Задача A. Поедание сыра

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

На сырном заводе во Флатландии живут мыши. Они очень любят сыр и часто уничтожают запасы сыра, приготовленные для отправки в магазин.

Всего на заводе живет m мышей. Для i-й мыши известна ее скорость поедания сыра s_i, мышь может съесть s_i грамм сыра в час.

Недавно мышам стал известен план работы завода на ближайшее время. Планируется изготовить n головок сыра. Про каждую головку известны r_i — к началу какого часа она будет изготовлена, d_i — в начале какого часа она начнет портиться, и p_i — вес головки сыра в граммах.

Мыши решили съесть весь сыр. В любой момент времени каждая мышь может есть некоторую головку сыра. Мыши — существа брезгливые, и одну и ту же головку сыра не могут есть одновременно несколько мышей. При этом в любой момент времени мышь может решить прекратить есть головку сыра и приняться за другую, в том числе ту, которую ранее ела другая мышь.

Мыши не любят есть сыр после того как он начал портиться. Но оставлять сыр недоеденным мыши не могут. Они решили организовать поедание сыра таким образом, чтобы величина t, такая что какую-либо головку все еще продолжают есть через t часов после того как она начала портиться, была минимальна. Помогите мышам выяснить, как это сделать.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($1 \le n \le 30$, $1 \le m \le 30$). Следующие n строк содержит по три целых числа: p_i, r_i и d_i ($1 \le p_i \le 10^5$, $0 \le r_i < d_i \le 10^7$). Далее следуют m строк, каждая из которых содержит по одному целому числу s_j ($1 \le s_j \le 10^5$).

Формат выходного файла

Выведите одно вещественное число — искомое минимальное t. Ваш ответ должен отличаться от правильного не больше чем на 10^{-4}.

Примеры

cheese.in	cheese.out
2 2 13 0 4 10 1 3 4 2	0.50000000000000
1 1 1 0 2 1	0.00000000000000

В первом примере мышам следует организовать поедание сыра следующим образом. Сначала первая мышь начинает есть первую головку сыра. Когда появляется вторая головка, она перестает есть первую и начинает есть вторую (в этот момент от первой осталось 9 граммов). Вторая мышь принимается есть первую головку сыра. Через 2.5 часа первая мышь доедает вторую головку сыра (на 0.5 часа позже чем она начала портиться) и снова начинает есть первую (вторая мышь за это время съела еще 5 граммов от первой головки и от нее осталось 4 грамма). Таким образом еще за час первая мышь доедает первую головку, также на 0.5 часа позже чем она начала портиться.

Во втором примере мышь успевает съесть сыр до того как он начинает портиться.

Задача B. $O2 \,||\, C_{max}$

Имя входного файла:	`o2cmax.in`
Имя выходного файла:	`o2cmax.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Задача $O2 \,||\, C_{max}$. Дано два станка и n деталей. Каждую деталь нужно обработать на каждом станке. Время обработки i-й детали на первом станке — a_i, на втором — b_i. Нужно минимизировать время завершения всех работ.

Формат входного файла

В первой строке дано целое число n ($1 \le n \le 200\,000$) — количество работ. Во второй строке заданы целые числа a_i ($0 < a_i \le 10^9$) — времена, необходимые для обработки деталей на первом станке. В третьей строке заданы целые числа b_i ($0 < b_i \le 10^9$) — времена, необходимые для обработки деталей на втором станке.

Формат выходного файла

В первой строке выведите единственное целое число — минимальное время завершения всех работ. Во второй строке выведите n целых чисел $t_{1,i}$ — момент времени, в который нужно начать обрабатывать i-ю деталь на первом станке. В третьей строке аналогично выведите числа $t_{2,i}$.

Примеры

o2cmax.in	o2cmax.out
3 1 2 3 2 1 3	6 0 1 3 3 5 0

Задача C. $1 \mid outtree \mid \sum w_i c_i$

Имя входного файла:	`p1outtreewc.in`
Имя выходного файла:	`p1outtreewc.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Задача $1 \mid outtree \mid \sum w_j C_j$. Дан один станок и множество работ размером n, причем для каждой работы задано её время выполнения p_j и вес w_j. Также между работами заданы зависимости такие, что $n-1$ работ зависят ровно от одной работы, а одна работа — не зависит ни от одной. Необходимо минимизировать взвешенную сумму времен завершения работ $\sum w_j C_j$.

