import itertools

def Read(name): # Создаём функцию, которая будет считывать данные из файла

    with open(name, 'r') as file: # Открываем файл, для чтения

        f = file.readlines() # считываем из файла строки с данными

        N, K, L = map(int, f[0].strip().split()) # Считываем данные из первой строки файла (идекс - 0)

        G = [f[i].strip().split() for i in range(1, K+1)] # Считываем данные из второй и последующих строк файла

    return N, L, G, K # Возвращаем данные: размер доски, количество фигур, которые нужно поставить и координаты уже стоящих фигур на доске, количество уже стоящих фигур

def main\_figure(G): # Создаем функцию для главной фигуры, которая уже стоит на доске

    U = [] # Создаем список клеток: где стоит фигура, куда она может ходить

    E = [] # Координаты, где стоит фигура

    for i in G:

        U.append((int(i[0]), int(i[1])))

        U.append((int(i[0])-1, int(i[1])))

        U.append((int(i[0])+1, int(i[1])))

        U.append((int(i[0])+1, int(i[1])-2))

        U.append((int(i[0])-1, int(i[1])-2))

        U.append((int(i[0])-1, int(i[1])+2))

        U.append((int(i[0])+1, int(i[1])+2))

        E.append((int(i[0]), int(i[1])))

    return U, E

def another(U): # Создаём функцию координат фигуры и её ходов

    U1 = [] # Записываем отрицательные координаты(то есть те координаты, которе являются лишними, клетки, которых в принципе нет)

    U2 = [] # Записываем все координтаы фигуры

    for i in U: # Проходимся по списку координат фигуры

        for j in i: # Проходимся по (х, у) каждой координаты

            if j < 0: # Если одна из координта отрицательна, то записываем пару координат в U1

                U1.append(i)

            else: # Добавляем координаты в U2

                U2.append(i)

    U3 = set(U2) - set(U1) # Создаём множество правильных координта(кооринаты, которые есть на доске)

    return U3

def step(N): # Создаем функцию для всех возможных клеток шахматной доски

    H = [(x, y) for x in range(N) for y in range(N)] # Добавляем в список все координаты клеток доски

    return set(H)

def Right(H, U): # Создаём функцию с верными координатами, то есть с клетками, куда можно поставить фигуры

    H1 = [(i[0], i[1]) for i in (set(H) - set(U))] # (х, у) для списка(все клетки минус клетки главной фигуры)

    return H1

def koordinates(name1, L, H1, E): # Создаем множество координат, которые будем записывать в файл вывода

    B = [] # Создаем список для хранения комбинаций расположения фигур на доске

    N1 = itertools.combinations(H1, L) # Создаем все возможные комбинации расстановки фигур

    for i in N1: # проходимся по каждой комбинации

        B.append(i) # Добавляем комбинацию в список

    with open(name1, 'w') as ko: # Открываем файл

        if len(B) > 0: # Если список не пустое, то

            for i in B: # проходимся по каждой комбинации

                while len(i) > 0: # если длина комбинации больше 0, то

                    for p in E: # По очереди записываем координаты клетки, где стоит фигура

                        ko.write(str(p))

                    for j in range(len(i)): # Продимся по каждому координате, входящей в комбинцию (для каждой комбинации новый цикл)

                        ko.write(str(i[j])) # записываем координату

                    ko.write('\n') # когда все координаты комбинации записываем, переходим на следующую строку для записи новой комбинации координат

                    break

        else: # В остальных случаях, то есть, если список пустой

            ko.write('No solution') # Если не нашлось подходящихклеток, то выводим "Нет решения"

def board(N, U3, E): # Создаём функцию, которая будет выводить доску с фигурами, уже стоящими на ней(и их ходами)

    d = [[(x, y) for x in range(N)] for y in range(N)] # создаем список координат клеток доски

    for i in reversed(d): # проходимся по кажтой строке в обратном порядке (то есто 19 -> 0)

        h = [] # создаём пустой список, куда будем запоминать символы клеток

        for j in i: # проходимс по каждой координате строки

            if j in E: # если координата - это координата главной фигуры, то обозначаем эту клетку #

                h.append('#') # добавляем символ в список

            elif j in U3: # если координата - это координата хода главной фигуры, то обозначаем эту клетку \*

                h.append('\*') # добавляем символ в список

            else: # в остальных случаях добавляем в список 0

                h.append('0')

        h = (str(h).ljust(20)) # превращаем список в строку и выравниваем доску по левому края

        print(\*[i for i in h if i in '0\*#']) # распаковываем каждую строку списка

def main(): # Создаём самую главную функцию, для того чтобы работать со всеми предыдущими функциями

    N, L, G, K = Read('1.txt') # Размер поля, количество фигур, которые нужно поставить, координаты фигуры, уже стоящей на шахматной доске

    U, E = main\_figure(G) # Список координат главной фигуры(работает с координатами, считанными из файла)

    H = step(N) # Все координаты доски (работает с размер доски, считанным из файла)

    H1 = Right(H, U) # Только подходящие координаты для расстановки фигур (работает с уже полученными множествами)

    U3 = another(U) # Клетки, которые есть на доске

    board(N, U3, E) # Выводит шахматную доску с фигурами (как она выглядит в самом начале)

    koordinates('2.txt', L, H1, E) # Записываем координаты в файл (работает с финальным списком координат и с количеством фигур(считанным из файла), которые нужно поставить на поле, координаты клетки,где стоит фигура)

if \_\_name\_\_ == '\_\_main\_\_': # Приводим в действие самую главную функцию

    main()