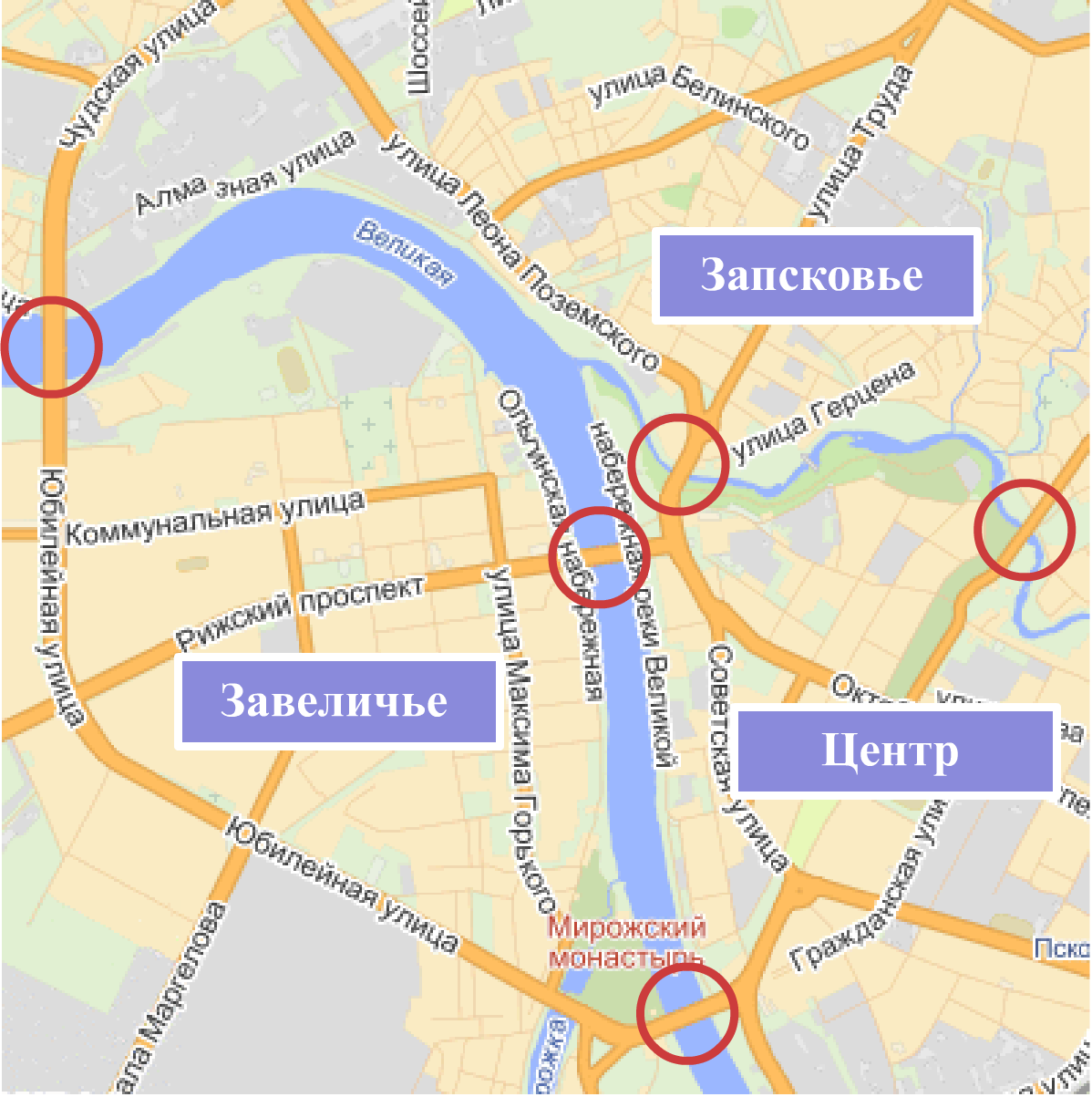
1. **Маршруты**

Ограничение по времени: **2 секунды**

Ограничение по памяти: **65 мегабайт**

В городе Пскове есть три больших района – Центр, Завеличье и Запсковье. Попасть из одного района в другой можно только по одному из мостов. Через реку, разделяющую Центр и Завеличье, имеется два моста, Завеличье и Запсковье соединяет один мост, Центр и Запсковье – два.



