

1. Гистограмма

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Вовочка ломает систему безопасности Пентагона. Для этого ему понадобилось узнать, какие символы в секретных зашифрованных посланиях употребляются чаще других. Для удобства изучения Вовочка хочет получить графическое представление встречаемости символов. Поэтому он хочет построить гистограмму количества символов в сообщении. Гистограмма — это график, в котором каждому символу, встречающемуся в сообщении хотя бы один раз, соответствует столбик, высота которого пропорциональна количеству этих символов в сообщении.

Формат ввода

Входной файл содержит зашифрованный текст сообщения. Он содержит строчные и прописные латинские буквы, цифры, знаки препинания («.», «!», «?», «:», «-», «,», «;», «(», «)»), пробелы и переводы строк. Размер входного файла не превышает 10000 байт. Текст содержит хотя бы один непробельный символ. Все строки входного файла не длиннее 200 символов.Для каждого символа с кроме пробелов и переводов строк выведите столбик из символов «#», количество которых должно быть равно количеству символов с в данном тексте. Под каждым столбиком напишите символ, соответствующий ему. Отформатируйте гистограмму так, чтобы нижние концы столбиков были на одной строке, первая строка и первый столбец были непустыми. Не отделяйте столбики друг от друга. Отсортируйте столбики в порядке увеличения кодов символов.

Формат вывода

Для каждого символа с кроме пробелов и переводов строк выведите столбик из символов «#», количество которых должно быть равно количеству символов с в данном тексте. Под каждым столбиком напишите символ, соответствующий ему. Отформатируйте гистограмму так, чтобы нижние концы столбиков были на одной строке, первая строка и первый столбец были непустыми. Не отделяйте столбики друг от друга. Отсортируйте столбики в порядке увеличения кодов символов.

Пример 1

Ввод

```
Hello, world!
```

Вывод

```
#
##
#####
!,Hde!orw
```

Пример 2

Ввод

```
Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.
```

Вывод

```
#
#
#
#
#
#      #
#  #      #
#  # ###  ####
## #####  ####
#####
#####  ##
#  # #####  ###
#####
,.;ADTabdeghilmnorstuvwy
```

2. Красивая строка

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Красотой строки назовем максимальное число идущих подряд одинаковых букв. (*красота* строки *abcaabdddettq* равна 3)
Сделайте данную вам строку как можно более *красивой*, если вы можете сделать не более *k* операций замены символа.

Формат ввода

В первой строке записано одно целое число k ($0 \leq k \leq 10^9$)
Во второй строке дана непустая строчка S ($|S| \leq 2 \cdot 10^5$). Строчка S состоит только из маленьких латинских букв.

Формат вывода

Выведите одно число — максимально возможную *красоту* строчки, которую можно получить.

Пример 1

Ввод 	Вывод 
2 abcaz	4

Пример 2

Ввод 	Вывод 
2 helto	3

3. Коллекционер Диего

Ограничение времени	2 секунды
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Диего увлекается коллекционированием наклеек. На каждой из них написано число, и каждый коллекционер мечтает собрать наклейки со всеми встречающимися числами. Диего собрал N наклеек, некоторые из которых, возможно, совпадают. Как-то раз к нему пришли K коллекционеров. i -й из них собрал все наклейки с номерами не меньшими, чем p_i . Напишите программу, которая поможет каждому из коллекционеров определить, сколько недостающих ему наклеек есть у Диего. Разумеется, гостей Диего не интересуют повторные экземпляры наклеек.



Формат ввода

В первой строке содержится единственное число N ($0 \leq N \leq 100\,000$) — количество наклеек у Диего. В следующей строке содержатся N целых неотрицательных чисел (не обязательно различных) — номера наклеек Диего. Все номера наклеек не превосходят 10^9 . В следующей строке содержится число K ($0 \leq K \leq 100\,000$) — количество коллекционеров, пришедших к Диего. В следующей строке содержатся K целых чисел p_i ($0 \leq p_i \leq 10^9$), где p_i — наименьший номер наклейки, не интересующий i -го коллекционера.

Формат вывода

Для каждого коллекционера в отдельной строке выведите количество **различных** чисел на наклейках, которые есть у Диего, но нет у этого коллекционера.

Пример 1

Ввод 	Вывод 
1	0
5	1
2	
4 6	

Пример 2

Ввод 	Вывод 
3	3
100 1 50	0
3	2
300 0 75	

4. Контрольная работа

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Петя и Вася — одноклассники и лучшие друзья, поэтому они во всём помогают друг другу. Завтра у них контрольная по математике, и учитель подготовил целых K вариантов заданий. В классе стоит один ряд парт, за каждой из них (кроме, возможно, последней) на контрольной будут сидеть ровно два ученика. Ученики знают, что варианты будут раздаваться строго по порядку: правый относительно учителя ученик первой парты получит вариант 1, левый — вариант 2, правый ученик второй парты получит вариант 3 (если число вариантов больше двух) и т.д. Так как K может быть меньше чем число учеников N , то после варианта K снова выдаётся вариант 1. На последней парте в случае нечётного числа учеников используется только место 1.

Петя самым первым вошёл в класс и сел на своё любимое место. Вася вошёл следом и хочет получить такой же вариант, что и Петя, при этом сидя к нему как можно ближе. То есть между ними должно оказаться как можно меньше парт, а при наличии двух таких мест с равным расстоянием от Пети Вася сядет позади Пети, а не перед ним. Напишите программу, которая подскажет Васе, какой ряд и какое место (справа или слева от учителя) ему следует выбрать. Если же один и тот же вариант Вася с Петей писать не смогут, то выдайте одно число - 1.

Формат ввода

В первой строке входных данных находится количество учеников в классе $2 \leq N \leq 10^9$. Во второй строке — количество подготовленных для контрольной вариантов заданий $2 \leq K \leq N$. В третьей строке — номер ряда, на который уже сел Петя, в четвёртой — цифра 1, если он сел на правое место, и 2, если на левое.

Формат вывода

Если Вася никак не сможет писать тот же вариант, что и Петя, то выведите - 1. Если решение существует, то выведите два числа — номер ряда, на который следует сесть Васе, и 1, если ему надо сесть на правое место, или 2, если на левое. Разрешается использовать только первые N мест в порядке раздачи вариантов.

Пример 1

Ввод 	Вывод 
25 2 1 2	2 2

Пример 2

Ввод 	Вывод 
25 13 7 1	−1

Примечания

В первом примере вариантов 2, поэтому наилучшее место для Васи находится сразу за Петей. Во втором примере Петя будет единственным, кто получит вариант 13.

5. Хорошая строка

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

На день рождения маленький Ипполит получил долгожданный подарок — набор дощечек с написанными на них буквами латинского алфавита. Теперь-то ему будет чем заняться долгими вечерами, тем более что мама обещала подарить ему в следующем году последовательность целых неотрицательных чисел, если он хорошо освоит этот набор. Ради такого богатства Ипполит готов на многое. Прямо сейчас юный исследователь полностью поглощён изучением хорошеести строк. Хорошестью строки называется количество позиций от 1 до L - 1 (где L — длина строки), таких, что следующая буква в строке является следующей по алфавиту. Например, хорошесть строки "abcdefghijklmnopqrstuvwxyz" равна 25, а строки "abdc" — только 1.

