

Задача А. Путь в лабиринте — лайт

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Задана карта лабиринта: прямоугольное клетчатое поле из r строк и c столбцов. Каждая клетка либо свободна, либо занята стеной.

Робот начинает движение в свободной клетке, обозначенной на карте буквой «R». За один шаг робот может переместиться на одну из четырёх соседних клеток лабиринта, если эта клетка свободна. Выходить за пределы лабиринта нельзя.

Задача робота — за минимальное количество шагов оказаться в свободной клетке, обозначенной на карте буквой «G». Найдите это минимальное количество шагов или выясните, что пути не существует.

Формат входных данных

В первой строке записаны два целых числа r и c — количество строк и столбцов в карте лабиринта ($1 \leq r, c \leq 500$). Каждая из следующих r строк содержит c символов, и все вместе они задают карту лабиринта. Вот значения символов:

- «.» (точка, ASCII-код 46) — пустая клетка,
- «#» (решётка, ASCII-код 35) — стена,
- «R» (от слова «robot») — начальное положение,
- «G» (от слова «goal») — конечное положение.

Символы «R» и «G» встречаются на карте ровно по одному разу.

Формат выходных данных

В первой строке выведите целое число k — минимальное количество шагов, которое потребуется роботу для прохода из начального положения в конечное, или -1 , если это невозможно.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 5##.R .G#..	8
2 2 R# #G	-1

Задача В. Путь в лабиринте

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Задана карта лабиринта: прямоугольное клетчатое поле из r строк и c столбцов. Каждая клетка либо свободна, либо занята стеной.

Робот начинает движение в свободной клетке, обозначенной на карте буквой «R». За один шаг робот может переместиться на одну из четырёх соседних клеток лабиринта, если эта клетка свободна. Выходить за пределы лабиринта нельзя.

Задача робота — за минимальное количество шагов оказаться в свободной клетке, обозначенной на карте буквой «G». Найдите это минимальное количество шагов или выясните, что пути не существует.

Если путь существует, найдите также любую подходящую последовательность шагов.

Формат входных данных

В первой строке записаны два целых числа r и c — количество строк и столбцов в карте лабиринта ($1 \leq r, c \leq 500$). Каждая из следующих r строк содержит c символов, и все вместе они задают карту лабиринта. Вот значения символов:

- «.» (точка, ASCII-код 46) — пустая клетка,
- «#» (решётка, ASCII-код 35) — стена,
- «R» (от слова «robot») — начальное положение,
- «G» (от слова «goal») — конечное положение.

Символы «R» и «G» встречаются на карте ровно по одному разу.

Формат выходных данных

В первой строке выведите целое число k — минимальное количество шагов, которое потребуется роботу для прохода из начального положения в конечное, или -1 , если это невозможно.

Если проход возможен, во второй строке выведите инструкцию из k шагов для такого прохода. Если возможных инструкций несколько, выведите любую. Обозначения шагов для движения по карте лабиринта:

- «N» (север) — вверх,
- «W» (запад) — влево,
- «S» (юг) — вниз,
- «E» (восток) — вправо.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 5##.R .G#..	8 WNWWWSSE
2 2 R# #G	-1

Задача С. Два робота в лабиринте — лайт

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Задана карта лабиринта: прямоугольное клетчатое поле из r строк и c столбцов. Каждая клетка либо свободна, либо занята стеной.

Два робота начинают движение в свободных клетках, обозначенных на карте цифрами «1» и «2». За один шаг робот может переместиться на одну из четырёх соседних клеток лабиринта, если эта клетка свободна. Клетки за пределами лабиринта следует считать стенами.

Оба робота выполняют одну и ту же инструкцию, состоящую из следующих команд:

- «N» (север) — шаг вверх,
- «W» (запад) — шаг влево,
- «S» (юг) — шаг вниз,
- «E» (восток) — шаг вправо.

При выполнении каждой команды, если в нужной стороне от робота свободная клетка, он перемещается в эту клетку. Если же там стена или граница лабиринта, робот игнорирует команду и переходит к следующей.

Задача роботов — за минимальное количество шагов оказаться в одной и той же клетке. Найдите это минимальное количество шагов или выясните, что решения не существует.

Формат входных данных

В первой строке записаны два целых числа r и c — количество строк и столбцов в карте лабиринта ($1 \leq r, c \leq 30$). Каждая из следующих r строк содержит c символов, и все вместе они задают карту лабиринта. Вот значения символов:

- «.» (точка, ASCII-код 46) — пустая клетка,
- «#» (решётка, ASCII-код 35) — стена,
- «1» — начальное положение первого робота,
- «2» — начальное положение второго робота.

Цифры «1» и «2» встречаются на карте ровно по одному разу.