Пётр Васильевич живёт в одном районе, а работает в другом. Ему не нравится ходить на работу одной и той же дорогой и ему также не хочется проходить через один и тот же мост более одного раза. Сколько различных путей можно предложить Петру Васильевичу из района, где он живёт, в район, где он работает? Два пути считаются различными, если они содержат различную последовательность мостов. При подсчёте учитывайте пожелания Петра Васильевича, даже если ему для этого придётся побывать в одном районе, включая начальный и конечный, несколько раз.

Входные данные

На входе имеются две строки, содержащие по одной латинской букве A, B или C.

A – Центр, B – Завеличье, C – Запсковье.

Выходные данные

Единственное число – количество различных путей из района, указанного в первой строке, в район, указанный во второй строке.

Примеры

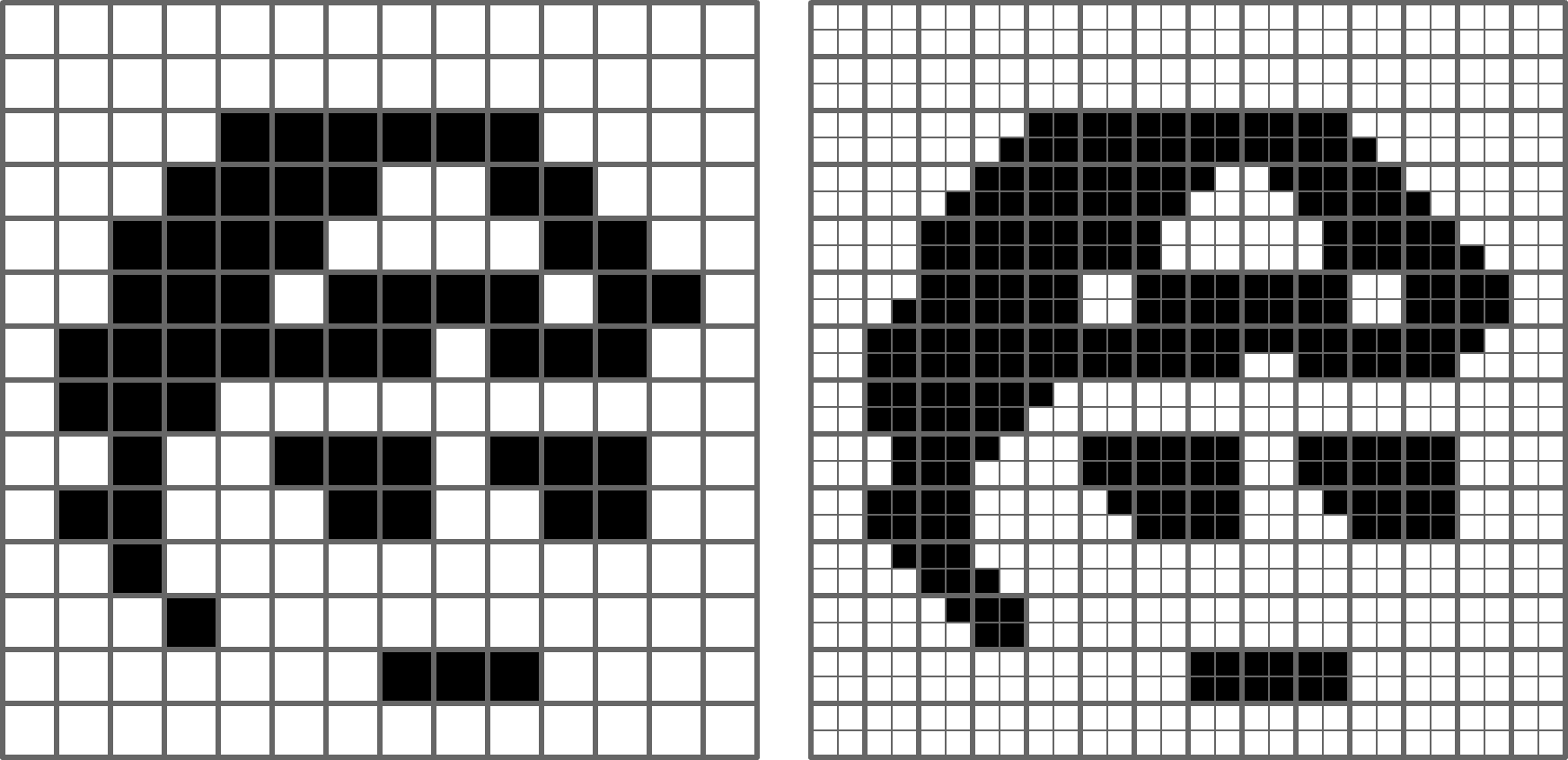
|  |  |
| --- | --- |
| Ввод | Вывод |
| A  B | 20 |
| A  C | 20 |