Ипполит размышляет над решением закономерно возникающей задачи: чему равна максимально возможная хорошесть строки, которую можно собрать, используя дощечки из данного набора? Вы-то и поможете ему с ней справиться.

Формат ввода

Первая строка ввода содержит единственное целое число N — количество различных букв в наборе ($1 \leq N \leq 26$). Обратите внимание: в наборе всегда используются N первых букв латинского алфавита. Следующие N строк содержат целые положительные числа c_i — количество букв соответствующего типа ($1 \leq c_i \leq 10^9$). Таким образом, первое число означает количество букв "a", второе число задаёт количество букв "b" и так далее.

Формат вывода

Выведите единственное целое число — максимально возможную хорошесть строки, которую можно собрать из имеющихся дощечек.

Пример 1

Ввод 	Вывод 
3	2
1	
1	
1	

Пример 2

Ввод 	Вывод 
2	3
3	
4	

Примечания

В первом тесте имеется по одной дощечке с каждой из 3 различных букв. Ответ 2 достигается на строке "abc"

6. Операционные системы lite

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Васин жесткий диск состоит из M секторов. Вася последовательно устанавливал на него различные операционные системы следующим методом: он создавал новый раздел диска из последовательных секторов, начиная с сектора номер a_i и до сектора b_i включительно, и устанавливал на него очередную систему. При этом, если очередной раздел хотя бы по одному сектору пересекается с каким-то ранее созданным разделом, то ранее созданный раздел «затирается», и операционная система, которая на него была установлена, больше не может быть загружена. Напишите программу, которая по информации о том, какие разделы на диске создавал Вася, определит, сколько в итоге работоспособных операционных систем установлено и работает в настоящий момент на Васином компьютере.



Формат ввода

Сначала вводятся натуральное число M — количество секторов на жестком диске ($1 \leq M \leq 10^9$) и целое число N — количество разделов, которое последовательно создавал Вася ($0 \leq N \leq 1000$). Далее идут N пар чисел a_i и b_i , задающих номера начального и конечного секторов раздела ($1 \leq a_i \leq b_i \leq M$).



Формат вывода

Выведите одно число — количество работающих операционных систем на Васином компьютере.

Пример 1

Ввод 	Вывод 
10	1
3	
1 3	
4 7	
3 4	

Пример 2

Ввод 	Вывод 
10	3
4	
1 3	
4 5	
7 8	
4 6	

7. SNTP

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Для того чтобы компьютеры поддерживали актуальное время, они могут обращаться к серверам точного времени SNTP (Simple Network Time Protocol). К сожалению, компьютер не может просто получить время у сервера, потому что информация по сети передаётся не мгновенно: пока сообщение с текущим временем дойдёт до компьютера, оно потеряет свою актуальность. Протокол взаимодействия клиента (компьютера, запрашивающего точное время) и сервера (компьютера, выдающего точное время) выглядит следующим образом:

- Клиент отправляет запрос на сервер и запоминает время отправления A (по клиентскому времени).
- Сервер получает запрос в момент времени B (по точному серверному времени) и отправляет клиенту сообщение, содержащее время B.
- Клиент получает ответ на свой запрос в момент времени C (по клиентскому времени) и запоминает его. Теперь клиент, из предположения, что сетевые задержки при передаче сообщений от клиента серверу и от сервера клиенту одинаковы, может определить и установить себе точное время, используя известные значения A, B, C.

Вам предстоит реализовать алгоритм, с точностью до секунды определяющий точное время для установки на клиенте по известным A, B и C. При необходимости округлите результат до целого числа секунд по правилам арифметики (в меньшую сторону, если дробная часть числа меньше 1/2, иначе в большую сторону).

Возможно, что, пока клиент ожидал ответа, по клиентскому времени успели наступить новые сутки, однако известно, что между отправкой клиентом запроса и получением ответа от сервера прошло менее 24 часов.

Формат ввода

Программа получает на вход три временные метки A, B, C, по одной в каждой строке. Все временные метки представлены в формате «hh:mm:ss», где «hh» — это часы, «mm» — минуты, «ss» — секунды. Часы, минуты и секунды записываются ровно двумя цифрами каждое (возможно, с дополнительными нулями в начале числа).

Формат вывода

Программа должна вывести одну временную метку в формате, описанном во входных данных, — вычисленное точное время для установки на клиенте. В выводе не должно быть пробелов, пустых строк в начале вывода.

Пример

Ввод

Вывод

15:01:00
18:09:45
15:01:40

18:10:05

8. Минимальный прямоугольник

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

На клетчатой плоскости закрашено K клеток. Требуется найти минимальный по площади прямоугольник, со сторонами, параллельными линиям сетки, покрывающий все закрашенные клетки.

Формат ввода

Во входном файле, на первой строке, находится число K ($1 \leq K \leq 100$). На следующих K строках находятся пары чисел X_i и Y_i – координаты закрашенных клеток ($|X_i|, |Y_i| \leq 10^9$).

Формат вывода

Выведите в выходной файл координаты левого нижнего и правого верхнего углов прямоугольника.

Пример

<div>Ввод<div>📄</div></div>	<div>Вывод<div>📄</div></div>
3	1 1 5 10
1 1	
1 10	
5 5	

9. Сумма в прямоугольнике

Ограничение времени	3 секунды
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Вам необходимо ответить на запросы узнать сумму всех элементов числовой матрицы N×M в прямоугольнике с левым верхним углом (x_1, y_1) и правым нижним (x_2, y_2)



Формат ввода

В первой строке находится числа N, M размеры матрицы $(1 \leq N, M \leq 1000)$ и K — количество запросов $(1 \leq K \leq 100000)$. Каждая из следующих N строк содержит по M чисел` — элементы соответствующей строки матрицы (по модулю не превосходят 1000). Последующие K строк содержат по 4 целых числа, разделенных пробелом $x_1\ y_1\ x_2\ y_2$ — запрос на сумму элементов матрице в прямоугольнике $(1 \leq x_1 \leq x_2 \leq N, 1 \leq y_1 \leq y_2 \leq M)$

Формат вывода

Для каждого запроса на отдельной строке выведите его результат — сумму всех чисел в элементов матрице в прямоугольнике $(x_1, y_1), (x_2, y_2)$

Пример

Ввод		Вывод	
3 3 2		28	
1 2 3		21	
4 5 6			
7 8 9			
2 2 3 3			
1 1 2 3			

10. Скучная лекция

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Лёша сидел на лекции. Ему было невероятно скучно. Голос лектора казался таким далеким и незаметным... Чтобы окончательно не уснуть, он взял листок и написал на нём свое любимое слово. Чуть ниже он повторил своё любимое слово, без первой буквы. Ещё ниже он снова написал своё любимое слово, но в этот раз без двух первых и последней буквы.