Формат выходных данных

В первой строке выведите целое число k — минимальное количество шагов, которое потребуется роботам, чтобы оказаться в одной клетке, или -1 , если это невозможно.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 5 ..#.. .2#1.	4
2 2 1# #2	-1

Задача D. Два робота в лабиринте

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Задана карта лабиринта: прямоугольное клетчатое поле из r строк и c столбцов. Каждая клетка либо свободна, либо занята стеной.

Два робота начинают движение в свободных клетках, обозначенных на карте цифрами «1» и «2». За один шаг робот может переместиться на одну из четырёх соседних клеток лабиринта, если эта клетка свободна. Клетки за пределами лабиринта следует считать стенами.

Оба робота выполняют одну и ту же инструкцию, состоящую из следующих команд:

- «N» (север) — шаг вверх,
- «W» (запад) — шаг влево,
- «S» (юг) — шаг вниз,
- «E» (восток) — шаг вправо.

При выполнении каждой команды, если в нужной стороне от робота свободная клетка, он перемещается в эту клетку. Если же там стена или граница лабиринта, робот игнорирует команду и переходит к следующей.

Задача роботов — за минимальное количество шагов оказаться в одной и той же клетке. Найдите это минимальное количество шагов или выясните, что решения не существует.

Если решение существует, найдите также любую подходящую инструкцию.

Формат входных данных

В первой строке записаны два целых числа r и c — количество строк и столбцов в карте лабиринта ($1 \leq r, c \leq 30$). Каждая из следующих r строк содержит c символов, и все вместе они задают карту лабиринта. Вот значения символов:

- «.» (точка, ASCII-код 46) — пустая клетка,
- «#» (решётка, ASCII-код 35) — стена,
- «1» — начальное положение первого робота,
- «2» — начальное положение второго робота.

Цифры «1» и «2» встречаются на карте ровно по одному разу.

Формат выходных данных

В первой строке выведите целое число k — минимальное количество шагов, которое потребуется роботам, чтобы оказаться в одной клетке, или -1 , если это невозможно.

Если задача разрешима, во второй строке выведите инструкцию из k шагов для такого прохода. Если возможных инструкций несколько, выведите любую.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 5 ..#.. .2#1.	4 SEEE
2 2 1# #2	-1

Задача Е. Полиция в лабиринте

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Задана карта лабиринта: прямоугольное клетчатое поле из r строк и c столбцов. Каждая клетка либо свободна, либо занята стеной.

В некоторых свободных клетках находятся полицейские. За один шаг полицейский может переместиться на одну из четырёх соседних клеток лабиринта, если эта клетка свободна. Выходить за пределы лабиринта полицейский не может.

Если обитатели лабиринта вызывают полицию в какую-то клетку, туда отправляется тот полицейский, который может добраться до этой клетки за минимальное количество шагов. Это количество шагов называется удалённостью клетки от полиции.

Для каждой клетки, до которой может добраться какой-нибудь полицейский, найдите её удалённость от полиции.

Формат входных данных

В первой строке записаны два целых числа r и c — количество строк и столбцов в карте лабиринта ($1 \leq r, c \leq 500$). Каждая из следующих r строк содержит c символов, и все вместе они задают карту лабиринта. Вот значения символов:

- «.» (точка, ASCII-код 46) — пустая клетка,
- «#» (решётка, ASCII-код 35) — стена,
- «P» (от слова «police») — полицейский.

Формат выходных данных

Выведите r строк, каждая из которых содержит c чисел — удалённость от полиции для всех клеток лабиринта. Разделяйте числа в строке пробелами. В клетках, куда полицейскому не добраться, выведите -1 .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 6P.P. .P.... P.....	3 2 2 1 0 1 2 1 2 1 0 1 1 0 1 2 1 2 0 1 2 3 2 3
3 5 .P#.P ##P.# ..#..	1 0 -1 1 0 -1 -1 0 1 -1 -1 -1 -1 2 3

Задача F. Динамит в лабиринте — лайт

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Задана карта лабиринта: прямоугольное клетчатое поле из r строк и c столбцов. Каждая клетка либо свободна, либо занята стеной.

Робот начинает движение в свободной клетке, обозначенной на карте буквой «R». За один шаг робот может переместиться на одну из четырёх соседних клеток лабиринта. При этом, если клетка была занята стеной, то робот тратит один заряд динамита, чтобы взорвать эту стену, после чего клетка становится свободной. Выходить за пределы лабиринта нельзя.

Задача робота — потратив минимальное количество зарядов динамита, оказаться в свободной клетке, обозначенной на карте буквой «G». Найдите это минимальное количество зарядов.

Формат входных данных

В первой строке записаны два целых числа r и c — количество строк и столбцов в карте лабиринта ($1 \leq r, c \leq 500$). Каждая из следующих r строк содержит c символов, и все вместе они задают карту лабиринта. Вот значения символов:

- «.» (точка, ASCII-код 46) — пустая клетка,
- «#» (решётка, ASCII-код 35) — стена,
- «R» (от слова «robot») — начальное положение,
- «G» (от слова «goal») — конечное положение.