1. **Масштабирование**

Ограничение по времени: **2 секунды**

Ограничение по памяти: **65 мегабайт**

Одна из турфирм города желает получить увеличенные туристические карты Пскова. Эту задачу она поручила своему молодому сотруднику Николаю. Николай придумал, как можно двукратно увеличить уже имеющиеся у фирмы карты. Он заметил, что при обычном увеличении каждый чёрный пиксель исходного изображения заменяется на четыре черных пикселя увеличенного изображения. Но при этом карта получается зернистой. Николай предлагает и некоторые белые пиксели заменить чёрными согласно разработанным им правилам. Эти правила достаточно просты, и Николай уверен, что, внимательно рассмотрев пример, Вы и сами их поймёте. Напишите программу, выполняющую масштабирование изображений по правилам Николая. Гарантируется, что результат увеличения пикселя зависит только от цвета самого пикселя и соседних по стороне пикселей.



Входные данные

В первой строке два целых числа ***n*** (2 <= ***n*** <= 100) и m (2 <= ***m*** <= 100) - высота и ширина изображения. В следующих n строках по m символов задают изображение. '.' - белый пиксель, '#' - чёрный пиксель.

Выходные данные

Выведите масштабированное изображение в том же формате, что и на входе, но без указания размеров.

Примеры

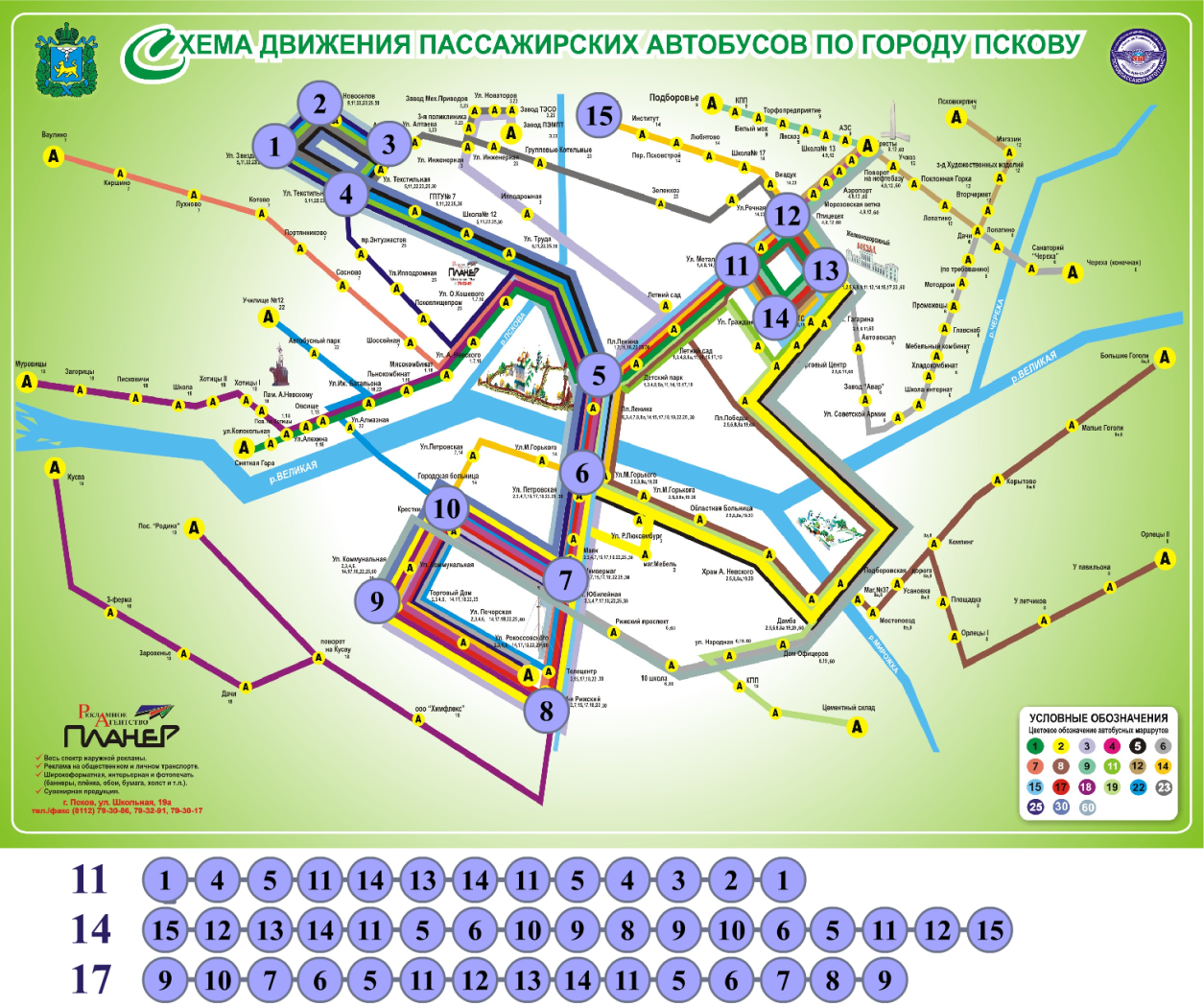
|  |  |
| --- | --- |
| Ввод | Вывод |
| 14 14  ..............  ..............  ....######....  ...####..##...  ..####....##..  ..###.####.##.  .#######.###..  .###..........  ..#..###.###..  .##...##..##..  ..#...........  ...#..........  .......###....  .............. | ............................  ............................  ............................  ............................  ........############........  .......##############.......  ......#########..#####......  .....#########....#####.....  ....#########......#####....  ....