Тут ему пришла в голову мысль — времени до конца лекции все равно ещё очень много, почему бы не продолжить выписывать всеми возможными способами это слово без какой-то части с начала и какой-то части с конца?

После лекции Лёша рассказал Максу, как замечательно он скоротал время. Максу стало интересно посчитать, сколько букв каждого вида встречается у Лёши в листочке. Но к сожалению, сам листочек куда-то запропастился.

Макс хорошо знает любимое слово Лёши, а ещё у него не так много свободного времени, как у его друга, так что помогите ему быстро восстановить, сколько раз Лёше пришлось выписать каждую букву.

Формат ввода

На вход подаётся строка, состоящая из строчных латинских букв — любимое слово Лёши. Длина строки лежит в пределах от 5 до 100 000 символов.

Формат вывода

Для каждой буквы на листочке Лёши, выведите её, а затем через двоеточие и пробел сколько раз она встретилась в выписанных Лёшей словах (см. формат вывода в примерах). Буквы должны следовать в алфавитном порядке. Буквы, не встречающиеся на листочке, выводить не нужно.

Пример 1

Ввод 	Вывод 
hello	e: 8 h: 5 l: 17 o: 5

Пример 2

Ввод 	Вывод 
abacaba	a: 44 b: 24 c: 16

Примечания

Пояснение к первому примеру. Если любимое Лёшино слово — "hello", то на листочке у Лёши будут выписаны следующие слова: "hello"

"hell"

"ello"

"hel"

"ell"

"llo"

"he"

"el"

"ll"

"lo"

"h"

"e"

"l"

"l"

"o"

Среди этих слов 8 раз встречается буква "e", 5 раз — буква "h", 17 раз — буква "l" и 5 раз буква "o".

11. Стек с защитой от ошибок

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Научитесь пользоваться стандартной структурой данных `stack` для целых чисел. Напишите программу, содержащую описание стека и моделирующую работу стека, реализовав все указанные здесь методы. Программа считывает последовательность команд и в зависимости от команды выполняет ту или иную операцию. После выполнения каждой команды программа должна вывести одну строчку. Возможные команды для программы:

`push n`
Добавить в стек число `n` (значение `n` задается после команды). Программа должна вывести `ok`.

`pop`
Удалить из стека последний элемент. Программа должна вывести его значение.

`back`
Программа должна вывести значение последнего элемента, не удаляя его из стека.

`size`
Программа должна вывести количество элементов в стеке.

`clear`
Программа должна очистить стек и вывести `ok`.

`exit`
Программа должна вывести `bye` и завершить работу.

Перед исполнением операций `back` и `pop` программа должна проверять, содержится ли в стеке хотя бы один элемент. Если во входных данных встречается операция `back` или `pop`, и при этом стек пуст, то программа должна вместо числового значения вывести строку `error`.



Формат ввода

Вводятся команды управления стеком, по одной на строке

Формат вывода

Программа должна вывести протокол работы стека, по одному сообщению на строке



Пример 1

Ввод 	Вывод 
<code>push 1</code>	<code>ok</code>
<code>back</code>	<code>1</code>
<code>exit</code>	<code>bye</code>

Пример 2

Ввод 	Вывод 
<code>size</code>	<code>0</code>
<code>push 1</code>	<code>ok</code>
<code>size</code>	<code>1</code>
<code>push 2</code>	<code>ok</code>
<code>size</code>	<code>2</code>
<code>push 3</code>	<code>ok</code>
<code>size</code>	<code>3</code>
<code>exit</code>	<code>bye</code>

Пример 3

Ввод 	Вывод 
<code>push 3</code>	<code>ok</code>
<code>push 14</code>	<code>ok</code>
<code>size</code>	<code>2</code>
<code>clear</code>	<code>ok</code>
<code>push 1</code>	<code>ok</code>
<code>back</code>	<code>1</code>
<code>push 2</code>	<code>ok</code>
<code>back</code>	<code>2</code>
<code>pop</code>	<code>2</code>
<code>size</code>	<code>1</code>
<code>pop</code>	<code>1</code>
<code>size</code>	<code>0</code>
<code>exit</code>	<code>bye</code>

12. Правильная скобочная последовательность

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Рассмотрим последовательность, состоящую из круглых, квадратных и фигурных скобок. Программа должна определить, является ли данная скобочная последовательность правильной. Пустая последовательность является правильной. Если A – правильная, то последовательности (A), [A], {A} – правильные. Если A и B – правильные последовательности, то последовательность AB – правильная.



Формат ввода

В единственной строке записана скобочная последовательность, содержащая не более 100000 скобок.



Формат вывода

Если данная последовательность правильная, то программа должна вывести строку yes, иначе строку no.



Пример 1

Ввод 	Вывод 
() []	yes

Пример 2

Ввод 	Вывод 
([]]	no

Пример 3

Ввод 	Вывод 
(no

13. Постфиксная запись

Ограничение времени	1 секунда
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В постфиксной записи (или обратной польской записи) операция записывается после двух операндов. Например, сумма двух чисел A и B записывается как A B +. Запись B C + D * обозначает привычное нам (B + C) * D, а запись A B C + D * + означает A + (B + C) * D. Достоинство постфиксной записи в том, что она не требует скобок и дополнительных соглашений о приоритете операторов для своего чтения.

Формат ввода

В единственной строке записано выражение в постфиксной записи, содержащее цифры и операции +, -, *. Цифры и операции разделяются пробелами. В конце строки может быть произвольное количество пробелов.

Формат вывода

Необходимо вывести значение записанного выражения.

