

Задача А. Выбор заявки

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

Дана лекционная аудитория, в которой несколько профессоров хотят прочесть свои лекции. Для составления расписания профессора подали заявки, вида $[s_i; f_i)$ – время начала и конца лекции. Лекция считается открытым полуинтервалом, то есть какая-то лекция может начаться в момент окончания другой, без перерыва. Составьте расписание занятий так, чтобы выполнить максимальное количество заявок.

Формат входных данных

В первой строке вводится натуральное число n , не более 1000 – общее количество заявок. Затем вводится n строк с описаниями заявок - по два числа в каждом s_i и f_i .

Гарантируется, что $s_i < f_i$. Время начала и окончания лекции – натуральное число, не превышает 1440 (в минутах с начала суток).

Формат выходных данных

Выведите одно число – максимальное количество заявок, которые можно выполнить.

Примеры

стандартный ввод	стандартный вывод
1 5 10	1
3 1 5 2 3 3 4	2

Замечание

Во втором примере можно выполнить вторую и третью заявки.

Задача В. Проблема сапожника

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

В некоей воинской части есть сапожник. Рабочий день сапожника длится n минут. Заведующий складом оценивает работу сапожника по количеству починенной обуви, независимо от того, насколько сложный ремонт требовался в каждом случае. Дано k сапог, нуждающихся в починке. Определите, какое максимальное количество из них сапожник сможет починить за один рабочий день.

Формат входных данных

В первой строке вводятся число n (натуральное, не превышает 1000), и число k (натуральное, не превышает 500). Затем идет k чисел — количество минут, которые требуются чтобы починить i -й сапог (времена — натуральные числа, не превосходят 100).

Формат выходных данных

Выведите единственное число — максимальное количество сапог, которые можно починить за один рабочий день.

Примеры

стандартный ввод	стандартный вывод
10 3 6 2 8	2
3 2 10 20	0

Задача С. Большая политика

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

В некотором королевстве есть n провинций. Король пожелал объединить все их под своей самодержавной властью. Естественно, чтобы никто не догадался об этих планах, он будет это делать поэтапно, а именно: раз в год он будет объединять какие-то две провинции в одну. Чтобы жителям обеих провинций не было обидно, новому территориальному образованию будет присвоено новое название, которое будет отличаться от обоих старых названий. Естественно, это потребует выпуска новых паспортов для жителей обеих провинций.

Очевидно, что если в первой провинции p_i жителей, а во второй — p_j жителей, то для них надо выпустить $p_i + p_j$ новых паспортов.

На следующий год король объединяет еще какие-то две провинции. И так далее, до тех пор пока вся территория королевства не будет объединена в одну большую провинцию. Определите, какое наименьшее количество новых паспортов придется выпустить, если король будет объединять провинции оптимально с этой точки зрения.

Формат входных данных

В первой строке вводится число n (натуральное, не превышает 10^5) — количество провинций. Затем вводится n чисел — количество жителей каждой провинции (натуральное, не превосходит 10^9). Гарантируется, что изначально в королевстве хотя бы две провинции.

Формат выходных данных

Выведите единственное число — количество новых паспортов, которые придется выпустить.

Примеры

стандартный ввод	стандартный вывод
2 2 6	8
3 6 2 4	18

Задача D. Круглый стол

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

Возрождая древние традиции английских рыцарей, в одном городе члены школьного клуба любителей информатики каждую неделю собираются за круглым столом и обсуждают результаты последних соревнований.

Руководитель клуба Иван Петрович недавно заметил, что не все ребята активно участвуют в обсуждении. Понаблюдав за несколькими заседаниями клуба, он заметил, что активность члена клуба зависит от того, кто с кем сидит рядом.

В клуб приходят на занятия m мальчиков и n девочек. Иван Петрович заметил, что мальчик активно участвует в обсуждении только тогда, когда непосредственно рядом с ним с обеих сторон от него сидят девочки, а девочка активно участвует в обсуждении только тогда, когда непосредственно рядом с ней с одной стороны от нее сидит мальчик, а с другой — девочка.

Желая сделать заседание клуба как можно более интересным, Иван Петрович решил разместить участников за круглым столом таким образом, чтобы как можно больше членов клуба приняло активное участие в обсуждении.

Требуется написать программу, которая по заданным числам m и n выведет такой способ размещения m мальчиков и n девочек за круглым столом, при котором максимальное количество членов клуба будет активно участвовать в обсуждении.

Формат входных данных

Входной файл содержит два целых числа m и n , разделенных ровно одним пробелом ($0 \leq m \leq 1000, 0 \leq n \leq 1000, m + n \geq 3$).

