

Содержание

Задача А. Гвоздики [0.5 sec, 256 mb]	2
Задача В. Longpath. Длиннейший путь [0.5 sec, 256 mb]	3
Задача С. Joseph Problem [0.5 sec, 256 mb]	4
Задача D. НПП [0.5 sec, 256 mb]	5
Задача Е. Палиндром [0.5 sec, 256 mb]	6
Задача F. Свертка [0.5 sec, 256 mb]	7
Задача G. Сплоченная команда [0.5 sec, 256 mb]	8
Задача H. Наилучшее приближение [2 sec, 16 mb]	9
Задача I. Ship [3 sec, 256 mb]	10
Задача J. Крышки [1 sec, 256 mb]	11
Задача K. Мостостроение [0.5 sec, 256 mb]	12
Задача L. Мостостроение [1 sec, 256 mb]	13
Задача M. Casino. Казино [1 sec, 256 mb]	14
Задача N. Неприводимые многочлены [0.5 sec, 256 mb]	15

Задача А. Гвоздики [0.5 сек, 256 mb]

На прямой доске вбиты гвоздики. Любые два гвоздика можно соединить ниточкой. Требуется соединить какие-то пары гвоздиков ниточками так, чтобы к каждому гвоздику была привязана хотя бы одна ниточка, а суммарная длина всех ниточек была минимальна.

Формат входных данных

В первой строке входного файла записано число N — количество гвоздиков ($2 \leq N \leq 100$). В следующей строке записано N чисел — координаты всех гвоздиков (неотрицательные целые числа, не превосходящие 10 000).

Формат выходных данных

В выходной файл нужно вывести единственное число — минимальную суммарную длину всех ниточек.

Пример

stdin	stdout
5 4 10 0 12 2	6

Задача В. Longpath. Длиннейший путь [0.5 сек, 256 mb]

Дан ориентированный граф без циклов. Требуется найти в нем длиннейший путь.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и дуг графа соответственно. Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается двумя натуральными числами b_i и e_i — началом и концом дуги соответственно ($1 \leq b_i, e_i \leq n$).

Входной граф не содержит циклов и петель.

$n \leq 10\,000$, $m \leq 100\,000$.

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — количество дуг в длиннейшем пути.

Пример

stdin	stdout
5 5 1 2 2 3 3 4 3 5 1 5	3

Задача C. Joseph Problem [0.5 sec, 256 mb]

N мальчиков стоят по кругу. Они начинают считать себя по часовой стрелке, счет ведется с единицы. Как только количество посчитанных достигает p , последний посчитанный (p -й) мальчик покидает круг, а процесс счета начинается со следующего за ним мальчика и вновь ведется с единицы.

Последний оставшийся в кругу выигрывает.

Можете ли вы посчитать, номер выигравшего мальчика в исходном кругу? (мальчики нумеруются числами от 1 до N по часовой стрелке, начиная с того самого мальчика, с которого начинался счет).

Формат входных данных

Во входном файле два целых числа — N и P ($1 \leq N, P \leq 10^6$).

Формат выходных данных

Выведите номер выигравшего мальчика.

Пример

stdin	stdout
3 4	2

Задача D. НПП [0.5 sec, 256 mb]

Для заданной числовой последовательности a_1, a_2, \dots, a_n требуется найти длину максимальной последовательнократной подпоследовательности.

Для последовательнократной подпоследовательности $a_{k_1}, a_{k_2}, \dots, a_{k_t}$ ($k_1 < k_2 < \dots < k_t$) верно, что $a_{k_i} | a_{k_j}$ при $1 \leq i < j \leq t$ (утверждение « $a|b$ » эквивалентно « b кратно a »). Подпоследовательность из одного элемента полагается последовательнократной по определению.

Формат входных данных

В первой строке входного файла записано одно натуральное число N ($1 \leq N \leq 1000$) — количество чисел в исходной последовательности. Далее следует N натуральных чисел, не превосходящих $2 \cdot 10^9$ — сама последовательность.

