

Задача 1. Расписание

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

Вася очень любит историю. И решил разобраться, как жили студенты в древние времена. Он выяснил, что расписание в то время было совсем другим!

Каждый день в университете проходил следующим образом. В 00:00 наступало утро, которое длилось до начала первой пары. Затем проходило 6 пар, разделенных переменами. Наконец, наступал вечер, который продолжался до 23:59 включительно. Зная время и древнее расписание звонков, определите, шла ли в это время пара, и если шла, то какая. Пара включает в себя минуту своего начала, но не включает минуту своего окончания.

Расписание звонков приведено в таблице:

Номер пары	Время начала	Время окончания
1	08:30	09:55
2	10:10	11:35
3	11:50	13:15
4	13:45	15:10
5	15:25	16:50
6	17:05	18:30

Формат входных данных

Время в формате `hh:mm` (с ведущими нулями). Гарантируется, что `hh` от 00 до 23, `mm` от 00 до 59.

Формат выходных данных

Строка «MORNING», если было утро; строка «EVENING», если был вечер; строка «BREAK», если была перемена; целое число (номер пары), если была пара.

Примеры

<i>standard input</i>	<i>standard output</i>
11:20	2

Задача 2. Подземелье

Имя входного файла:	<i>standard input</i>
Имя выходного файла:	<i>standard output</i>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 MiB

Герой находится в подземелье, полном монстров. Его изначальная сила равна 1. Подземелье представляет собой прямоугольное поле из $m \times n$ клеток. Каждая клетка может быть либо стеной, либо содержать монстра (крысу силой 1, гоблина силой 2, минотавра силой 3, слона силой 4, или дракона силой 5), либо содержать драгоценный камень, либо содержать героя, либо быть пустой. Ваша задача – за минимальное число ходов попасть на клетку с драгоценным камнем.

За один ход можно выполнить одно из двух действий:

- Перейти на соседнюю клетку, если она пуста или содержит драгоценный камень.
- Победить монстра на соседней клетке, если сила героя не меньше, чем сила монстра, и перейти на его клетку (монстр исчезает). Если при этом сила побежденного монстра была равна силе героя, сила героя увеличивается на 1.

Соседними считаются клетки, находящиеся слева, справа, сверху и снизу от клетки героя. После любого перехода героя предыдущая занимаемая им клетка становится пустой.

Обозначения клеток подземелья приведены в таблице:

Символ	Значение
'.'	пустая клетка
'#'	стена
'*'	драгоценный камень
'@'	начальное положение героя
'r'	крыса
'g'	гоблин
'm'	минотавр
'E'	слон
'D'	дракон

Формат входных данных

В первой строке разделенные пробелом целые числа m и n ($1 \leq m, n \leq 100$) – размеры подземелья. Затем m строк по n символов – описание подземелья. Гарантируется, что строки содержат только символы из таблицы и ровно один символ ровно одной строки – это символ '@'.

Формат выходных данных

Строка «IMPOSSIBLE», если попасть на клетку с драгоценным камнем невозможно, иначе число t – минимальное количество ходов, за которой можно попасть на клетку с драгоценным камнем.

Примеры

<i>standard input</i>	<i>standard output</i>
2 3 @#* .r.	4

Задача 3. Подмножества

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

Пусть A – конечное множество натуральных чисел. Поставим в соответствие каждому его непустому подмножеству B число $f(B)$, равное произведению элементов этого подмножества. Требуется вычислить последние три разряда числа, равного сумме $\sum_{B \subset A, B \neq \emptyset} f(B)$.

Формат входных данных

В первой строке одно целое число n , $1 \leq n \leq 1\,000$ – мощность множества A . Во второй строке n натуральных чисел a_i , $1 \leq a_i \leq 10^9$ – элементы множества A .

Формат выходных данных

Три последние цифры числа $\sum_{B \subset A, B \neq \emptyset} f(B)$. Если количество цифр в этой сумме меньше трех, то вывести саму сумму.

Примеры

<i>standard input</i>	<i>standard output</i>
3 1 2 3	23

Задача 4. Скорее в укрытие

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

Наконец-то пороссятам удалось вырваться из волчьей норы. Теперь им надо как можно быстрее добраться до своего домика. Помогите пороссятам выбрать самый короткий путь.

В лесу имеется N полянок и M тропинок между ними. Каждая тропинка соединяет две различные полянки. Две полянки могут быть соединены не более чем одной тропинкой. Бегать в лесу можно только по тропинкам.

Пороссята выбегают с полянки, на которой находится волчья нора, и бегут до полянки, на которой находится их домик. Пороссята давно живут в лесу и знают, сколько времени у них займет пробежка по каждой из тропинок.