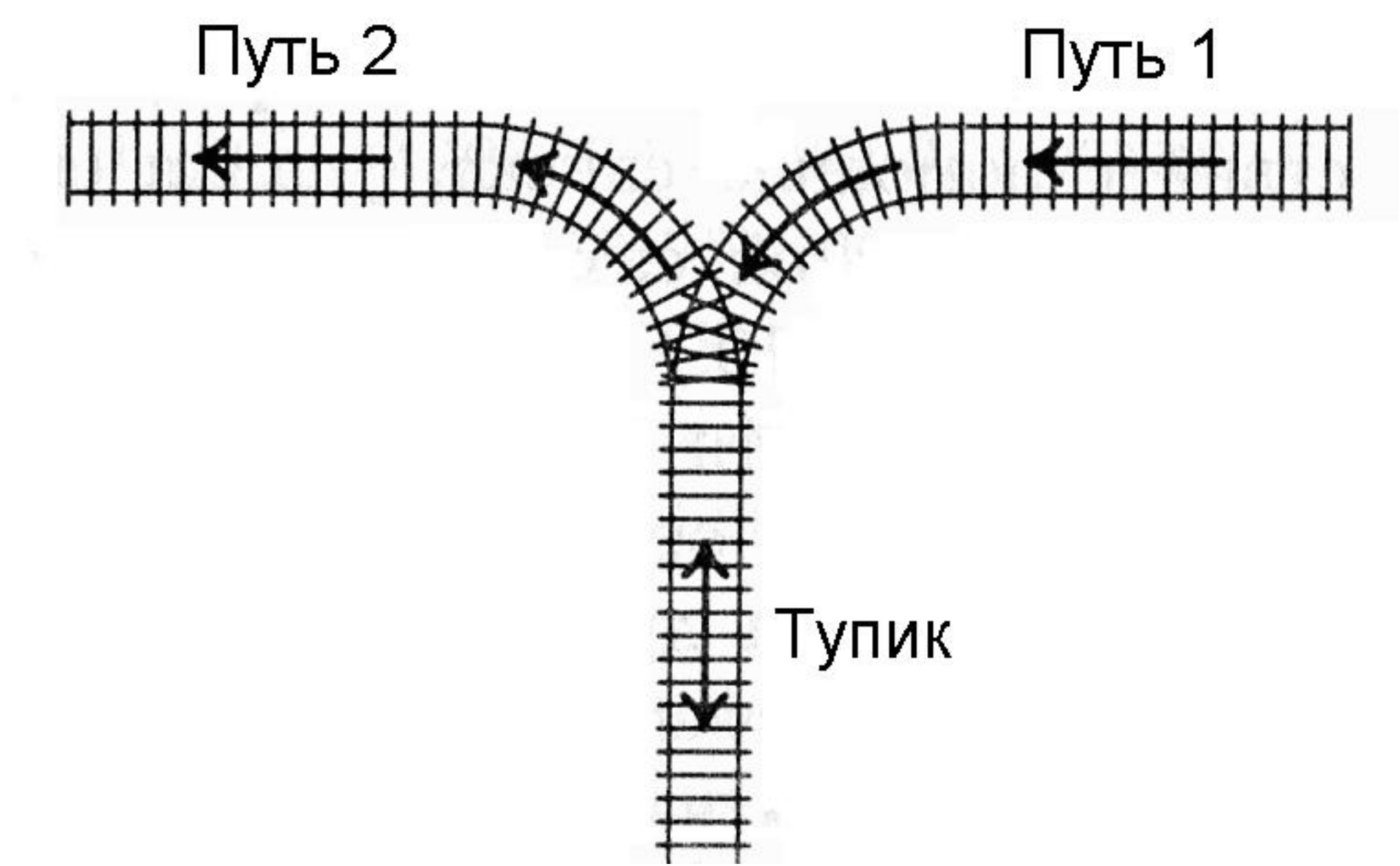
Пример

Ввод	<div>8 9 + 1 7 - *</div>	Вывод	<div>-102</div>
------	--------------------------	-------	-----------------

14. Сортировка вагоновlite

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

К тупику со стороны пути 1 (см. рисунок) подъехал поезд. Разрешается отцепить от поезда один или сразу несколько первых вагонов и завезти их в тупик (при желании, можно даже завезти в тупик сразу весь поезд). После этого часть из этих вагонов вывезти в сторону пути 2. После этого можно завезти в тупик еще несколько вагонов и снова часть оказавшихся вагонов вывезти в сторону пути 2. И так далее (так, что каждый вагон может лишь один раз заехать с пути 1 в тупик, а затем один раз выехать из тупика на путь 2). Заезжать в тупик с пути 2 или выезжать из тупика на путь 1 запрещается. Нельзя с пути 1 попасть на путь 2, не заезжая в тупик.



Известно, в каком порядке изначально идут вагоны поезда. Требуется с помощью указанных операций сделать так, чтобы вагоны поезда шли по порядку (сначала первый, потом второй и т.д., считая от головы поезда, едущего по пути 2 в сторону от тупика). Напишите программу, определяющую, можно ли это сделать.

Формат ввода

Вводится число N — количество вагонов в поезде ($1 \leq N \leq 100$). Далее идут номера вагонов в порядке от головы поезда, едущего по пути 1 в сторону тупика. Вагоны пронумерованы натуральными числами от 1 до N , каждое из которых встречается ровно один раз.

Формат вывода

Если сделать так, чтобы вагоны шли в порядке от 1 до N , считая от головы поезда, когда поезд поедет по пути 2 из тупика, можно, выведите сообщение YES, если это сделать нельзя, выведите NO.

Пример 1

Ввод	Вывод
3	YES
3 2 1	

Пример 2

Ввод	Вывод
4	YES
4 1 3 2	

Пример 3

Ввод	Вывод
3	NO
2 3 1	

15. Великое Лайнландское переселение

Ограничение времени	1 секунда
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Лайнландия представляет из себя одномерный мир, являющийся прямой, на котором располагаются N городов, последовательно пронумерованных от 0 до $N - 1$. Направление в сторону от первого города к нулевому названо западным, а в обратную — восточным.

Когда в Лайнландии неожиданно начался кризис, все были жители мира стали испытывать глубокое смятение. По всей Лайнландии стали ходить слухи, что на востоке живётся лучше, чем на западе.

Так и началось Великое Лайнландское переселение. Обитатели мира целыми городами отправились на восток, покинув родные улицы, и двигались до тех пор, пока не приходили в город, в котором средняя цена проживания была меньше, чем в родном.

Формат ввода

В первой строке дано одно число N ($2 \leq N \leq 10^5$) — количество городов в Лайнландии. Во второй строке дано N чисел a_i ($0 \leq a_i \leq 10^9$) — средняя цена проживания в городах с нулевого по $(N - 1)$ -ый соответственно.

Формат вывода

Для каждого города в порядке с нулевого по $(N - 1)$ -ый выведите номер города, в который переселятся его изначальные жители. Если жители города не остановятся в каком-либо другом городе, отправившись в Восточное Бесконечное Ничто, выведите -1 .

Пример

Ввод	<div>📄</div>	Вывод	<div>📄</div>
10		-1 4 3 4 -1 6 9 8 9 -1	
1 2 3 2 1 4 2 5 3 1			

16. Очередь с защитой от ошибок

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Научитесь пользоваться стандартной структурой данных queue для целых чисел. Напишите программу, содержащую описание очереди и моделирующую работу очереди, реализовав все указанные здесь методы. Программа считывает последовательность команд и в зависимости от команды выполняет ту или иную операцию. После выполнения каждой команды программа должна вывести одну строку.

Возможные команды для программы:

push n
Добавить в очередь число n (значение n задается после команды). Программа должна вывести ok.

pop
Удалить из очереди первый элемент. Программа должна вывести его значение.

front
Программа должна вывести значение первого элемента, не удаляя его из очереди.

size
Программа должна вывести количество элементов в очереди.

clear
Программа должна очистить очередь и вывести ok.

exit
Программа должна вывести bye и завершить работу.