Символы «R» и «G» встречаются на карте ровно по одному разу.

Формат выходных данных

В первой строке выведите целое число k — минимальное количество зарядов динамита, которое потребуется роботу для прохода из начального положения в конечное.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 5 ..#.. ##.#R G###.	2
2 2 R# #G	1

Задача G. Динамит в лабиринте

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Задана карта лабиринта: прямоугольное клетчатое поле из r строк и c столбцов. Каждая клетка либо свободна, либо занята стеной.

Робот начинает движение в свободной клетке, обозначенной на карте буквой «R». За один шаг робот может переместиться на одну из четырёх соседних клеток лабиринта. При этом, если клетка была занята стеной, то робот тратит один заряд динамита, чтобы взорвать эту стену, после чего клетка становится свободной. Выходить за пределы лабиринта нельзя.

Задача робота — потратив минимальное количество зарядов динамита, оказаться в свободной клетке, обозначенной на карте буквой «G». Найдите это минимальное количество зарядов.

Найдите также любую подходящую последовательность шагов.

Формат входных данных

В первой строке записаны два целых числа r и c — количество строк и столбцов в карте лабиринта ($1 \leq r, c \leq 500$). Каждая из следующих r строк содержит c символов, и все вместе они задают карту лабиринта. Вот значения символов:

- «.» (точка, ASCII-код 46) — пустая клетка,
- «#» (решётка, ASCII-код 35) — стена,
- «R» (от слова «robot») — начальное положение,
- «G» (от слова «goal») — конечное положение.

Символы «R» и «G» встречаются на карте ровно по одному разу.

Формат выходных данных

В первой строке выведите целое число k — минимальное количество зарядов динамита, которое потребуется роботу для прохода из начального положения в конечное.

Во второй строке выведите инструкцию для такого прохода. Если возможных инструкций несколько — выведите любую, в которой не больше 1 000 000 шагов. Обозначения шагов для движения по карте лабиринта:

- «N» (север) — вверх,
- «W» (запад) — влево,
- «S» (юг) — вниз,
- «E» (восток) — вправо.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 5 ..#.. ##.#R G###.	2 NWWWSS
2 2 R# #G	1 SNSE

Задача Н. Карта лабиринта

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Лабиринт — это прямоугольное клетчатое поле. Каждая клетка либо свободна, либо занята стеной.

Робот начинает движение в свободной клетке лабиринта и принимает команды. Каждая команда — перемещение в одну из четырёх сторон. Если соседняя клетка в этой стороне существует и свободна, робот перемещается туда, иначе — остаётся на месте. На каждую команду робот отвечает, получило ли перемещение.

Известно, что лабиринт связан, то есть из любой его свободной клетки робот может попасть в любую другую. Постройте и выведите полную карту лабиринта.

Протокол взаимодействия

Это интерактивная задача. В каждом тесте лабиринт и начальная клетка для робота зафиксированы заранее. Гарантируется, что в лабиринте не более 50 строк и не более 50 столбцов.

Решение выводит команды для робота и получает ответы на них. Каждая команда — буква, выведенная на отдельной строке.

Если команда — это буква «N» (север, вверх), «W» (запад, влево), «S» (юг, вниз) или «E» (восток, вправо), то робот пытается перейти на соседнюю клетку в указанном направлении. После этого решение получает строку «Yes», если перемещение удалось, и «No», если нет. Всего можно дать не более 20 000 таких команд.

Если команда — это буква «M» (карта), то в следующей строке следует вывести через пробел целые числа r и c — количество строк и столбцов в карте лабиринта. В каждой из следующих r строк выведите c символов: все вместе они должны задавать карту лабиринта. Выводите «.» (точку, ASCII-код 46) для обозначения пустых клеток и «#» (решётку, ASCII-код 35) на месте стен. После вывода карты сразу завершите работу программы.

В выведенной карте не должно быть лишних строк и столбцов, состоящих целиком из стен. Другими словами, хотя бы одна пустая клетка должна быть в первой строке, в последней строке, в первом столбце и в последнем столбце.

Помните, что после вывода каждой команды нужно очищать буфер вы-

вода (например, функцией «`fflush(stdout)`» в языке C, «`cout.flush()`» в C++ или «`sys.stdout.flush()`» в Python). В противном случае выведенные команды могут не дойти до проверяющей программы, и тогда решение получит вердикт «`Idleness Limit Exceeded`».

Пример

В примере слева — вывод программы участника (команды и карта), а справа — то, что подаётся ей на вход (ответы на них).

