

Задача А. Алёна, помни возраст Вити!

Входные данные:	стандартный ввод
Выходные данные:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Витя безумно любит статистику. Ещё бы — у них со старшим братом день рождения приходится на один и тот же день года! Теперь каждый год в свой день рождения он записывает, сколько лет ему и его брату, и пытается найти в этих записях что-нибудь интересное.

Сегодня у Вити день рождения, и он показал свои записи Алёне. Витя знает, что она тоже любит исследовать всякие наборы чисел и находить в них закономерности. Алёна тут же заметила интересный момент: когда в один из прошлых дней рождения Вите было n лет, его брату было m лет, а сегодня Витя младше своего брата ровно в k раз!

Вернувшись вечером домой, Алёна заинтересовалась вопросом: а достаточно ли этих данных, чтобы вычислить, сколько лет исполнилось Вите сегодня? Алёна быстро справилась, а сможете ли вы решить эту сложную задачу и выяснить по числам n , m и k , сколько лет Вите?

Формат входных данных

Ввод состоит из трех строк, которые содержат по одному натуральному числу: n , m и k — возраст Вити и его брата в былые времена, а также во сколько раз Витя сегодня младше своего брата ($1 \leq n < m \leq 10\,000$, $2 \leq k \leq 10\,000$).

Формат выходных данных

Если описанная в условии ситуация могла произойти, выведите единственное число t — сколько лет сегодня исполнилось Вите.

Если Витя и Алёна ошиблись, и описанной ситуации быть не могло, выведите число -1 .

Примеры

стандартный ввод	стандартный вывод
4 15 2	11
4 15 3	-1
4 8 2	-1

Задача В. Тройной Фибоначчи

Входные данные:	стандартный ввод
Выходные данные:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Пятиклассник Лёня недавно прочитал статью о числах Фибоначчи.

Числами Фибоначчи называется числовая последовательность $F_1, F_2, \dots, F_n, \dots$, которая устроена следующим образом: $F_1 = 1$, $F_2 = 2$, а каждое следующее число вычисляется как сумма двух предыдущих: если $i \geq 3$, то $F_i = F_{i-1} + F_{i-2}$. Последовательность чисел Фибоначчи, таким образом, начинается с чисел 1, 2, 3, 5, 8, 13, 21, ...

Сегодня Лёня изучает числа Фибоначчи с номерами от L до R , включительно. Так как Лёня очень любит число 3, ему стало интересно, сколько чисел Фибоначчи среди тех, которые он изучает сегодня, делятся на 3. Например, если $L = 3$ и $R = 7$, то Лёня будет изучать числа $F_3 = 3$, $F_4 = 5$, $F_5 = 8$, $F_6 = 13$ и $F_7 = 21$. Среди них на 3 делятся два числа: $F_3 = 3$ и $F_7 = 21$.

Напишите программу, которая поможет Лёне найти ответ на волнующий его вопрос.

Формат входных данных

Первая строка входных данных содержит число L , а вторая — число R ($1 \leq L \leq R \leq 10^5$).

Формат выходных данных

Выведите единственное число — количество чисел Фибоначчи с номерами от L до R , включительно, которые делятся на 3.

Пример

стандартный ввод	стандартный вывод
3 7	2

Задача С. Марсианские нолики

Входные данные:	стандартный ввод
Выходные данные:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

На Марсе используют систему счисления с основанием k . В отличие от привычной нам десятичной системы счисления, в этой системе счисления k цифр со значениями от 0 до $k - 1$, а вес цифры в i -м разряде равен k^i .

Например, пусть $k = 8$. Запись 357_8 означает число $3 \cdot 8^2 + 5 \cdot 8 + 7$, в более привычной землянам десятичной системе счисления это число записывается как 239_{10} . А число 192_{10} , в системе счисления с основанием 8 записывается как 300_8 .