Формат входного файла

В первой строке дано целое число n ($1 \le n \le 50\,000$) — количество работ. Во второй строке даны n целых чисел p_j ($1 \le p_j \le 1000$) — времена выполения работ. Во третьей строке даны n целых чисел w_j ($1 \le w_j \le 1000$) — веса работ. В следующих $n-1$ строках даны пары целых чисел u и v ($1 \le u, v \le n$) — пара означает, что перед выполнением работы u необходимо выполнить работу v.

Формат выходного файла

В первой строке выведите целое число — оптимальное значение целевой функции. В следующей строке выведите произвольное оптимальное расписание — n целых чисел t_i (моменты времени, в которые нужно начать выполнять i -ю работу).

Примеры

p1outtreewc.in	p1outtreewc.out
3 1 3 2 1 6 4 2 1 3 1	49 0 1 4
4 3 4 2 1 2 3 3 2 1 2 4 3 3 2	64 7 0 4 6
7 1 2 3 4 5 6 7 7 6 5 4 3 2 1 2 1 3 1 4 2 5 2 6 3 7 3	210 0 1 3 6 10 15 21

Примечание

В первом примере также оптимально расписание "0 3 1".

Задача D. $1 \mid\mid \sum U_i$

Имя входного файла:	`p1sumu.in`
Имя выходного файла:	`p1sumu.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Задача $1 \mid\mid \sum U_i$. Дан один станок, на котором необходимо обработать n деталей. Для каждой детали известно время, необходимое для ее обработки — p_i, а также момент времени, до которого необходимо закончить работу над этой деталью — d_i. Если обработка детали закончилась после дедлайна, эта деталь считается необработанной. Нужно минимизировать количество необработанных деталей. Станок начинает работу в нулевой момент времени.

Формат входного файла

В первой строке дано целое число n ($1 \le n \le 100\,000$) — количество деталей. В следующих n строках содержатся по два числа p_i ($1 \le p_i \le 10^9$)и d_i ($1 \le d_i \le 10^9$) — время необходимое на обработку i-ой детали и дедлайн времени обработки детали, соответственно.

Формат выходного файла

В первой строке выведите единственное целое число m — количество обработанных деталей. Во второй строке выведите n целых чисел b_i, где b_i — момент времени, в который нужно начать обрабатывать i-ю деталь, если деталь не будет обработана, $b_i = -1$.

Примеры

p1sumu.in	p1sumu.out
3 1 2 2 3 3 1	2 0 1 -1

Задача E. $1 \mid p_i = 1 \mid \sum w_i U_i$

Имя входного файла:	`p1sumwu.in`
Имя выходного файла:	`p1sumwu.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Задача $1 \mid p_i = 1 \mid \sum w_i U_i$. Имеется множество из n работ $j_1, j_2, ..., j_n$. Про каждую работу извесен ее дедлайн d_i, а также стоимость опоздания w_i. Каждая работа выполняется единицу времени. Пусть U_i равна единице, если работа i выполнилась после дедлайна, и нулю в противном случае. Необходимо построить такое расписание, чтобы величина $\sum_{i=1}^{n} w_i U_i$ была минимальна.

Формат входного файла

В первой строке дано единственное натуральное число n ($1 \le n \le 200000$) — количество работ. Затем следует n строк, в каждой из которых содержится по два числа d_i и w_i ($1 \le d_i \le 200000, 1 \le w_i \le 200000$) — дедлайн и стоимость опоздания i-ой работы соответственно.

Формат выходного файла

В первой строке выведите единственное число, равное минимальной возможной сумме $\sum_{i=1}^{n} w_i U_i$. Во второй строке через пробел выведите n чисел, где i-ое число — время начала выполнения i-ой работы. Если возможно несколько оптимальных расписаний, выведите любое из них.