<i>стандартный вывод</i>	<i>стандартный ввод</i>
S	Yes
W	No
S	No
E	Yes
S	No
E	No
N	Yes
E	No
N	No
W	Yes
N	No
W	Yes
W	No
N	No
M	
2 3	
...	
#..	

Задача I. Непересекающиеся пути — лайт

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Задана карта лабиринта: прямоугольное клетчатое поле из r строк и c столбцов. Каждая клетка либо свободна, либо занята стеной.

Несколько улиток начинают движение в свободной клетке, обозначенной на карте буквой «R». За один шаг улитка может переместиться на одну из четырёх соседних клеток лабиринта, если эта клетка свободна. Выползть за пределы лабиринта нельзя.

Задача улиток — оказаться в свободной клетке, обозначенной на карте буквой «G». Но при этом нужно, чтобы каждую клетку лабиринта либо вообще не посетили улитки, либо один раз посетила одна из улиток. Исключение составляют начальная и конечная клетки: их может по одному разу посетить каждая улитка, а не только какая-то одна.

Сколько улиток могут по этим правилам добраться до конечной клетки? Найдите пути, позволяющие как можно большему количеству улиток это сделать. Количество шагов в найденных путях может быть любым.

Гарантируется, что начальная и конечная клетки не являются соседними.

Формат входных данных

В первой строке записаны два целых числа r и c — количество строк и столбцов в карте лабиринта ($1 \leq r, c \leq 50$). Каждая из следующих r строк содержит c символов, и все вместе они задают карту лабиринта. Вот значения символов:

- «.» (точка, ASCII-код 46) — пустая клетка,
- «#» (решётка, ASCII-код 35) — стена,
- «R» (от слова «robot») — начальное положение,
- «G» (от слова «goal») — конечное положение.

Символы «R» и «G» встречаются на карте ровно по одному разу и не находятся в соседних клетках.

Формат выходных данных

В первой строке выведите целое число k — количество найденных путей, оно должно быть максимальным возможным. В следующих k строках выведите сами пути. Каждый путь должен быть выведен как последовательность шагов, начинаться в начальной клетке и заканчиваться в конечной.

Все остальные клетки всех путей должны быть попарно различны и отличаться от начальной и конечной клеток. Количество шагов в выведенных путях может быть любым.

Обозначения шагов для движения по карте лабиринта:

- «N» (север) — вверх,
- «W» (запад) — влево,
- «S» (юг) — вниз,
- «E» (восток) — вправо.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 5 # # . R . G . . .	2 NWWWSE WSWW
2 2 R# #G	0

Задача J. Непересекающиеся пути

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Задана карта лабиринта: прямоугольное клетчатое поле из r строк и c столбцов. Каждая клетка либо свободна, либо занята стеной.

Несколько улиток начинают движение в свободной клетке, обозначенной на карте буквой «R». За один шаг улитка может переместиться на одну из четырёх соседних клеток лабиринта, если эта клетка свободна. Выползть за пределы лабиринта нельзя.

Задача улиток — оказаться в свободной клетке, обозначенной на карте буквой «G». Но при этом нужно, чтобы каждую клетку лабиринта либо вообще не посетили улитки, либо один раз посетила одна из улиток. Исключение составляют начальная и конечная клетки: их может по одному разу посетить каждая улитка, а не только какая-то одна.

Сколько улиток могут по этим правилам добраться до конечной клетки? Найдите пути, позволяющие как можно большему количеству улиток это сделать. При этом суммарное количество шагов в найденных путях должно быть наименьшим из возможных для такого количества улиток.

Гарантируется, что начальная и конечная клетки не являются соседними.

Формат входных данных

В первой строке записаны два целых числа r и c — количество строк и столбцов в карте лабиринта ($1 \leq r, c \leq 50$). Каждая из следующих r строк содержит c символов, и все вместе они задают карту лабиринта. Вот значения символов:

- «.» (точка, ASCII-код 46) — пустая клетка,
- «#» (решётка, ASCII-код 35) — стена,
- «R» (от слова «robot») — начальное положение,
- «G» (от слова «goal») — конечное положение.

Символы «R» и «G» встречаются на карте ровно по одному разу и не находятся в соседних клетках.

Формат выходных данных

В первой строке выведите целое число k — количество найденных путей, оно должно быть максимальным возможным. В следующих k строках выведите сами пути. Каждый путь должен быть выведен как последовательность шагов, начинаться в начальной клетке и заканчиваться в конечной.

Все остальные клетки всех путей должны быть попарно различны и отличаться от начальной и конечной клеток. Суммарное количество шагов в выведенных путях должно быть наименьшим из возможных для k улиток.

Обозначения шагов для движения по карте лабиринта:

- «N» (север) — вверх,
- «W» (запад) — влево,
- «S» (юг) — вниз,
- «E» (восток) — вправо.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 5 # # . R . G . . .	2 NWWWSS WSWW
2 2 R# #G	0