Ильдар — юный марсианин, и он очень любит круглые числа. Ильдар называет число *достаточно круглым*, если его запись в системе счисления с основанием k заканчивается хотя бы на n нулей. Сегодня Ильдар хочет найти i -е по порядку достаточно круглое число.

Помогите Ильдару, найдите i -е достаточно круглое в системе счисления с основанием k натуральное число и выведите его в десятичной системе счисления. Ильдар очень дружелюбен и гарантирует, что ответ в десятичной системе счисления не превосходит 10^{18} .

Формат входных данных

Все ограничения на числа в этой задаче заданы в десятичной системе счисления. Все числа во вводе также записаны в десятичной системе счисления.

Первая строка входных данных содержит число k — основание системы счисления, которую использует Ильдар ($2 \leq k \leq 10^9$).

Вторая строка входных данных содержит число n — минимальное количество нулей на конце достаточно круглого числа ($0 \leq n \leq 100$).

Третья строка входных данных содержит число i — порядковый номер достаточно круглого числа, которое интересует Ильдара ($1 \leq i \leq 10^9$).

Формат выходных данных

Выведите одно число — запись в десятичной системе счисления i -го по порядку достаточно круглого в системе счисления с основанием k натурального числа. Гарантируется, что ответ не превышает 10^{18} .

Обратите внимание, что ответ может не поместиться в стандартный 32-битный тип данных. Надо использовать 64-битный тип, в паскале он называется «int64», в C++ «long long», в Java «long». Если вы пишете на языке Python, то волноваться не надо, в Python встроенный целочисленный тип не имеет ограничений на величину числа.

Пример

стандартный ввод	стандартный вывод
8	192
2	
3	

Задача D. Спорт или еда

Входные данные:	стандартный ввод
Выходные данные:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Арсений — молодой перспективный спортсмен. Всё, что любит делать Арсений — это тренироваться и вкусно есть. Также он отличается пунктуальностью. Только что он составил расписание из n пунктов: для каждого из следующих n часов он решил, что будет делать в это время — тренироваться или есть.

Арсений показал расписание своему тренеру, но ему оно не до конца понравилось. Тренер объяснил, что тренироваться в следующий час после приёма пищи вредно для здоровья.

Теперь Арсений хочет изменить свое расписание так, чтобы он ни разу не тренировался непосредственно после еды. При этом он хочет изменить минимальное число пунктов своего расписания. Помогите Арсению!

Формат входных данных

В первой строке входных данных содержится число n — количество пунктов в расписании Арсения ($1 \leq n \leq 10^5$).

Во второй строке содержится исходное расписание Арсения. Это строка s длины n , состоящая только из латинских букв «t» и «e», при этом если на позиции i в строке s стоит буква «t», то это значит, что в i -м часу Арсений запланировал тренироваться, а если на этой позиции стоит буква «e», то это значит, что в i -м часу Арсений запланировал есть.

Формат выходных данных

В первой строке выведите минимальное количество пунктов расписания, которые необходимо изменить.

Во второй строке выведите строку из n латинских букв «t» и «e» — изменённое расписание в том же формате, что и во входных данных. Если подходящих расписаний несколько, выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
6 tttete	1 ttteee
5 tttte	0 tttte
9 eeeeettt	4 eeeeeeee

Задача Е. Сборная Юпитера

Входные данные:	стандартный ввод
Выходные данные:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Каждый год в одном из уголков нашей вселенной проходят Интеллектуальные Олимпийские Игры. В этом году честь проводить это масштабное мероприятие выпала планете Юпитер. Вам предстоит отобрать две команды на Игры из имеющихся n кандидатов в сборную.

Все кандидаты являются школьниками, каждый кандидат учится в определенном классе. Как и на Земле, на Юпитере 11 классов, пронумерованных от 1 до 11. На Игры отбирается две команды по результатам отборочных соревнований. На соревнованиях проводится пять отборочных туров. По итогам каждого тура каждый школьник может набрать от 0 до 300 баллов, чем больше, тем лучше.