Перед исполнением операций front и pop программа должна проверять, содержится ли в очереди хотя бы один элемент. Если во входных данных встречается операция front или pop, и при этом очередь пуста, то программа должна вместо числового значения вывести строку error.

Формат ввода

Вводятся команды управления очередью, по одной на строке

Формат вывода

Требуется вывести протокол работы очереди, по одному сообщению на строке

Пример 1

Ввод	Вывод
push 1	ok
front	1
exit	bye

Пример 2

Ввод	Вывод
size	0
push 1	ok
size	1
push 2	ok
size	2
push 3	ok
size	3
exit	bye

Пример 3

Ввод	Вывод
push 3	ok
push 14	ok
size	2
clear	ok
push 1	ok
front	1
push 2	ok
front	1
pop	1
size	1
pop	2
size	0
exit	bye

17. Игра в пьяницу

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В игре в пьяницу карточная колода раздается поровну двум игрокам. Далее они вскрывают по одной верхней карте, и тот, чья карта старше, забирает себе обе вскрытые карты, которые кладутся под низ его колоды. Тот, кто остается без карт – проигрывает. Для простоты будем считать, что все карты различны по номиналу, а также, что самая младшая карта побеждает самую старшую карту ("шестерка берет туза"). Игрок, который забирает себе карты, сначала кладет под низ своей колоды карту первого игрока, затем карту второго игрока (то есть карта второго игрока оказывается внизу колоды). Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает. В игре участвует 10 карт, имеющих значения от 0 до 9, большая карта побеждает меньшую, карта со значением 0 побеждает карту 9.



Формат ввода

Программа получает на вход две строки: первая строка содержит 5 чисел, разделенных пробелами — номера карт первого игрока, вторая – аналогично 5 карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой.

Формат вывода

Программа должна определить, кто выигрывает при данной раздаче, и вывести слово first или second, после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении 10^6 ходов игра не заканчивается, программа должна вывести слово botva.



Пример 1

Ввод 	Вывод 
1 3 5 7 9 2 4 6 8 0	second 5

Пример 2

Ввод 	Вывод 
2 4 6 8 0 1 3 5 7 9	first 5

Пример 3

Ввод 	Вывод 
1 7 3 9 4 5 8 0 2 6	second 23

18. Дек с защитой от ошибок

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Научитесь пользоваться стандартной структурой данных deque для целых чисел. Напишите программу, содержащую описание дека и моделирующую работу дека, реализовав все указанные здесь методы. Программа считывает последовательность команд и в зависимости от команды выполняет ту или иную операцию. После выполнения каждой команды программа должна вывести одну строку. Возможные команды для программы:

push_front n
Добавить (положить) в начало дека новый элемент. Программа должна вывести ok.

push_back n
Добавить (положить) в конец дека новый элемент. Программа должна вывести ok.

pop_front
Извлечь из дека первый элемент. Программа должна вывести его значение.

pop_back
Извлечь из дека последний элемент. Программа должна вывести его значение.

front
Узнать значение первого элемента (не удаляя его). Программа должна вывести его значение.

back
Узнать значение последнего элемента (не удаляя его). Программа должна вывести его значение.

size
Вывести количество элементов в деке.

clear
Очистить дек (удалить из него все элементы) и вывести ok.

exit
Программа должна вывести bye и завершить работу.

Гарантируется, что количество элементов в деке в любой момент не превосходит 100. Перед исполнением операций pop_front, pop_back, front, back программа должна проверять, содержится ли в деке хотя бы один элемент. Если во входных данных встречается операция pop_front, pop_back, front, back, и при этом дек пуст, то программа должна вместо числового значения вывести строку error.

Формат ввода

Вводятся команды управления деком, по одной на строке.

Формат вывода

Требуется вывести протокол работы дека, по одному сообщению на строке

Пример 1

Ввод	Вывод
push_back 1	ok
back	1
exit	bye

Пример 2

Ввод	Вывод
size	0
push_back 1	ok
size	1
push_back 2	ok
size	2
push_front 3	ok
size	3
exit	bye

Пример 3

Ввод	Вывод
push_back 3	ok
push_front 14	ok
size	2
clear	ok
push_front 1	ok
back	1
push_back 2	ok
front	1
pop_back	2
size	1
pop_front	1
size	0
exit	bye

19. Хипуй

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В этой задаче вам необходимо самостоятельно (не используя соответствующие классы и функции стандартной библиотеки) организовать структуру данных *Неар* для хранения целых чисел, над которой определены следующие операции: а) `Insert(k)` – добавить в *Неар* число *k* ; б) `Extract` достать из *Неар* наибольшее число (удалив его при этом).



Формат ввода

В первой строке содержится количество команд *N* ($1 \leq N \leq 100000$), далее следуют *N* команд, каждая в своей строке. Команда может иметь формат: “0 <число>” или “1”, обозначающий, соответственно, операции `Insert(<число>)` и `Extract`. Гарантируется, что при выполнении команды `Extract` в структуре находится по крайней мере один элемент.



Формат вывода

Для каждой команды извлечения необходимо отдельной строкой вывести число, полученное при выполнении команды `Extract`.

Пример 1

Ввод 	Вывод 
2	10000
0 10000	
1	

Пример 2

Ввод 	Вывод 
14	345
0 1	4346
0 345	2435
1	365
0 4346	235
1	5
0 2435	1
1	
0 235	
0 5	
0 365	
1	
1	
1	
1	

20. Пирамидальная сортировка

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Отсортируйте данный массив. Используйте пирамидальную сортировку.

Формат ввода

Первая строка входных данных содержит количество элементов в массиве N, $N \leq 10^5$. Далее задаются N целых чисел, не превосходящих по абсолютной величине 10^9 .

Формат вывода

Выведите эти числа в порядке неубывания.

Пример 1

Ввод 	Вывод 
1	1
1	

Пример 2

Ввод 	Вывод 
2	1 3
3 1	

21. Три единицы подряд

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

По данному числу N определите количество последовательностей из нулей и единиц длины N , в которых никакие три единицы не стоят рядом.

Формат ввода

Во входном файле написано натуральное число N , не превосходящее 35 .

Формат вывода

Выведите количество искомых последовательностей. Гарантируется, что ответ не превосходит $2^{31}-1$.

Пример

Ввод 	Вывод 
1	2

22. Кузнечик

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

У одного из студентов в комнате живёт кузнечик, который очень любит прыгать по клетчатой одномерной доске. Длина доски — N клеток. К его сожалению, он умеет прыгать только на $1, 2, \dots, k$ клеток вперёд. Однажды студентам стало интересно, сколькими способами кузнечик может допрыгать из первой клетки до последней. Помогите им ответить на этот вопрос.

Формат ввода

В первой и единственной строке входного файла записано два целых числа — N и k ($1 \leq N \leq 30, 1 \leq k \leq 10$).

Формат вывода

Выведите одно число — количество способов, которыми кузнечик может допрыгать из первой клетки до последней.