Также пороссята уверены, что духи леса на их стороне, и рассчитывают на их волшебную силу: один раз духи могут мгновенно перенести пороссят по любой из тропинок (не зависимо от того, сколько времени заняло бы это перемещение у пеших пороссят).

Формат входных данных

Первая строка входного файла содержит числа N , M ($2 \leq N \leq 2000, 1 \leq M \leq 4000$). Следующие M строк содержат по три числа: B_i , E_i – номера полянок, которые соединяет i -я тропинка, и T_i – время, за которое пороссята могут по ней пробежать ($1 \leq T_i \leq 1000$). В следующей (последней) строке два числа: номер полянки, на которой находится волчье логово, и номер полянки, на которой стоит дом пороссят. Все числа во входном файле целые и в пределах одной строки разделены пробелами.

Формат выходных данных

Одно целое число – минимальное время, за которое пороссята доберутся до дома. Гарантируется, что решение задачи существует (нельзя же обрекать пороссят на вечные скитания по лесу).

Примеры

<i>standard input</i>	<i>standard output</i>
7 10 1 2 2 1 4 15 2 3 3 2 6 4 3 7 3 3 4 6 3 6 3 4 5 7 4 7 2 5 7 2 1 5	4

Задача 5. Перепутанные диски

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

Вася – страстный любитель старых компьютерных игр. Его коллекция насчитывает много дисков с играми. Однако он очень неаккуратный мальчик. Коробки с дисками в полном беспорядке раскиданы по его столу, и найти что-либо на столе практически невозможно.

Поэтому Вася действует следующим образом. Когда он хочет поиграть в очередную игру, он берет произвольную коробку с диском со стола и вставляет из нее диск. Если при этом в CD-приводе уже есть какой-нибудь диск, то вместо того, чтобы найти коробку от этого диска и убрать его туда, Вася убирает диск в коробку, из которой он только что достал очередной диск.

Например, пусть у Васи три игры – «Цивилизация», «Герои» и «Сапер». Если Вася сначала играл в «Цивилизацию», а затем в «Героев», то диск с «Цивилизацией» окажется в коробке от «Героев». Если затем Вася играл в «Сапера», то диск от «Героев» окажется в коробке от «Сапера». Если после этого Вася снова играл в «Цивилизацию» (которую ему пришлось достать из коробки от «Героев»), то «Сапер» окажется в коробке от «Героев», а «Цивилизация» – в CD-приводе Васиного компьютера.

Предполагая, что вначале все диски с играми находятся в своих коробках, напишите программу, которая по заданной последовательности игр, в которые играл Вася, определит, в какой коробке в результате оказался каждый из дисков с играми.

Формат входных данных

Первая строка входного файла содержит число N – количество игр, в которые играл Вася ($1 \leq N \leq 1000$). Следующие N строк содержат названия игр в том порядке, в котором в них играл Вася. Все названия состоят из латинских букв, цифр и пробелов, длина названия не превышает 50 символов. Вася мог играть в одну и ту же игру несколько раз.

Формат выходных данных

В выходной файл выводится K строк, где K – количество различных игр, в которые играл Вася. Каждая строка должна иметь вид « $A - B$ », где A – название игры, а B – название игры, в коробке от которой лежит игра A . Если соответствующая игра лежит в CD-приводе компьютера, вместо B выводится «*» (звездочка). Игры выводятся в лексикографическом порядке.

Примеры

<i>standard input</i>	<i>standard output</i>
4	Civilization - *
Civilization	HoMM - Minesweeper
HoMM	Minesweeper - HoMM
Minesweeper	
Civilization	

Задача 6. Дорожные работы

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

Зяц Геннадий очень любит морковь и экономить. На его ферме есть амбар, береза, сарай и дом. Геннадий решил сэкономить и нанял для прокладки дорожек крокодила Евгения. Но крокодил был не очень внимателен и поэтому проложил только прямые дорожки от амбара до березы и от сарая до дома. Геннадий не был сильно доволен таким исходом, и поэтому попросил Евгения сделать еще одну дорожку, соединяющую уже проложенные. Евгений, морщась, согласился, но только при том условии, что эта дорожка будет иметь наименьшую длину. Помогите Геннадию найти длину недостающего участка дорожки.

Формат входных данных

В первой строке целые числа A_x, A_y, B_x, B_y . A_x и A_y – x и y координаты амбара, B_x и B_y – x и y координаты березы. Во второй строке целые числа C_x, C_y, D_x, D_y . C_x и C_y – x и y координаты сарая, D_x и D_y – x и y координаты дома. Числа отделены друг от друга пробелами. Все координаты по модулю не превосходят 1000

Формат выходных данных

В единственной строке вещественное число – минимально возможная длина последнего участка дорожки. Если добавление новой дорожки не требуется – вывести 0. Абсолютная погрешность ответа не должна превышать 10^{-3} .