В первую команду попадают четыре лучших школьника по сумме баллов, набранных на всех отборочных турах. Во вторую команду попадают четверо лучших по сумме баллов из тех, кто не попал в первую команду, и при этом не учится в 11 классе.

Все туры уже проведены, и получилось так, что любые два кандидата набрали в сумме различное количество баллов. Осталось лишь написать программу, которая по имеющимся данным о кандидатах и результатах туров определит тех восьмерых, которые защитят честь Юпитера на Интеллектуальных Олимпийских Играх и докажут, что Юпитер — суперпланета!

Формат входных данных

Первая строка входных данных содержит единственное число n — количество кандидатов в сборную Юпитера ($8 \leq n \leq 500$).

Следующие n строк содержат информацию о кандидатах. Каждая строка содержит 6 целых чисел — номер класса, в котором учится очередной кандидат, и его результаты на отборочных турах.

Номер класса является числом от 1 до 11, а результат на каждом туре — числом от 0 до 300.

Гарантируется, что все участники имеют различные суммарные баллы.

Гарантируется, что есть хотя бы 8 кандидатов, обучающихся не в 11 классе.

Формат выходных данных

Выведите две строки.

Первая строка должна содержать четыре целых числа, разделенных пробелами — номера кандидатов, которые попадут в первую команду Юпитера. Кандидаты нумеруются с единицы в порядке их появления во вводе. Выводить номера следует в порядке возрастания.

Вторая строка должна описывать вторую команду Юпитера в аналогичном формате.

Пример

стандартный ввод	стандартный вывод
10	3 4 5 9
9 50 271 287 282 42	1 2 6 10
10 230 241 137 14 240	
10 276 109 300 197 300	
8 205 292 194 232 74	
10 294 291 299 300 255	
9 195 275 265 134 9	
11 204 259 96 263 83	
7 141 223 85 84 26	
11 286 294 289 221 261	
10 277 52 117 272 262	

Задача F. Маша и матрёшки

Входные данные:	стандартный ввод
Выходные данные:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Маше на день рождения подарили набор матрёшек!

Теперь Маша сидит и вкладывает их одна в другую. Она заметила, что матрёшки отличаются по размеру, и одна помещается внутри другой, только если ее размеры строго меньше. Так, если есть две матрёшки i и j , а их размеры a_i и a_j соответственно, то матрёшка i вкладывается внутрь матрёшки j тогда и только тогда, когда $a_i < a_j$. Разумеется, непосредственно внутрь матрёшки можно вложить только одну другую матрёшку, иначе получится неаккуратно, а Маша — очень аккуратная девочка.

Маше особенно нравится, если она может, вкладывая матрёшки друг в друга, добиться того, что все они оказываются внутри одной самой большой матрёшки. Но, к сожалению, это не всегда возможно. Поэтому Маша решила убрать часть матрёшек в шкаф, оставив такой набор, чтобы их все можно было вложить друг в друга. Помогите Маше понять, какое максимальное количество матрёшек может быть в таком наборе.

Формат входных данных

В первой строке находится число n — количество матрёшек, подаренных Маше ($1 \leq n \leq 1000$). В следующей строке через пробел находятся n чисел — размеры матрёшек. Число a_i , стоящее на месте i , задает размер матрёшки с номером i ($1 \leq a_i \leq 10\,000$).

Формат выходных данных

Выведите одно число — максимальное количество матрёшек, которые можно вложить друг в друга.