Пример

Ввод	Вывод
8 2	21

23. Калькулятор

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Имеется калькулятор, который выполняет следующие операции:

- умножить число X на 2 ;
- умножить число X на 3 ;
- прибавить к числу X единицу.

Определите, какое наименьшее количество операций требуется, чтобы получить из числа I число N .

Формат ввода

Во входном файле написано натуральное число N , не превосходящее 10^6 .

Формат вывода

В первой строке выходного файла выведите минимальное количество операций. Во второй строке выведите числа, последовательно получающиеся при выполнении операций. Первое из них должно быть равно 1 , а последнее N . Если решений несколько, выведите любое.

Пример 1

Ввод	Вывод
1	0 1

Пример 2

Ввод	Вывод
5	3 1 3 4 5

24. Покупка билетов

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

За билетами на премьеру нового мюзикла выстроилась очередь из N человек, каждый из которых хочет купить 1 билет. На всю очередь работала только одна касса, поэтому продажа билетов шла очень медленно, приводя «постояльцев» очереди в отчаяние. Самые сообразительные быстро заметили, что, как правило, несколько билетов в одни руки кассир продаёт быстрее, чем когда эти же билеты продаются по одному. Поэтому они предложили нескольким подряд стоящим людям отдавать деньги первому из них, чтобы он купил билеты на всех. Однако для борьбы со спекулянтами кассир продавала не более 3-х билетов в одни руки, поэтому договориться таким образом между собой могли лишь 2 или 3 подряд стоящих человека.

Известно, что на продажу i-му человеку из очереди одного билета кассир тратит A_i секунд, на продажу двух билетов — B_i секунд, трех билетов — C_i секунд. Напишите программу, которая подсчитает минимальное время, за которое могли быть обслужены все покупатели.

Обратите внимание, что билеты на группу объединившихся людей всегда покупает первый из них. Также никто в целях ускорения не покупает лишних билетов (то есть билетов, которые никому не нужны).



Формат ввода

На вход программы поступает сначала число N — количество покупателей в очереди ($1 \leq N \leq 5000$). Далее идет N троек натуральных чисел A_i, B_i, C_i . Каждое из этих чисел не превышает 3600. Люди в очереди нумеруются, начиная от кассы.

Формат вывода

Требуется вывести одно число — минимальное время в секундах, за которое могли быть обслужены все покупатели.

Пример

Ввод <div></div>	Вывод <div></div>
5 5 10 15 2 10 15 5 5 5 20 20 1 20 1 1	12

25. Гвоздики

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В дощечке в один ряд вбиты гвоздики. Любые два гвоздика можно соединить ниточкой. Требуется соединить некоторые пары гвоздиков ниточками так, чтобы к каждому гвоздику была привязана хотя бы одна ниточка, а суммарная длина всех ниточек была минимальна.

Формат ввода

В первой строке входных данных записано число N — количество гвоздиков ($2 \leq N \leq 100$). В следующей строке заданы N чисел — координаты всех гвоздиков (неотрицательные целые числа, не превосходящие 10000).

Формат вывода

Выведите единственное число — минимальную суммарную длину всех ниточек.

Пример

Ввод	Вывод
6	5
3 13 12 4 14 6	

26. Самый дешевый путь

Ограничение времени	1 секунда
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В каждой клетке прямоугольной таблицы $N \times M$ записано некоторое число. Изначально игрок находится в левой верхней клетке. За один ход ему разрешается перемещаться в соседнюю клетку либо вправо, либо вниз (влево и вверх перемещаться запрещено). При проходе через клетку с игрока берут столько килограммов еды, какое число записано в этой клетке (еду берут также за первую и последнюю клетки его пути). Требуется найти минимальный вес еды в килограммах, отдав которую игрок может попасть в правый нижний угол.

Формат ввода

Вводятся два числа N и M — размеры таблицы ($1 \leq N \leq 20, 1 \leq M \leq 20$). Затем идет N строк по M чисел в каждой — размеры штрафов в килограммах за прохождение через соответствующие клетки (числа от 0 до 100).

Формат вывода

Выведите минимальный вес еды в килограммах, отдав которую можно попасть в правый нижний угол.

Пример

Ввод 	Вывод 
5 5	11
1 1 1 1 1	
3 100 100 100 100	
1 1 1 1 1	
2 2 2 2 1	
1 1 1 1 1	

27. Вывести маршрут максимальной стоимости

Ограничение времени	1 секунда
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В левом верхнем углу прямоугольной таблицы размером $N \times M$ находится черепашка. В каждой клетке таблицы записано некоторое число. Черепашка может перемещаться вправо или вниз, при этом маршрут черепашки заканчивается в правом нижнем углу таблицы.

Подсчитаем сумму чисел, записанных в клетках, через которую проползла черепашка (включая начальную и конечную клетку). Найдите наибольшее возможное значение этой суммы и маршрут, на котором достигается эта сумма.

Формат ввода

В первой строке входных данных записаны два натуральных числа N и M , не превосходящих 100 — размеры таблицы. Далее идет N строк, каждая из которых содержит M чисел, разделенных пробелами — описание таблицы. Все числа в клетках таблицы целые и могут принимать значения от 0 до 100.

Формат вывода

Первая строка выходных данных содержит максимальную возможную сумму, вторая — маршрут, на котором достигается эта сумма. Маршрут выводится в виде последовательности, которая должна содержать $N-1$ букву D, означающую передвижение вниз и $M-1$ букву R, означающую передвижение направо. Если таких последовательностей несколько, необходимо вывести ровно одну (любую) из них.

Пример

Ввод 	Вывод 
5 5	74
9 9 9 9 9	D D R R R R D D
3 0 0 0 0	
9 9 9 9 9	
6 6 6 6 8	
9 9 9 9 9	

28. Ход конём

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Дана прямоугольная доска $N \times M$ (N строк и M столбцов). В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски. В данной задаче конь может перемещаться на две клетки вниз и одну клетку вправо или на одну клетку вниз и две клетки вправо. Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.

Формат ввода

Входной файл содержит два натуральных числа N и M ($1 \leq N, M \leq 50$).

Формат вывода

В выходной файл выведите единственное число — количество способов добраться конём до правого нижнего угла доски.

Пример 1

Ввод	Вывод
3 2	1

Пример 2

Ввод	Вывод
31 34	293930

29. Кафе

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Около Петиного университета недавно открылось новое кафе, в котором действует следующая система скидок: при каждой покупке более чем на 100 рублей покупатель получает купон, дающий право на один бесплатный обед (при покупке на сумму 100 рублей и меньше такой купон покупатель не получает). Однажды Пете на глаза попался прейскурант на ближайшие N дней. Внимательно его изучив, он решил, что будет обедать в этом кафе все N дней, причем каждый день он будет покупать в кафе ровно один обед. Однако стипендия у Пети небольшая, и поэтому он хочет по максимуму использовать предоставляемую систему скидок так, чтобы его суммарные затраты были минимальны. Требуется найти минимально возможную суммарную стоимость обедов и номера дней, в которые Пете следует воспользоваться купонами.