Примеры

p1sumwu.in	p1sumwu.out
3 1 2 1 3 3 1	2 2 0 1

Задача F. $1 \mid prec \mid f_{max}$

Имя входного файла:	`p1precfmax.in`
Имя выходного файла:	`p1precfmax.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Имеется множество из n работ $j_1, j_2, ..., j_n$ с временами выполнения $p_1, p_2, \ldots, p_n$, на котором задан граф зависимостей выполнения: если $j_a \to j_b$, то работа a должна быть выполнена *раньше* работы b. Также для каждой работы i задана не убывающая на $x \in [0, +\infty)$ функция $f_i(x) = \sum_{j=0}^{m} a_{ij}x^j$ — полином степени m. Требуется составить такое расписание, чтобы величина $f_{max} = \max(f_i(C_i))$, где C_i — время окончания выполнения работы i, была минимальна.

Формат входного файла

В первой строке дано целое число n ($1 \le n \le 1000$) — количество работ. Во второй строке дано n чисел p_i ($1 \le p_i \le 10^3$), где i-ое число — времена выполнения i-ой работы.

Каждая из следующих n строк содержит описание функций. В i-ой строке задается функция для i-ой работы. Первым в строке идет число m ($0 \le m \le 8$) — степень полинома, затем $(m + 1)$ число $a_{im}, a_{i(m-1)}, \ldots, a_{i0}$ ($0 \le a_{ij} \le 50$) — коэффициенты полинома в порядке *убывания* степеней. Гарантируется, что каждая функция не убывает на $x \in [0, +\infty)$.

На следующей строке задается число d — количество отношений, заданных на множестве работ. Каждая из следующих d строк содержит пару чисел a, b — номера работ такие, что $j_a \to j_b$. Гарантируется, что существует хотя бы одно расписание, удовлетворяющее заданным входным данным.

Формат выходного файла

В первой строке выведите единственное число — минимальное значение f_{max}.

Во второй строке через пробел выведите n чисел, где i-ое число — время начала выполнения i-ой работы. Расписание должно соответствовать минимальному значению f_{max}. Если возможно несколько расписаний, то выведите любое из них.

Примеры

p1precfmax.in	p1precfmax.out
3 1 5 4 1 9 7 1 2 2 1 1 3 3 1 2 2 3 1 3	16 0 1 6
4 1 2 3 4 1 5 4 1 4 3 1 3 3 1 2 0 5 1 2 2 4 1 3 3 4 1 4	21 0 1 3 6

Задача G. $R2 \mid\mid C_{max}$

Имя входного файла:	`r2cmax.in`
Имя выходного файла:	`r2cmax.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Задача $R2 \mid\mid C_{max}$. Дано два разных неоднородных станка, которые работают параллельно. Есть n работ, время выполнения которых на первом и втором станке различное. Нужно минимизировать время завершения всех работ.

Формат входного файла

В первой строке дано целое число n ($1 \le n \le 1000$) — количество работ. Во второй строке дано n чисел $p_{i,1}$ ($0 \le p_{i,1} \le 100$) — времена выполнения i -ой работы на первом станке. В третьей строке дано n чисел $p_{i,2}$ ($0 \le p_{i,2} \le 100$) — времена выполнения i -ой работы на втором станке.

Формат выходного файла

Вывести единственное целое число — ответ на задачу, то есть минимальное время, за которое выполнятся все работы.

Примеры

r2cmax.in	r2cmax.out
3 1 2 3 4 2 3	3

Задача H. $1 \mid prec; pmtn; r_i \mid f_{max}$

Имя входного файла:	`p1precpmtnrifmax.in`
Имя выходного файла:	`p1precpmtnrifmax.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Задача $1 \mid prec; pmtn; r_i \mid f_{max}$. Дан один станок и n деталей с временами выполнения $p_1, p_2, \ldots, p_n$. Также даны времена появления работ $r_1, r_2, \ldots, r_n$ (работу нельзя делать раньше, чем она становится доступной для выполнения) и граф зависимостей работ. Выполнение работы можно приостанавливать. Для каждой работы задана неубывающая функция $f_i = a_i x^2 + b_i x + c_i$. Требуется составить такое расписание, чтобы величина $f_{max} = \max(f_i(C_i))$, где C_i — время окончания выполнения работы i, была минимальна.

Формат входного файла

В первой строке дано целое число n ($1 \le n \le 1000$) — количество работ.