Примеры

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

Задача G. ASCII-графика

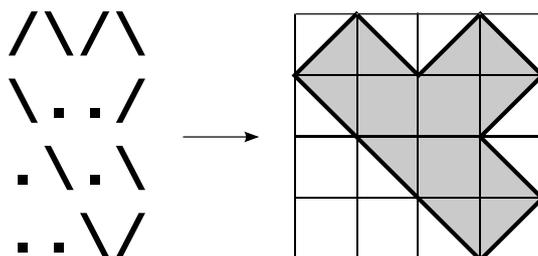
Входные данные: стандартный ввод
Выходные данные: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Давным-давно большинство персональных компьютеров были оборудованы видеокартами, работавшими только в текстовом режиме. Если программист хотел изобразить картинку на экране, ему приходилось использовать псевдографику или ASCII-графику. Вот пример картинка, нарисованный с ее помощью:

```
  ^  ^  
  .  .  
 (00)  
 /  \  
 ( ) ( )
```

Вам дан многоугольник, нарисованный с помощью ASCII-графики. Ваша задача состоит в том, чтобы посчитать количество его сторон.

Картинка состоит из символов '.', '\', '/' и '^'. Каждый символ изображает единичный квадрат картинка. Символ '.' обозначает пустой квадрат, символ '/' — квадрат с отрезком из левого нижнего угла в правый верхний, а символ '\' — квадрат с отрезком из левого верхнего угла в правый нижний.



Формат входных данных

Первая строка входных данных содержит два числа h и w — высота и ширина изображения ($2 \leq h, w \leq 100$). Следующие h строк, по w символов в каждой, содержат описание многоугольника, нарисованного с помощью ASCII-графики.

Гарантируется, что картинка содержит ровно один многоугольник, не имеющий самопересечений и самокасаний.

Формат выходных данных

Выведите одно число — количество сторон многоугольника.

Пример

стандартный ввод	стандартный вывод
4 4 /\/ \../ .\. ..\	8

Задача Н. Робот

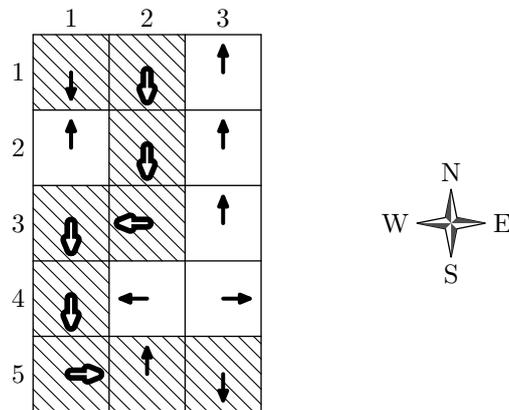
Входные данные:	стандартный ввод
Выходные данные:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

На Марсе есть большой полигон для испытания роботов. Он представляет собой таблицу из n строк и m столбцов, столбцы которой направлены с севера на юг, а строки — с запада на восток. В каждой клетке таблицы находится ускоритель, направленный в одну из сторон света.

Находясь в клетке, робот может воспользоваться ускорителем в ней. При этом он попадает в соседнюю с текущей клетку в направлении ускорителя и не тратит топлива. Если соседней клетки в направлении ускорителя нет, то им нельзя воспользоваться. Также робот может не пользоваться ускорителем и переместиться в любую из соседних клеток, затратив один литр топлива.

Сегодня роботу дали следующее задание: ему надо доехать из первой клетки первого столбца в последнюю клетку последнего столбца. Напишите программу, которая определит, какое минимальное количество топлива ему придется для этого потратить.

Рисунок ниже показывает полигон, заданный в примере. Ускорители показаны в виде стрелок. Заштрихованы клетки, по которым оптимально проехать роботу. Ускорители, которыми робот воспользовался, показаны жирными стрелками.



Формат входных данных

Первая строка входных данных содержит два числа n и m — протяженность полигона с севера на юг и с запада на восток ($1 \leq n, m \leq 20$).

Следующие n строк содержат по m латинских букв, описывающих направления ускорителей в очередной строке. Буква соответствуют направлению ускорителя: N — на север, W — на запад, S — на юг, E — на восток. Строки занумерованы с севера на юг, а столбцы с запада на восток. Таким образом, направление на север соответствует уменьшению номера строки, направление на юг — увеличению номера строки, направление на запад — уменьшению номера столбца, а направление на восток — увеличению номера столбца.