Формат выходных данных

Вывести единственное число, равное искомому количеству.

Примеры

stdin	stdout
4 3 6 5 12	3

Задача Е. Палиндром [0.5 сек, 256 mb]

Палиндромом называется строка, которая читается одинаково как слева направо, так и справа налево. Требуется найти самый длинный палиндром P , получающийся из данной строки S удалением любого (возможно, нулевого) количества символов.

Формат входных данных

Входной файл содержит строчку S , состоящую из строчных латинских букв (a–z). Длина S не превышает 1 000 символов.

Формат выходных данных

Выходной файл должен содержать искомый палиндром. Если таких палиндромов несколько, выведите любой из них.

Примеры

stdin	stdout
anna	anna
perevorot	ror

Задача F. Свертка [0.5 sec, 256 mb]

Петя хочет сократить запись последовательности, состоящей из заглавных латинских букв. Для этого он может свернуть ее повторяющиеся подпоследовательности. Например, последовательность `AAAAAAAAABABABCCD` может быть записана как `10(A)2(BA)B2(C)D`.

Формальное определение свернутой последовательности и соответствующей ей операции развертки дается следующим образом:

- Последовательность, которая содержит единственный символ от 'A' до 'Z' представляет из себя свернутую последовательность. При развертке такой последовательности получается она сама.
- Если S и Q — свернутые последовательности, то SQ также свернутая последовательность. Если при развертке строки S получается строка S' , а при развертке Q получается Q' , то при развертке SQ получается строка $S'Q'$.
- Если S — свернутая последовательность, то $X(S)$ также свернутая последовательность, где X это десятичное представление целого числа большего единицы. Если при развертке строки S получается строка S' , то при развертке $X(S)$ получается строка S' , повторенная X раз.

Петя хочет свернуть заданную последовательность таким образом, чтобы результат содержал наименьшее число символов.

Формат входных данных

Входной файл содержит непустую строку, состоящую из заглавных латинских букв. Длина строки не превышает 100 символов.

Формат выходных данных

В выходной файл выведите одну строку, содержащую наименьшую последовательность развертка которой даст строку, заданную во входном файле.

Если ответов несколько — выведите любой из них.

Пример

stdin	stdout
AAAAAAAAABABABCCD	9(A)3(AB)CCD
NEERCYESYESYESNEERCYESYESYES	2(NEERC3(YES))

Задача G. Сплоченная команда [0.5 сек, 256 mb]

Скоро в ЛКШ планируется провести игру «Форт Баярд». В ней могут принимать участие команды с числом игроков от L до R , причем, если один ЛКШонок играет в некоторой команде, то в этой же команде обязательно должны играть все его друзья. Будем называть количество пар друзей в составе команды ее *сплоченностью*. Помогите школьникам собрать как можно более сплоченную команду.

Формат входных данных

В первой строке входного файла через пробел записаны четыре натуральных числа: N , M , L и R ($1 \leq N \leq 2000$, $0 \leq M \leq 10^5$, $1 \leq L \leq R \leq N$), где N — количество школьников, готовых принять участие в игре, а M — количество пар друзей. В каждой из следующих M строк записана пара чисел a_i, b_i ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$), означающая, что ЛКШата с номерами a_i и b_i являются друзьями.

Формат выходных данных

В первой строке выведите количество человек в искомой команде. Во второй строке через пробел запишите номера всех игроков, входящих в ее состав. Если команду собрать нельзя, в единственной строке выходного файла выведите -1.

Примеры

stdin	stdout
12 13 7 9 1 3 1 2 2 3 3 4 5 6 6 8 6 7 5 7 5 8 7 8 11 9 9 12 10 9	8 1 2 3 4 5 6 7 8
7 6 5 6 1 4 4 6 3 7 2 3 2 7 5 7	-1

Задача Н. Наилучшее приближение [2 sec, 16 mb]

Вам даны N целых чисел. Ваша задача — вставить ровно по одному знаку “+” или “-” между каждой парой соседних таким образом, чтобы сделать значение получившегося выражения максимально близким к заданному числу A .