Во второй строке заданы целые числа p_i ($0 \le p_i \le 1000$) — времена, необходимые для выполнения деталей.

В третьей строке заданы целые числа r_i ($0 \le r_i \le 10^5$) — времена появления деталей.

В четвертой строке дано целое число m ($1 \le m \le 20000$) — количество ребер в графе зависимостей работ.

В следующих m строках заданы сами ребра — пары целых чисел u и v ($1 \le u, v \le n$) — пара означает, что перед выполнением работы v необходимо завершить работу u. Граф зависимостей является ациклическим.

В следующих n строках заданы целые числа a_i, b_i, c_i ($0 \le a_i, b_i, c_i \le 50$) — параметры функций f_i. Гарантируется, что в оптимальном расписании все значения $f_i(C_i)$ помещаются в знаковый 64-битный тип данных.

Формат выходного файла

В первой строке выведите единственное целое число — f_{max}.

В каждой из следующих n строк выведите k_i — количество частей, на которые разбита работа с номером i, а также k пар чисел $start_{ij}, end_{ij}$ — времена начала и конца выполнения каждой из k_i частей. Если существует несколько оптимальных расписаний, выведите любое.

Примеры

p1precpmtnrifmax.in	p1precpmtnrifmax.out
5 3 2 2 2 3 12 0 6 0 4 3 2 4 2 5 3 1 0 0 10 0 1 1 0 2 0 1 1 0 0 1 0	20 1 12 15 1 0 2 1 6 8 1 2 4 2 4 6 8 9

Задача I. Производство мебели

Имя входного файла:	`furniture.in`
Имя выходного файла:	`furniture.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 megabytes

Джон — директор мебельной фабрики. Его фабрика производит мебель в наборах деталей, из которых собственно мебель собирает уже сам заказчик. Это дает возможность производить различные компоненты одного и того же предмета в любом порядке. Однако, мебель, производимая фабрикой, достаточно дорогая, и поэтому должна быть очень качественной. Это накладывает некоторые ограничения на производственный процесс. Мебель производится из редкого и дорогостоящего дерева, поэтому необходимо точное совпадение рисунка и цвета деталей. Из-за этого ограничения пропадает возможность делать одновременно две детали для одного и того же предмета мебели одновременно, так как цвет и текстура каждой детали подбирается на основании цвета и текстуры уже сделанных деталей.

Недавно Джон получил n заказов на секретеры из красного дерева. Каждый секретер состоит из m различных деталей, каждая из которых производится отдельной группой специально обученных рабочих. Поскольку многие операции делаются вручную, на производство каждой детали уходит ровно один день. Как уже было описано выше, детали для каждого секретера можно производить в любом порядке, однако нет возможности в один и тот же день создавать две различных детали для одного и того же секретера. Также невозможно в один день создать две одинаковых детали для разных секретеров, поскольку на заводе всего m групп рабочих.

Для каждого секретера Джон знает, в какой день d_i его потребует заказчик. Если к концу этого дня все детали не будут готовы, Джону придется заплатить за этот секретер v долларов штрафа.

Помогите Джону таким образом составить расписание работ, чтобы уплаченный им штраф был минимален.

Формат входного файла

Первая строка входного файла содержит три целых числа n, m and v ($1 \le n \le 200$, $1 \le m \le 100$, $1 \le v \le 10^6$) — количество секретеров, количество деталей в каждом и размер штрафа соответственно. Вторая строка содержит n целых чисел d_i ($1 \le d_i \le 1000$).

Формат выходного файла

В первую строку входного файла выведите одно целое число — минимальный размер штрафа, который придется заплатить Джону. Следующие n строк должны содержать планы выполнения работ для каждого секретера. Каждая строка должна содержать m целых чисел, j-ое число в i-ой строке должно содержать номер дня, в который будет сделана j-ая деталь для i-го секретера. Номер дня должен не превышать 10^5.

Если оптимальных решений несколько, выведите любое.

Примеры

furniture.in	furniture.out
5 3 100 3 4 5 4 5	100 1 2 3 2 3 4 3 4 5 4 1 2 5 6 1

Примечание

В приведенном примере невозможно сделать все секретеры вовремя. Один придется сделать с опозданием, поэтому штраф составит 100 долларов.