Формат выходных данных

Выходной файл должен содержать строку с расположенными в некотором порядке m символами «B» (заглавная латинская буква) и n символами «G» (заглавная латинская буква). Символ «B» означает мальчика, а символ «G» — девочку.

Символы следует расположить в том порядке, в котором нужно разместить членов клуба вокруг стола. Соседние символы соответствуют членам клуба, которые сидят рядом. Рядом сидят также члены клуба, соответствующие первому и последнему символу выведенной строки.

Примеры

стандартный ввод	стандартный вывод
1 2	GGB
2 2	GGBB

Задача Е. Путешествие

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

Андрей едет из пункта A в пункт B на автомобиле. Расстояние между этими пунктами равно n километров. Известно, что с полным баком автомобиль способен проехать k километров. Дана карта, на которой отмечены координаты бензоколонок, относительно пункта A . Определите минимальное число заправок, которые придется сделать Андрею, чтобы успешно достичь пункта B . Известно, что при выезде из пункта A бак был полон.

Формат входных данных

В первой строке вводятся числа n и k (натуральные, не превосходят 1000). В следующей строке вводится количество бензоколонок s , потом следует s натуральных чисел, не превосходящих n — расстояния от пункта A до каждой заправки. Заправки упорядочены по удаленности от пункта A .

Формат выходных данных

Если при данных условиях пункта B достичь невозможно, то вывести число -1 . Если решение существует, то вывести минимальное количество остановок на дозаправку, которое нужно, чтобы достичь пункта B .

Примеры

стандартный ввод	стандартный вывод
100 20 1 50	-1
100 100 3 10 20 80	0
100 50 1 50	1

Задача F. Костюмы

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

Команда ЛКШ по плаванию состоит из n игроков, известна базовая скорость каждого игрока v_i . В шкафчике находится k магических плавательных костюмов, про которые тренер пустил слух, что они дают бонус к скорости. Костюмы бывают двух типов - спецназовские костюмы с шипами дают процентный бонус, а обычные плавки дают количественный бонус. Мощность воздействия костюма описывается целым числом от 1 до 300. Для спецназовских костюмов оно показывает, на сколько процентов увеличится базовая скорость, а для плавков - на какую величину.

Требуется раздать плавательные костюмы так, чтобы суммарная скорость команды была максимальной. Ясно, что каждый игрок получает не больше одного костюма, если ему не достается костюма, то он идет в шапочке.

Формат входных данных

В первой строке записано число n ($0 \leq n \leq 400$) - число спортсменов, далее N чисел, которые описывают их базовые скорости (целое число от 1 до 10^4). Далее записано число k ($0 \leq k \leq 800$) - количество костюмов, затем k пар целых чисел, описывающих соответствующую костюмы (тип и мощность). Тип пары описывается либо единичкой (спецназовские костюмы), либо двоечкой (плавки).

Формат выходных данных

Выведите максимальную суммарную скорость команды с точностью до 6 знаков после запятой.

Пример

стандартный ввод	стандартный вывод
7	82.9800000000
8 7 4 5 3 4 2	
9	
2 5	
1 8	
2 9	
2 4	
1 100	
2 13	
2 10	
1 11	
1 14	

Задача G. Интернет

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

Новый интернет-провайдер предоставляет услугу доступа в интернет с посекундной тарификацией. Для подключения нужно купить карточку, позволяющую пользоваться интернетом определенное количество секунд. При этом компания продает карточки стоимостью $1, 2, 4, \dots, 2^{30}$ рублей на $a_0, a_1, a_2, \dots, a_{30}$ секунд соответственно.

Родители разрешили Пете пользоваться интернетом m секунд. Определите, за какую наименьшую сумму он сможет купить карточки, которые позволят ему пользоваться интернетом не менее m секунд. Естественно, Петя может купить как карточки различного достоинства, так и несколько карточек одного достоинства.

Формат входных данных

В первой строке содержится единственное натуральное число m ($1 \leq m \leq 10^9$). Во второй строке задаются натуральные числа a_0, a_1, \dots, a_{30} , не превосходящие 10^9 .

Формат выходных данных

Выведите единственное число — наименьшую сумму денег, которую Пете придется потратить.

Пример

стандартный ввод	стандартный вывод
11 1 1 10 1	5

Задача Н. Раскраска кубиков

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

На день рождения Пете подарили коробку кубиков. На каждом кубике написано некоторое целое число. Петя выложил все n своих кубиков в ряд, так что числа на кубиках оказались расположены в некотором порядке $a[1], a[2], \dots, a[n]$. Теперь он хочет раскрасить кубики в разные цвета таким образом, чтобы для каждого цвета последовательность чисел на кубиках этого цвета была строго возрастающей. То есть, если кубики с номерами $i[1], i[2], \dots, i[k]$ покрашены в один цвет, то $a[i[1]] < a[i[2]] < \dots < a[i[k]]$. Петя хочет использовать как можно меньше цветов. Помогите ему!