Формат ввода

В первой строке входного файла записано целое число N ($0 \leq N \leq 100$). В каждой из последующих N строк записано одно целое число, обозначающее стоимость обеда в рублях на соответствующий день. Стоимость — неотрицательное целое число, не превосходящее 300.

Формат вывода

В первой строке выдайте минимальную возможную суммарную стоимость обедов. Во второй строке выдайте два числа K_1 и K_2 — количество купонов, которые останутся неиспользованными у Пети после этих N дней и количество использованных им купонов соответственно. В последующих K_2 строках выдайте в возрастающем порядке номера дней, когда Пете следует воспользоваться купонами. Если существует несколько решений с минимальной суммарной стоимостью, то выдайте то из них, в котором значение K_1 максимально (на случай, если Петя когда-нибудь ещё решит заглянуть в это кафе). Если таких решений несколько, выведите любое из них.

Пример

Ввод <div></div>	Вывод <div></div>
5	235
35	0 1
40	5
101	
59	
63	

30. НОП с восстановлением ответа

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Даны две последовательности, требуется найти и вывести их наибольшую общую подпоследовательность.



Формат ввода

В первой строке входных данных содержится число N — длина первой последовательности ($1 \leq N \leq 1000$). Во второй строке заданы члены первой последовательности (через пробел) — целые числа, не превосходящие 10000 по модулю. В третьей строке записано число M — длина второй последовательности ($1 \leq M \leq 1000$). В четвертой строке задаются члены второй последовательности (через пробел) — целые числа, не превосходящие 10000 по модулю.


Формат вывода

Требуется вывести наибольшую общую подпоследовательность данных последовательностей, через пробел.

Пример 1

Ввод 	Вывод 
3	2 3
1 2 3	
3	
2 3 1	

Пример 2

Ввод 	Вывод 
3	1
1 2 3	
3	
3 2 1	

31. Поиск в глубину

	Все языки	Python 3.6
Ограничение времени	2 секунды	5 секунд
Ограничение памяти	256Mb	256Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

Дан неориентированный граф, возможно, с петлями и кратными ребрами. Необходимо построить компоненту связности, содержащую первую вершину.

Формат ввода

В первой строке записаны два целых числа N ($1 \leq N \leq 10^3$) и M ($0 \leq M \leq 5 * 10^5$) — количество вершин и ребер в графе. В последующих M строках перечислены ребра — пары чисел, определяющие номера вершин, которые соединяют ребра.

Формат вывода

В первую строку выходного файла выведите число K — количество вершин в компоненте связности. Во вторую строку выведите K целых чисел — вершины компоненты связности, перечисленные в порядке возрастания номеров.

Пример

Ввод	Вывод
4 5 2 2 3 4 2 3 1 3 2 4	4 1 2 3 4

32. Компоненты связности

	Все языки	Python 3.6
Ограничение времени	2 секунды	5 секунд
Ограничение памяти	256Mb	256Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

Дан неориентированный невзвешенный граф. Необходимо посчитать количество его компонент связности и вывести их.



Формат ввода

Во входном файле записано два числа N и M ($0 < N \leq 100000$, $0 \leq M \leq 100000$). В следующих M строках записаны по два числа i и j ($1 \leq i, j \leq N$), которые означают, что вершины i и j соединены ребром.



Формат вывода

В первой строчке выходного файла выведите количество компонент связности. Далее выведите сами компоненты связности в следующем формате: в первой строке количество вершин в компоненте, во второй - сами вершины в произвольном порядке.

Пример 1

Ввод 	Вывод 
6 4	3
3 1	3
1 2	1 2 3
5 4	2
2 3	4 5
	1
	6

Пример 2

Ввод 	Вывод 
6 4	2
4 2	5
1 4	1 2 3 4 6
6 4	1
3 6	5

33. Списывание

	Все языки	Python 3.6
Ограничение времени	2 секунды	5 секунд
Ограничение памяти	256Mb	256Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

Во время контрольной работы профессор Флойд заметил, что некоторые студенты обмениваются записками. Сначала он хотел поставить им всем двойки, но в тот день профессор был добрым, а потому решил разделить студентов на две группы: списывающих и дающих списывать, и поставить двойки только первым. У профессора записаны все пары студентов, обменявшихся записками. Требуется определить, сможет ли он разделить студентов на две группы так, чтобы любой обмен записками осуществлялся от студента одной группы студенту другой группы.

Формат ввода

В первой строке находятся два числа N и M — количество студентов и количество пар студентов, обменивающихся записками ($1 \leq N \leq 10^2$, $0 \leq M \leq N(N-1)/2$). Далее в M строках расположены описания пар студентов: два числа, соответствующие номерам студентов, обменивающихся записками (нумерация студентов идёт с 1). Каждая пара студентов перечислена не более одного раза.

Формат вывода

Необходимо вывести ответ на задачу профессора Флойда. Если возможно разделить студентов на две группы - выведите YES; иначе выведите NO.

Пример 1

Ввод 	Вывод 
3 2 1 2 2 3	YES

Пример 2

Ввод 	Вывод 
3 3 1 2 2 3 1 3	NO

34. Топологическая сортировка

Ограничение времени	2 секунды
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Дан ориентированный граф. Необходимо построить топологическую сортировку.

Формат ввода

В первой строке входного файла два натуральных числа N и M ($1 \leq N, M \leq 100\,000$) — количество вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат вывода

Выведите любую топологическую сортировку графа в виде последовательности номеров вершин (перестановка чисел от 1 до N). Если топологическую сортировку графа построить невозможно, выведите -1.

Пример

Ввод	Вывод
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5

35. Поиск цикла

	Все языки	Python 3.6
Ограничение времени	2 секунды	5 секунд
Ограничение памяти	256Mb	256Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

Дан неориентированный граф. Требуется определить, есть ли в нем цикл, и, если есть, вывести его.



Формат ввода

В первой строке дано одно число n — количество вершин в графе ($1 \leq n \leq 500$). Далее в n строках задан сам граф матрицей смежности.



Формат вывода

Если в исходном графе нет цикла, то выведите «NO». Иначе, в первой строке выведите «YES», во второй строке выведите число k — количество вершин в цикле, а в третьей строке выведите k различных чисел — номера вершин, которые принадлежат циклу в порядке обхода (обход можно начинать с любой вершины цикла). Если циклов несколько, то выведите любой.



Пример 1

Ввод 	Вывод 
3	YES
0 1 1	3
1 0 1	3 2 1
1 1 0	

Пример 2

Ввод 	Вывод 
4	NO
0 0 1 0	
0 0 0 1	
1 0 0 0	
0 1 0 0	

Пример 3

Ввод 	Вывод 
5	YES
0 1 0 0 0	3
1 0 0 0 0	5 4 3
0 0 0 1 1	
0 0 1 0 1	
0 0 1 1 0	

36. Длина кратчайшего пути

	Все языки	Python 3.6
Ограничение времени	1 секунда	5 секунд
Ограничение памяти	64Mb	256Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

В неориентированном графе требуется найти длину минимального пути между двумя вершинами.