Задача J. $P1 \mid p_i = 1 \mid \sum U_i$

Имя входного файла:	`p1p1sumu.in`
Имя выходного файла:	`p1p1sumu.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Задача $P1 \mid p_i = 1 \mid \sum U_i$. Дана одна машина и n работ. Для каждой работы известен её дедлайн d_i. Каждая работа выполняется за единицу времени. Нужно минимизировать количество невыполненных работ.

Формат входного файла

В первой строке дано семь целых чисел: n ($2 \le n \le 30\,000\,000$), d_1, d_2, A, B, C, D ($0 \le d_1, d_2, A, B, C \le 1\,000\,000\,000$, $1 \le D \le 1\,000\,000\,000$), где n — количество работ, d_1, d_2 — дедлайны первой и второй работ соответственно, A, B, C, D — коэффициенты для расчета дедлайнов последующих работ. Для $i > 2$ дедлайн рассчитывается по формуле $d_i = (A \cdot d_{i-2} + B \cdot d_{i-1} + C)\%D$.

Формат выходного файла

В первой строке выведите единственное целое число — максимальное количество выполненных работ.

Примеры

p1p1sumu.in	p1p1sumu.out
5 1 1 3 1 2 10	2

Задача K. $P2 \mid prec; p_i = 1 \mid L_{max}$

Имя входного файла:	`p2precp1lmax.in`
Имя выходного файла:	`p2precp1lmax.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Задача $P2 \mid prec; p_i = 1 \mid L_{max}$. Даны два одинаковых станка, на которых необходимо обработать n деталей. Каждую деталь можно обрабатывать на любом станке. Время, необходимое для обработки любой детали, одинаково и равно одному. Для каждой детали известны момент времени, до которого необходимо закончить работу над этой деталью — d_i, а также номера деталей, зависимых от нее. Необходимо минимизировать максимальное опоздание, где опоздание рассчитывается как разность между временем, когда была закончена обработка детали, и временем дедлайна для этой детали. Станки начинают работу в нулевой момент времени.

Формат входного файла

В первой строке дано целое число n ($1 \le n \le 1400$) — количество деталей. Следующая строка содержит n чисел d_i ($0 \le d_i \le 10^9$) — дедлайны времени обработки детали. Следующие n строк содержат матрицу смежности графа зависимостей. На i-ой строке в j-ой позиции стоит 1, если работа с номером j зависит от работы с номером i, и 0 в противном случае. Гарантируется, что граф зависимостей не содержит циклов.

Формат выходного файла

В первой строке выведите два целых числа: l — максимальное опоздание в оптимальном расписании и t — время работы станков. Во второй строке выведите t целых чисел a_i ($1 \le a_i \le n$), где a_i — номер детали, обрабатываемой на первом станке в i-ый момент времени, если станок простаивал в i-ый момент, $a_i = -1$. В третьей строке выведите t целых чисел b_i ($1 \le b_i \le n$) — номера деталей, обрабатываемых на втором станке в формате, аналогичном формату для первого станка.

Примеры

p2precp1lmax.in	p2precp1lmax.out
4 4 2 1 1 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0	1 2 1 2 3 4

Задача L. $P \mid intree, p_i = 1 \mid L_{max}$

Имя входного файла:	`pintreep1l.in`
Имя выходного файла:	`pintreep1l.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Задача $P \mid intree, p_i = 1 \mid L_{max}$. Дано m машин и n работ. Для каждой работы известен её дедлайн d_i. Каждая работа выполняется за единицу времени. Дано дерево зависимостей. Работа может быть выполнена только тогда, когда выполнятся все работы в поддереве. Величина $L_i = c_i - d_i$, где c_i — это время завершения выполнения i-ой работы, называется опозданием. Нужно минимизировать максимальное опоздание.

Формат входного файла

В первой строке дано два целых числа n, m ($1 \le n, m \le 100\,000$) — количество работ и машин. Во второй строке заданы целые числа d_i ($0 \le d_i \le 10^9$) — дедлайны работ. Следующие $n-1$ строк — описание ребер в формате x_i y_i ($1 \le x_i, y_i \le n$), где x_i — работа, которая должна быть выполнена до y_i.

