Нужна программа на языке python.   
  
Есть ограничивающий контур, который задаётся массивом точек. Точек может быть любое количество. Например, [(105, 470), (78.5, 470), (72, 444), (50, 296), (37.5, 226), (30, 174), (30, 132), (60, 220), (94, 336)].  
  
Внутри ограничивающего контура есть ломаные. Каждая из которых соответствует целому числу – точка забора. Сами ломаные задаются следующей структурой: точка забора, точка начала, массив точек перелома, точка конца. Ломаная строится последовательным соединением точек из заданной структуры. Например:   
lines = [

{‘collection\_point’: 120, 'start': (61.5, 372), 'points': [(79.5, 427)], 'end': (89, 470)},

{` collection\_point’: 30, 'start': (30, 146), 'points': [(80, 318)], 'end': (99, 403)}

]

Тут для первой ломанной мы ставим точку 61.5, 372 и соединяем с 79.5, 427, далее точку 79.5, 427 соединяем с 89, 470.  
  
Нужно построить ограничивающий контур, а также ломаные внутри контура. И написать код для решения следующей задачи. На ввод даётся целое число – entrance\_collection\_point. Нужно построить новую ломаную следующим образом. Выбираем ломаную из lines прямую, относительно которой будем строить новую ломаную. Это делается следующим обарзом, берём entrance\_collection\_point и сравниваем со всеми collection\_point из массива lines, нам нужна ломаная с min|collection\_point- entrance\_collection\_point|. Теперь у нас есть ломаная относительно которой мы будем строить новую ломаную, назовём её found\_line.

Когда мы определились с прямой found\_line мы должны посчитать расстояние, на которое будем переносить точки найденной прямой. Это расстояние равно dist = abs(collection\_point- entrance\_collection\_point). Далее мы берём points у found\_line и получаем из них новые точки следующим обарзом: если entrance\_collection\_point > collection\_point от нашей found\_line, то новые точки будут получены из start, points и end путём прибавления к каждой точке dist. А если entrance\_collection\_point < collection\_point то наоборот, мы из start, points, end вычитаем dist.  
  
В итоге мы получаем new\_line, у которой есть start, points, end. Осталось сделать так, чтобы эта прямая не выходила за контур.   
  
Для этого мы поступаем следующим оразом: new\_line содержит точку start и первую точку в массиве points, по этим двум точкам строится прямая и происходит поиск точек пересечения с ограничивающим контуром. Таких точек будет две, нам нужна та из них назовём её find\_point, у которой координата ближе по расстоянию к точке start, в точку start у new\_line присваивается find\_point.  
  
Теперь нам нужно найти новую точку end у new\_line. Для этого мы берём в new\_line точку end и последнюю точку из массива points в new\_line, по ним мы строим прямую и находим точки пересечения с ограничивающим контуром. Таких точек тоже должно быть две, берётся та из них, которая по расстоянию ближе к точке end, назовём её find\_end\_point. В точку end у new\_line записывается find\_end\_point.  
  
Теперь у нас есть в new\_line все точки, мы наносим эту прямую на график.