Формат входных данных

Первая строка входного файла содержит число n — количество кубиков у Пети ($1 \leq n \leq 2 \cdot 5 \cdot 10^5$). Затем следует n чисел, разделенных пробелами и/или переводами строки — $a[1], a[2], \dots, a[n]$.

Формат выходных данных

На первой строке выходного файла выведите число m — наименьшее количество цветов, которое потребуется Пете. На следующей строке выведите n чисел из диапазона от 1 до m — цвета, в которые Петя должен покрасить кубики. Если ответов с минимальным m несколько, выведите любой.

Примеры

стандартный ввод	стандартный вывод
10 2 3 1 3 2 1 2 2 4 3	5 1 1 2 2 3 4 4 5 2 5
10 3 1 4 1 5 9 2 6 5 3	4 1 2 1 3 1 1 3 3 2 4

Задача I. Cookie clicker

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

В игре Cookie Clicker игрок зарабатывает печенье (cookies), щёлкая мышкой по изображению большой печеньки. Тратя заработанные печенье, игрок может покупать различные усовершенствования (ферму, фабрику и т. д.), которые также производят дополнительные печенье.

Рассмотрим упрощённый вариант этой игры. Пусть игрок может сделать один щелчок мышкой в секунду, что принесит ему одну печеньку. Также в любой момент времени игрок может потратить C печенек на покупку фабрики (при этом у игрока должно быть не меньше C печенек, после покупки фабрики количество его печенек моментально уменьшается на C). Каждая купленная фабрика увеличивает ежесекундное производство печенек на P штук (то есть если у игрока одна фабрика, то он получает $1 + P$ печенек в секунду, две фабрики — $1 + 2P$ печенек, три фабрики — $1 + 3P$ печенек и т. д.). Игрок может приобрести неограниченное число фабрик стоимостью C печенек каждая. Фабрика начинает производить дополнительные печенье сразу же, например, если после какой-то секунды игры у игрока стало C печенек, то игрок может купить фабрику и уже на следующей секунде его производство печенек увеличится на P штук.

Оригинальная игра никогда не заканчивается, но мы будем считать, что целью игры является набрать N печенек. Определите минимальное время, за которое может быть достигнута цель игры.

Формат входных данных

Программа получает на вход три целых положительных числа, записанных в отдельных строках: (стоимость фабрики), P (производительность одной фабрики) и N (необходимое количество печенек). Все числа не превосходят 10^9 .

Формат выходных данных

Программа должна вывести одно целое число — минимальное время в секундах, за которое игрок может получить не менее N печенек.

Примеры

стандартный ввод	стандартный вывод
50 3 100	75
99 10 100	100

Задача J. Города

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

Юный программист решил придумать собственную игру. Игра происходит на поле размером $n \times n$ клеток, в некоторых клетках которого расположены города (каждый город занимает одну клетку; в каждой клетке может располагаться не более одного города). Всего должно быть чётное количество городов.

Изначально про каждую клетку игрового поля известно, расположен ли в ней город или нет. Чтобы начать игру, необходимо разделить игровое поле на два государства так, чтобы в каждом государстве было поровну клеток-городов.

Граница между государствами должна проходить по границам клеток таким образом, чтобы из любой клетки каждого государства существовал путь по клеткам этого же государства в любую другую его клетку (из клетки можно перейти в соседнюю, если они имеют общую сторону). Каждая клетка игрового поля должна принадлежать только одному из двух государств, при этом государства не обязаны состоять из одинакового количества клеток.

Требуется написать программу, которая с учетом сказанного разделит клетки заданного игрового поля между двумя государствами.

Формат входных данных

Первая строка входного файла содержит одно целое положительное число n , задающее размер игрового поля ($1 \leq n \leq 50$).

Последующие n строк содержат по n заглавных латинских букв (без пробелов), кодирующих соответствующие клетки игрового поля: 'С' обозначает клетку, занятую городом, 'D' – пустую клетку. Гарантируется, что на поле есть хотя бы два города и всего их четное число.

Формат выходных данных

Выходной файл должен содержать n строк по n цифр (без пробелов) в каждой, кодирующих соответствующие клетки. Цифра 1 обозначает, что данная клетка принадлежит первому государству, цифра 2 – данная клетка принадлежит второму государству. Если решений несколько, необходимо вывести любое из них.

Пример

стандартный ввод	стандартный вывод
3	111
DDD	111
DDC	222
DDC	