Формат ввода

Первым на вход поступает число N – количество вершин в графе ($1 \leq N \leq 100$). Затем записана матрица смежности (0 обозначает отсутствие ребра, 1 – наличие ребра). Далее задаются номера двух вершин – начальной и конечной.

Формат вывода

Выведите L – длину кратчайшего пути (количество ребер, которые нужно пройти).
Если пути нет, нужно вывести -1.

Пример 1

Ввод

```
10
0 1 0 0 0 0 0 0 0 0
1 0 0 1 1 0 1 0 0 0
0 0 0 0 1 0 0 0 1 0
0 1 0 0 0 0 1 0 0 0
0 1 1 0 0 0 0 0 0 1
0 0 0 0 0 0 1 0 0 1
0 1 0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0 1 0
0 0 1 0 0 0 0 1 0 0
0 0 0 0 1 1 0 0 0 0
5 4
```

Вывод

```
2
```

Пример 2

Ввод

```
5
0 1 0 0 1
1 0 1 0 0
0 1 0 0 0
0 0 0 0 0
1 0 0 0 0
3 5
```

Вывод

```
3
```

37. Путь в графе

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В неориентированном графе требуется найти минимальный путь между двумя вершинами.

Формат ввода

Первым на вход поступает число N – количество вершин в графе ($1 \leq N \leq 100$). Затем записана матрица смежности (0 обозначает отсутствие ребра, 1 – наличие ребра). Далее задаются номера двух вершин – начальной и конечной.

Формат вывода

Выведите сначала L – длину кратчайшего пути (количество ребер, которые нужно пройти), а потом сам путь. Если путь имеет длину 0, то его выводить не нужно, достаточно вывести длину. Необходимо вывести путь (номера всех вершин в правильном порядке). Если пути нет, нужно вывести -1.

Пример

Ввод 	Вывод 
10	2
0 1 0 0 0 0 0 0 0 0	5 2 4
1 0 0 1 1 0 1 0 0 0	
0 0 0 0 1 0 0 0 1 0	
0 1 0 0 0 0 1 0 0 0	
0 1 1 0 0 0 0 0 0 1	
0 0 0 0 0 0 1 0 0 1	
0 1 0 1 0 1 0 0 0 0	
0 0 0 0 0 0 0 0 1 0	
0 0 1 0 0 0 0 1 0 0	
0 0 0 0 1 1 0 0 0 0	
5 4	

38. Блохи

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

На клеточном поле, размером NxM ($2 \leq N, M \leq 250$) сидит Q ($0 \leq Q \leq 10000$) блох в различных клетках. "Прием пищи" блохами возможен только в кормушке - одна из клеток поля, заранее известная. Блохи перемещаются по полю странным образом, а именно, прыжками, совпадающими с ходом обыкновенного шахматного коня. Длина пути каждой блохи до кормушки определяется как количество прыжков. Определить минимальное значение суммы длин путей блох до кормушки или, если собраться блохам у кормушки невозможно, то сообщить об этом. Сбор невозможен, если хотя бы одна из блох не может попасть к кормушке.



Формат ввода

В первой строке входного файла находится 5 чисел, разделенных пробелом: N, M, S, T, Q. N, M - размеры доски (отсчет начинается с 1); S, T - координаты клетки - кормушки (номер строки и столбца соответственно), Q - количество блох на доске. И далее Q строк по два числа - координаты каждой блохи.



Формат вывода

Содержит одно число - минимальное значение суммы длин путей или -1, если сбор невозможен.

Пример 1

Ввод 	Вывод 
2 2 1 1 1 2 2	-1

Пример 2

Ввод 	Вывод 
4 4 1 1 16 1 1 1 2 1 3 1 4 2 1 2 2 2 3 2 4 3 1 3 2 3 3 3 4 4 1 4 2 4 3 4 4	42

39. Путь спелеолога

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Пещера представлена кубом, разбитым на N частей по каждому измерению (то есть на N^3 кубических клеток). Каждая клетка может быть или пустой, или полностью заполненной камнем. Исходя из положения спелеолога в пещере, требуется найти, какое минимальное количество перемещений по клеткам ему требуется, чтобы выбраться на поверхность. Переходить из клетки в клетку можно, только если они обе свободны и имеют общую грань.

Формат ввода

В первой строке содержится число N ($1 \leq N \leq 30$). Далее следует N блоков. Блок состоит из пустой строки и N строк по N символов: # - обозначает клетку, заполненную камнями, точка - свободную клетку. Начальное положение спелеолога обозначено заглавной буквой S. Первый блок представляет верхний уровень пещеры, достижение любой свободной его клетки означает выход на поверхность. Выход на поверхность всегда возможен.

Формат вывода

Вывести одно число - длину пути до поверхности.

Пример

<div>Ввод<div><div></div></div></div>	<div>Вывод<div><div></div></div></div>
<div>3</div>	<div>6</div>
<div>### ### .#.</div>	
<div>... ###</div>	

Примечания

Нужно спуститься на уровень вниз, сделать два движения на запад, подняться на уровень вверх, сделать движение на юг, подняться на уровень вверх.

40. Метро

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Метрополитен состоит из нескольких линий метро. Все станции метро в городе пронумерованы натуральными числами от 1 до N. На каждой линии расположено несколько станций. Если одна и та же станция расположена сразу на нескольких линиях, то она является станцией пересадки и на этой станции можно пересесть с любой линии, которая через нее проходит, на любую другую (опять же проходящую через нее).
Напишите программу, которая по данному вам описанию метрополитена определит, с каким минимальным числом пересадок можно добраться со станции A на станцию B. Если данный метрополитен не соединяет все линии в одну систему, то может так получиться, что со станции A на станцию B добраться невозможно, в этом случае ваша программа должна это определить.

Формат ввода

Сначала вводится число N — количество станций метро в городе ($2 \leq N \leq 100$). Далее следует число M — количество линий метро ($1 \leq M \leq 20$). Далее идет описание M линий. Описание каждой линии состоит из числа P_i — количество станций на этой линии ($2 \leq P_i \leq 50$) и P_i чисел, задающих номера станций, через которые проходит линия (ни через какую станцию линия не проходит дважды).
Затем вводятся два различных числа: A — номер начальной станции, и B — номер станции, на которую нам нужно попасть. При этом если через станцию A проходит несколько линий, то мы можем спуститься на любую из них. Так же если через станцию B проходит несколько линий, то нам не важно, по какой линии мы приедем.

Формат вывода

Выведите минимальное количество пересадок, которое нам понадобится. Если добраться со станции A на станцию B невозможно, программа должна вывести одно число −1 (минус один).

Пример

Ввод	<div>📄</div>	Вывод	<div>📄</div>
5		0	
2			
4 1 2 3 4			
2 5 3			
3 1			