Формат входных данных

Первая строка входного файла содержит два целых числа: N ($1 \leq N \leq 10\,000$) и A , которое по модулю не превосходит 10 000. Далее следуют N строк, в каждой из которых содержится ровно одно целое число X_i , не превосходящее по модулю 10 000. Кроме того, гарантируется, что сумма абсолютных величин всех N чисел также не превосходит 10 000.

Формат выходных данных

В первой строке необходимо вывести значение получившегося выражения (которое должно быть настолько близко к A , насколько это возможно). Во второй строке необходимо вывести само выражение, дающее такое значение, в форме $X_1[+|-]X_2[+|-]\dots X_{N-1}[+|-]X_N$. Если оптимальных решений несколько, то разрешается выводить любое из них.

Пример

stdin	stdout
3 0	0
3	3+-2-1
-2	
1	

Задача I. Ship [3 sec, 256 mb]

A ship has been assigned to transport cargo across the river Daugava (Western Dvina) that has the capacity to carry at most t tons of cargo. There is a hangar on the right bank of the river that contains n cargo containers to be transported by the ship. The weight of each container is an integer number of tons and it does not exceed t tons. The containers are arranged in a single line from the start to the end of the hangar. A container can be carried onto the ship by a liftloader that can pick only a container either at the start or at the end of the line. The total weight of the containers placed on a ship cannot exceed the capacity of the ship. None of the containers are allowed to be opened, thus it is impossible to redistribute the cargo among the containers. It is also expensive to move the containers so once the container has been picked up by the liftloader, it is then directly carried onto the ship. Only then the liftloader can pick another container from the hangar. Once the ship is loaded, it can transport the containers to the left bank and return to the right bank empty.

Write a program that calculates the minimum possible number of trips the ship has to make to transport all of the containers to the left river bank!

Формат входных данных

The first line of input contains two integers t and n ($1 \leq t \leq 10^9$, $1 \leq n \leq 10^4$), the capacity of the ship and the number of containers in the hangar, accordingly. The next line contains n integers, the weight of the containers in tons. The i -th number w_i ($1 \leq w_i \leq t$) denotes the weight of the i -th container from the start of the hangar. On each line the numbers are separated by spaces.

Формат выходных данных

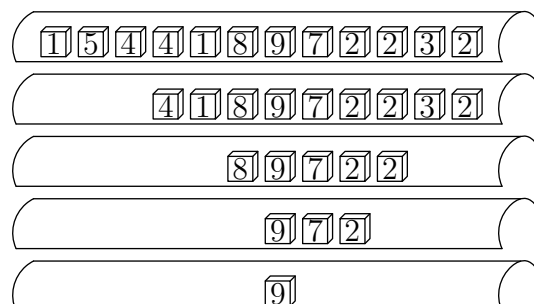
Output a single integer: the minimum possible trip number for the ship to transport all of the containers to the other river bank.

Примеры

stdin	stdout
10 12 1 5 4 4 1 8 9 7 2 2 3 2	5

Замечание

The following figure shows the remaining containers in the hangar before and after each trip in the given example.



Задача J. Крышки [1 сек, 256 mb]

У программиста Петрова есть хобби — собирать крышки от пивных бутылок. Ничего странного, он знает еще с сотню программистов, которые очень уважают пиво. Да, они тоже собирают крышки. Не все из них, конечно, но некоторые. Если честно, то часть своих крышек он просто купил, уже без бутылок. Да, это не совсем спортивно, зато коллекция теперь более солидная. Одна вот беда — не хватает ему для полноты коллекции еще нескольких редких крышек. Он даже нашел в Интернете программистов, которые согласны продать их ему. Некоторые даже продают крышки сразу наборами, с большой скидкой. Почему продают, да еще со скидкой? А вы попробуйте объяснить жене, какая польза от пивных крышек. Она же не программист. Осталось выбрать оптимальные предложения. Если объяснить жене зачем надо хранить крышки еще возможно, то почему на них надо тратить деньги — точно не поймет. Поэтому надо купить как можно дешевле. Петров выписал на бумажку все варианты и задумался. Купить сразу все не получится, никаких денег не хватит. Тогда надо купить самые необходимые, но подешевле. Да уж, без программы тут не обойдешься...