Примеры

<i>standard input</i>	<i>standard output</i>
-1 0 1 0 0 -1 0 1	0
-1 0 1 1 0 0 1 0	0.447214

Задача 7. Волшебные колеса

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 MiB

Один из изобретателей волшебной страны решил сделать самодвижущуюся карету! Для этого он изобрел волшебные колеса, которые крутятся сами по себе. Изготовив прототип и проведя испытания, он заметил, что карету сильно трясет. Что было неудивительно – ведь колеса имели форму выпуклого многоугольника. Чтобы улучшить потребительские качества своего изобретения, он решил минимизировать коэффициент тряски, который он определил как разность между максимальной и минимальной высотой, на которой карета находится от дороги в процессе движения. Чтобы выбрать оптимальный вариант, он хочет, чтобы можно было вычислять коэффициент тряски, не переделывая прототип. Напишите программу, которая по заданной форме колеса и положению оси вычислит коэффициент тряски (предполагая, что все колеса у кареты имеют одинаковую форму, стартуют в одинаковом положении и движутся синхронно).

Формат входных данных

В первой строке одно целое число N – количество точек в многоугольнике, определяющем форму колеса, $3 \leq N \leq 10$. В следующих N строках по два целых числа X_i и Y_i – координаты i -й точки многоугольника. Координаты перечисляются в обходе против часовой стрелки. В последней строке два целых числа X_c и Y_c – координаты точки, в которой колесо крепится к оси. Гарантируется, что эта точка находится внутри колеса. Все координаты не превышают по модулю 1 000.

Формат выходных данных

В первой и единственной строке одно вещественное число – коэффициент тряски для заданной модели колеса, с погрешностью, не превышающей 10^{-3} .

Примеры

тест	ответ
4 0 0 2 0 2 2 0 2 1 1	0.41421

Задача 8. Сильный шифр

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

Петя хочет работать в ФСБ. В свободное время он увлекается созданием собственных шифров. Однажды Петя придумал шифр. В результате шифрования он может преобразовать слово в зашифрованное и состоящее только из уникальных символов. Петя показал свой шифр Илье и он предложил усилить его следующим образом: Слово модифицируется добавлением дополнительных символов, но при этом количество уникальных символов не увеличивается. Также исходное слово является наименьшей лексикографической подпоследовательностью получившегося слова. Подпоследовательность не обязана быть непрерывной. Помогите Пете получить расшифрованное слово.

Формат входных данных

В первой строке одно целое число n , $1 \leq n \leq 100\,000$ – длина строки.

Во второй строке строка длины n . Строка состоит из строчных символов английского алфавита.

Формат выходных данных

В единственной строке – расшифрованное слово Пети.

Примеры

<i>standard input</i>	<i>standard output</i>
5 bcabc	abc

Задача 9. Капуста

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

Заяц Геннадий решил посадить капусту. Он знает, что грядки с номерами меньше чем N и больше чем M заняты морковкой. На каждой чётной грядке в этом году сажать нельзя. Номер грядки соответствует количеству необходимых семян. Помогите Геннадию найти сколько ему нужно купить семян капусты.

Формат входных данных

В единственной строке два целых числа N и M , $1 \leq N \leq M < 10^9$, разделенных пробелом.

Формат выходных данных

В единственной строке одно целое число – количество семян капусты.

Примеры

<i>standard input</i>	<i>standard output</i>
4 9	21

Задача 10. Волшебная репка

Имя входного файла: *standard input*
Имя выходного файла: *standard output*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 MiB

Заяц Геннадий решил посадить репку. Да не простую, а волшебную. Правда, как оказалось, все ее волшебство заключается в том, что если посадить три репки подряд, то они исчезают. Это очень расстроило Геннадия, но семена он уже купил. Поэтому он решил определить, сколькими способами можно высадить на грядку репку и морковь так, чтобы ни одна репка не исчезла.

Поскольку заяц Геннадий очень любит порядок, грядка должна состоять ровно из N овощей (репки или морковки), высаженных подряд.

Формат входных данных

В первой и единственной строке одно целое число N , $1 \leq N \leq 10^7$ – количество овощей на грядке.

Формат выходных данных

В единственной строке одно целое положительное число по модулю $10^9 + 7$ – количество способов посадить грядку так, чтобы ни одна репка не исчезла.

Примеры

<i>standard input</i>	<i>standard output</i>
5	24