#########......######...  ....######..########..####..  ...#######..########..####..  ..#######################...  ..##############..######....  ..#######...................  ..######....................  ...####...######..######....  ...###....######..######....  ..####.....#####...#####....  ..####......####....####....  ...###......................  ....###.....................  .....###....................  ......##....................  ..............######........  ..............######........  ............................  ............................ |

1. **Автобусы**

Ограничение по времени: **2 секунды**

Ограничение по памяти: **65 мегабайт**

В будущем городские автобусы будут управляться роботами. А сейчас в качестве эксперимента решено смоделировать движение трёх таких автобусов-роботов на трёх разных маршрутах (11, 14, 17). Маршрут автобуса с помощью контрольных пунктов разбит на несколько участков, по каждому из которых автобус движется с постоянной скоростью. Перечень контрольных пунктов каждого из трёх маршрутов вы можете увидеть на рисунке. Известно также время, за которое автобус конкретного маршрута преодолевает конкретный участок. Автобусы выходят со своих конечных остановок одновременно и выполняют один рейс по своему маршруту. Вам нужно проверить, не произойдёт ли на каком-нибудь участке столкновение двух автобусов различных маршрутов. Автобусы на участке (включая контрольные точки) могут столкнуться, если они движутся по этому участку в одном направлении и в некоторый момент времени оказываются в одной и той же точке участка.



Входные данные

В первой строке записаны времена переезда между контрольными пунктами для "11" маршрута. Время переезда между контрольными пунктами – целое число от 1 до 5000 включительно. В следующих двух строках записаны времена переездов для "14" и "17" маршрутов в аналогичном формате.