Формат входных данных

В первой строке записано число N — количество недостающих Петрову крышек ($1 \leq N \leq 20$). Далее идет N строк — цена, за которую можно купить эту крышку, если покупать ее отдельно. В следующей строке стоит число M ($0 \leq M \leq 100$) — количество предложений по продаже наборов, содержащих нужные ему крышки. Далее идет M строк описывающих эти наборы. В каждой строке первое число — цена набора, второе — количество крышек в наборе, далее перечислены номера крышек (каждый номер от 1 до N), которые в этот набор входят. Номера в наборе не повторяются. Все цены — положительные числа, не превосходящие 1000. В последней строке перечислен минимальный набор крышек, который Петров намерен купить в любом случае.

Формат выходных данных

Выведите минимальную сумму, необходимую Петрову, чтобы купить все крышки из приведённого в последней строке набора.

Примеры

stdin	stdout
4 10 11 12 13 3 17 2 1 3 25 3 2 3 4 15 2 3 4 3 1 3 4	25

Задача К. Мостостроение [0.5 сек, 256 mb]

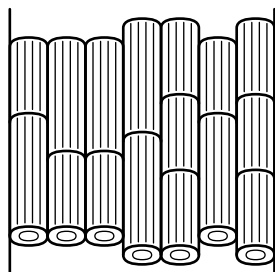
В деревне Гадюкино регулярно идут проливные дожди, в результате чего речка Вонючка, которую обычно можно просто перешагнуть, выходит из берегов. Чтобы можно было перейти разлившуюся реку, планируется построить плавучий мост из бревен, оставшихся от строительства бани бизнесмена, поселившегося неподалеку.

Все оставшиеся бревна имеют одинаковую толщину. При этом есть x бревен длины a и y бревен длины b .

Построенный мост должен состоять из l рядов, каждый из которых составлен из одного или нескольких бревен. Пилить бревна нельзя, так как последняя пила утонула при разливе Вонючки.

Главный инженер хочет построить мост максимальной возможной ширины, при этом ширина моста определяется по минимальной ширине ряда бревен.

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



Формат входных данных

Входной файл содержит пять натуральных чисел: x , a , y , b и l . Все числа не превышают 150. Общее количество бревен не меньше l .

Формат выходных данных

Выведите в выходной файл одно число — максимальную возможную ширину моста.

Примеры

stdin	stdout
6 3 10 2 7	5
10 7 20 9 25	9

Задача L. Мостостроение [1 сек, 256 mb]

Давным давно, в 2009-м году...

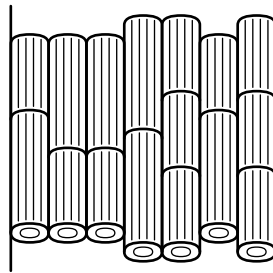
В деревне Зайкино регулярно идут проливные дожди, в результате чего речка Дубровка, которую обычно можно просто перешагнуть, выходит из берегов. Чтобы можно было перейти разлившуюся реку, планируется построить плавучий мост из брёвен, оставшихся от строительства бани бизнесмена, поселившегося неподалёку.

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

Для каждого тестового случая на отдельной строке выведите одно число — максимальную возможную ширину моста.

Пример

stdin	stdout
6 3 10 2 7	5
10 7 20 9 25	9
106 126 135 28 137	112

Задача М. Casino. Казино [1 sec, 256 mb]

Вновь открытое казино предложило оригинальную игру.

