений a_i и b_i . Формально, для каждой карты выполняется условие:

$$\min(a_i, b_i) \leq 100$$

Формат входных данных

Каждый тест содержит несколько наборов данных. Первая строка содержит число тестов t ($1 \leq t \leq 2000$).

Описание тестов следует далее.

Каждый тест начинается строкой, содержащей два целых числа n и k ($1 \leq n \leq 10^5$, $1 \leq k \leq \min(n, 5)$) — общее количество карт и количество карт, которые нужно выбрать.

Далее следуют n строк, каждая из которых содержит два целых числа a_i и b_i ($1 \leq a_i, b_i \leq 10^9$) — значения карты. Для каждой карты гарантируется, что $\min(a_i, b_i) \leq 100$.

Суммарное количество карт во всех тестах не превышает 10^5 .

Формат выходных данных

Для каждого теста выведите одно целое число — максимальное возможное количество очков, которое можно получить, выбрав ровно k карт.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 5 5 1 1 2 2 3 3 4 4 5 5	225
1 6 5 1 1 2 6 3 5 4 4 5 3 6 2	400

Замечание

В первом примере используются все карты. Сумма равна $(1 + 2 + 3 + 4 + 5) \cdot (1 + 2 + 3 + 4 + 5) = 15^2 = 225$.

Во втором примере нужно убрать одну карту. Лучшее множество карт $[2, 3, 4, 5, 6]$ даёт значение 400.

Интересный факт: в настоящей игре *Balatro* значение a_i карты зависит от её покерного достоинства, а b_i может быть 0, 4 или 20 (значение 20 выпадает с вероятностью 20%). Кроме того, в игре существуют сотни бонусных карт и правил, влияющих на то, как вычисляются суммы a и b .