Выходные данные

Выведите QUITE, если не произойдёт ни одного столкновения. Выведите CRASH в противном случае.

Пример

|  |  |
| --- | --- |
| Ввод | Вывод |
| 5 5 5 5 5 5 5 5 5 5 5 5  5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5  5 5 5 5 5 5 5 5 5 5 5 5 5 5 | QUITE |
| 5 5 5 5 5 5 5 5 5 5 5 5  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  5 5 5 5 5 5 5 5 5 5 5 5 5 5 | CRASH |

1. **Квест**

Ограничение по времени: **2 секунды**

Ограничение по памяти: **65 мегабайт**

Квест – приключенческая игра, цель которой – добраться до финиша. Игрокам сообщается первое место. По прибытии на место игроки находят подсказку к следующему месту или заканчивают игру. Какое расстояние преодолеют игроки?

Входные данные

В первой строке единственное число ***n*** (1 <= ***n*** <= 100) – количество пунктов, участвующих в квесте.

В каждой из следующих n строк описываются пункты в следующем формате:

***name*** ***x y next***

где ***name*** – название пункта, ***x*** и ***y*** – координаты пункта, ***next*** – название следующего пункта, которое обязательно встречается в какой-нибудь строке, либо слово finish, которое означает, что квест заканчивается в данном пункте. Названия пунктов – строки, состоящие из латинских буков, длиной не более 20 символов. Координаты пунктов целые числа  
от – 10 000 до + 10 000 включительно. Координаты всех пунктов различны. Старт всегда находится в начале координат.

Выходные данные

Выведите расстояние, которое преодолеют игроки от старта до финиша с точностью не ниже 10–2. Расстояние между двумя пунктами равно длине отрезка, соединяющего две точки.

Пример

|  |  |
| --- | --- |
| Ввод | Вывод |
| 3  PskovSU 0 0 Imperial  Maximus 5 3 finish  Imperial -4 3 Maximus | 14.000000 |
| 3  Cafe 0 0 finish  Pool 3 11 Cafee  Square 8 7 Pool | 0.000000 |

Пояснение

Длина отрезка AB [(x1, y1), (x2, y2)] вычисляется по формуле .

1. **Улица**

Ограничение по времени: **2 секунды**

Ограничение по памяти: **65 мегабайт**

При подготовке к проведению переписи населения волонтёру было дано задание перечислить номера существующих домов на некоторой улице. Волонтёр не может внести в список несуществующий дом, но может по невнимательности пропустить некоторые номера домов. Для надёжности было решено пустить по улице двух отдельных волонтёров, после чего сравнить их списки домов. Используя списки двух волонтёров, перечислите номера домов, имеющихся на улице. Номера домов и в списках волонтёров, и в вашем списке могут идти в произвольном порядке, но не должны повторяться.

Входные данные

В первой строке записано количество номеров домов в списке первого волонтёра. Во второй строке записаны номера домов через пробел. Список второго волонтёра записан в следующих двух строках в том же формате.

Волонтёры непоследовательны и их списки могут содержать номера в любом порядке.

Гарантируется, что во входном файле все числа целые в диапазоне от 1 до 100.

Выходные данные

В первой строке выведите количество номеров домов в результирующем списке. Во второй строке выведите сами номера домов через пробел в любом порядке.

Примеры

|  |  |
| --- | --- |
| Ввод | Вывод |
| 9  1 3 5 7 8 10 6 4 2  9  1 2 3 4 5 6 7 8 9 | 10  1 2 3 4 5 10 9 8 6 7 |
| 2  3 4  2  4 7 | 3  7 4 3 |

1. **Таблица умножения**

Ограничение по времени: **2 секунды**

Ограничение по памяти: **65 мегабайт**

Выведите таблицу умножения для чисел до ***n***.

Входные данные

На входе единственное целое число ***n*** – от 2 до 9.

Выходные данные

Следуйте формату вывода указанному в примерах. Выравнивать таблицу не требуется.

Примеры

|  |  |
| --- | --- |
| Ввод | Вывод |
| 2 | \* 2  2 4 |
| 4 | \* 2 3 4  2 4 6 8  3 6 9 12  4 8 12 16 |