В начале игры крупье выставляет в ряд несколько фишек разных цветов. Кроме того, он объявляет, какие последовательности фишек игрок может забирать себе в процессе игры. Далее игрок забирает себе одну из заранее объявленных последовательностей фишек, расположенных подряд. После этого крупье сдвигает оставшиеся фишки, убирая разрыв. Затем игрок снова забирает себе одну из объявленных последовательностей и так далее. Игра продолжается до тех пор, пока игрок может забирать фишки.

Рассмотрим пример. Пусть на столе выставлен ряд фишек `'rrrgggbbb'`, и крупье объявил последовательности `'rg'` и `'gb'`. Игрок, например, может забрать фишки `'rg'`, лежащие на третьем и четвёртом местах слева. После этого крупье сдвинет фишки, и на столе получится ряд `'rrggbbb'`. Ещё дважды забрав фишки `'rg'`, игрок добьётся того, что на столе останутся фишки `'bbb'` и игра закончится, так как игроку больше нечего забрать со стола. Игрок мог бы действовать и по-другому — на втором и третьем ходах забрать не последовательности `'rg'`, а последовательности `'gb'`. Тогда на столе остались бы фишки `'rrb'`. Аналогично, игрок мог бы добиться того, чтобы в конце остались ряды `'rrr'` или `'rbb'`.

После окончания игры полученные фишки игрок меняет на деньги. Цена фишки зависит от её цвета. требуется написать программу, определяющую максимальную сумму, которую сможет получить игрок.

Формат входных данных

В первой строке входного файла записано число K ($1 \leq K \leq 26$) — количество цветов фишек. Каждая из следующих K строк начинается со строчной латинской буквы, обозначающей цвет. Далее в той же строке через пробел следует целое число X_i ($1 \leq X_i \leq 150$, $i = 1..K$) — цена фишки соответствующего цвета. В $(K + 2)$ -ой строке описан ряд фишек, лежащих на столе в начале игры. Ряд задается L строчными латинскими буквами ($1 \leq L \leq 150$), которые обозначают цвета фишек ряда. В следующей строке содержится число N ($1 \leq N \leq 150$) — количество последовательностей, которые были объявлены крупье. В следующих N строках записаны эти последовательности. Гарантируется, что сумма длин этих N строк не превосходит 150 символов, и все они непустые.

Формат выходных данных

В выходной файл выведите единственное целое число — максимальную сумму денег, которую может получить игрок.

Пример

stdin	stdout
6 a 1 b 4 d 2 x 3 f 1 e 3 fxeeabadd 2 aba ed	16

Задача N. Неприводимые многочлены [0.5 sec, 256 mb]

Найти количество неприводимых унитарных $f \in \mathbb{Z}_p[x] : \deg f = n$ ($p \in P$).

Напомним, \mathbb{Z}_p — это поле вычетов по модулю p . $\mathbb{Z}_p[x]$ — множество всех многочленов над полем \mathbb{Z}_p , то есть многочленов, коэффициенты которых являются остатками по модулю p .

Ненулевой многочлен называется *унитарным*, если его старший коэффициент равен 1.

Многочлен $f \in \mathbb{Z}_p[x]$ называется *неприводимым*, если не существует таких многочленов $g, h \in \mathbb{Z}_p[x] : \deg g > 0, \deg h > 0$ и $f(x) = g(x)h(x)$.

Формат входных данных

В единственной строке входного файла два числа: простое p и натуральное n .
 $1 \leq p, n \leq 10^9; 1 \leq p^n < 10^{18}$.

Формат выходных данных

Выведите единственное число — количество неприводимых унитарных многочленов степени n над полем остатков по модулю p .

Примеры

stdin	stdout
2 2	1
3 4	18

Замечание

В первом примере единственный такой многочлен — $x^2 + x + 1$.