Формат выходных данных

Выведите одно целое число — минимальное количество литров топлива, которое должен потратить робот, чтобы выполнить задание.

Пример

стандартный ввод	стандартный вывод
5 3 SSN NSN SWN SWE ENS	2

Задача I. Нужно больше золота

Входные данные:	стандартный ввод
Выходные данные:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Петя очень любит компьютерные игры. Недавно он обнаружил в интернете интересную ролевую игру. Управляя героем, надо искать магические артефакты и получать золото.

К сожалению, первый же квест в игре поставил Петю в тупик. Выполнив задание, он получил n магических артефактов, которые можно использовать для получения золота. Для каждого артефакта известна его *ценность*, для i -го артефакта она равна w_i . Для получения золота артефакты можно *активировать*. Каждый артефакт можно активировать только один раз. Петя может активировать артефакты в произвольном порядке.

У героя, которым управляет Петя, есть *магическая сила*, исходно она равна нулю. Есть два способа активировать артефакт: с помощью магии и с помощью силы. Если активировать артефакт с ценностью w с помощью магии, то магическая сила героя увеличивается на w . Если же активировать артефакт с ценностью w с помощью силы, то герой получает xw золотых монет, где x — магическая сила героя в момент активации артефакта.

Например, если герой Пети получил 4 артефакта с ценностями 1, 1, 2 и 2, то можно получить 9 золотых монет, действуя следующим образом. Сначала надо активировать с помощью магии по одному артефакту с ценностью 1 и 2. После этого магическая сила героя равна 3, теперь можно активировать с помощью силы оставшиеся артефакты и получить 3 и 6 золотых монет, соответственно.

Помогите Пете определить, какое максимальное количество золотых монет он может получить, активировав артефакты в правильном порядке оптимальным образом.

Формат входных данных

Первая строка входных данных содержит единственное число n — количество магических артефактов ($1 \leq n \leq 100$).

Вторая строка входных данных содержит n чисел w_1, w_2, \dots, w_n — ценности артефактов ($1 \leq w_i \leq 100$).

Формат выходных данных

Выведите максимальное возможное число золотых монет, которые можно получить с помощью магических артефактов.

Пример

стандартный ввод	стандартный вывод
4 1 1 2 2	9

Задача J. Башни

Входные данные:	стандартный ввод
Выходные данные:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Егор очень любит Москву и часто ходит на прогулку, чтобы полюбоваться ее видами.

Москва в представлении Егора представляет собой длинное прямое шоссе, вдоль которого расположено n башен, пронумерованных в порядке их следования числами от 1 до n . Все башни имеют различные высоты, которые выражаются целыми числами от 1 до n . Высота башни с номером i составляет h_i .

Однако Егор не любит возрастающие последовательности. Его разочаровывают тройки башен, таких что с возрастанием их индексов их значения тоже возрастают. Более формально, тройка башен с номерами i , j и k разочаровывает Егора, если $i < j < k$ и $h_i < h_j < h_k$.

Егор заранее знает высоты всех башен в Москве и хочет узнать, сильно ли он разочаруется на прогулке. Помогите ему определить, сколько есть троек башен, которые его разочаруют.

Формат входных данных

Первая строка входных данных содержит единственное число n — количество башен в Москве ($1 \leq n \leq 8000$).

Вторая строка входных данных содержит n различных натуральных чисел, i -е из них h_i — высота i -й башни ($1 \leq h_i \leq n$).

Формат выходных данных

Выведите одно число — количество троек башен, которые разочаруют Егора.

Обратите внимание, что ответ может не поместиться в стандартный 32-битный тип данных. Надо использовать 64-битный тип, в паскале он называется «int64», в C++ «long long», в Java «long».

Примеры

стандартный ввод	стандартный вывод
5 1 3 4 2 5	5
3 3 1 2	0