Формат выходного файла

В первой строке выведите минимально возможное максимальное опоздание. Во второй строке выведите n целых чисел a_i, где a_i — момент времени, в который нужно начать выполнять i-ю работу.

Примеры

pintreep1l.in	pintreep1l.out
4 1 4 2 1 3 2 1 3 1 4 3	1 3 2 1 0

Задача M. $R \,||\, \sum C_i$

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

Задача $R \,||\, \sum c_i$. Имеется множество из n работ и m станков. Время выполнения i - той работы на j - том станке равно $p_{i,j}$. Каждую работу надо выполнить ровно один раз на одном из m станков без прерываний. Минимизируйте $\sum c_i$ — сумму времен окончания работ.

Формат входного файла

В первой строке даны два целых числа: n ($1 \le n \le 40$) — количество работ и m ($1 \le m \le 40$) — количество станков. В следующих n строчках дано по m чисел в каждой, где в i-той строке j-тое число равно $p_{i,j}$ ($0 \le p_{i,j} \le 10^6$) — время выполнения i - той работы на j - том станке.

Формат выходного файла

В первой строке выведете единственное число — $\sum C_i$. В k - той строчке ($2 \le k \le m+1$) выведете сначало количество работ, которые будут сделаны на $k-1$ - станке, а затем выведете сами работы в том порядке, в котором они будут выполнятся на этом же станке.

Примеры

rsumc.in	rsumc.out
2 2 2 100 1 100	4 2 2 1 0
2 2 2 3 100 200	103 1 2 1 1

Задача N. $F2 \mid\mid C_{max}$

Имя входного файла:	`f2cmax.in`
Имя выходного файла:	`f2cmax.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Задача $F2 \mid\mid C_{max}$. Имеется множество из n работ и два станка. Время выполнения i-ой работы на j-ом станке равно $p_{i,j}$. Каждую работу надо выполнить сначала на первом станке, потом на втором.

Минимизируйте C_{max} — время выполнения последней работы на втором станке.

Формат входного файла

В первой строке дано одно целое число n ($1 \le n \le 100000$) — количество работ. В следующих двух строчках дано по n чисел в каждой, где в j-ой строке i-ое число равно $p_{i,j}$ ($0 \le p_{i,j} \le 10^6$) — время выполнения i-ой работы на j-ом станке.

Формат выходного файла

В первой строке выведите единственное число — C_{max}. В k - той строчке ($2 \le k \le 3$) выведите работы в том порядке, в котором они будут выполняться на $k-1$ станке.

Примеры

f2cmax.in	f2cmax.out
3 1 2 3 5 5 5	16 1 3 2 1 2 3
2 3 2 1 3	6 2 1 2 1

Задача O. $Q \mid\mid \sum C_i$

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

Задача $Q \mid\mid \sum C_i$. Дано n работ и m однородных станков. Для каждого станка j задана его характеристика t_j — время на выполнение единицы работы. Каждую работу нужно обработать на одном из станков, количество работы, которое нужно выполнить равно p_i, соответственно, если работа i выполняется на j-м станке, время ее выполнения будет равно $t_j p_i$. Необходимо минимизировать суммарное время окончания обработки.

Формат входного файла

В первой строке входного файла заданы целые числа n ($1 \le n \le 50000$) — количество работ и m ($1 \le m \le 10000$) — количество станков. Во второй строке задано n целых чисел p_i ($1 \le p_i \le 10^4$) — количества работы. В третьей строке задано m целых чисел t_j ($1 \le t_j \le 10^4$) — характеристика j-го станка.

Формат выходного файла

В первой строке выходного файлы выведите оптимальное значение целевой функции. В следующих n строках через пробел выведите пары чисел m_i, s_i — станок, на который назначена соответствующая работа и время начала ее обработки.

Примеры

qsumci.in	qsumci.out
4 1 5 2 3 1 2	42 1 12 1 2 1 6 1 0
6 2 2 2 2 2 2 2 1 2	32 2 0 1 0 1 2 2 4 1 4 1 6
7 3 1 1 4 13 3 2 8 2 4 1	62 3 0 2 0 3 4 3 8 3 1 1 0 1 4