Как сдавать задачи?

Для каждой задачи рекомендуется сделать свое решение в VS или в другой IDE. После выполнения задачи, ее следует выложить в свой репозиторий Github или в любой другой сервис удаленных репозиториев (GitLab, BitBucket и т.д.) и прислать ссылку на него в переписку в hh или на рабочую почту **aproskurnin@dentalbot.ru**.

В случае успешного выполнения технического задания, вы будете приглашены на финальное очное собеседование (возможно собеседование в Zoom).

Общие требования:

Язык: C++

Использование ООП, STL и особенностей языка приветствуется.

Задача 1

Условие:

Есть два манипулятора. Их инструменты находятся в точках **O1** и **O2** в декартовой системе координат **(X, Y)**. У каждого манипулятора есть свой радиус эффективного действия **R1** и **R2**.
Есть целевая точка **P1.**

Задача:

Определите, каким манипулятором оптимальнее дойти до точки P1.
Если оба манипулятора не могут достать, то вывести сообщение: “OUT OF RANGE”.
Если точка доступна одному манипулятору – использовать его.

Задача 2

Условие:

Движение манипуляторов это, конечно, хорошо, но обычно с помощью кода они двигаются по определенным координатам, для сборки того или иного компонента.

Даны два манипулятора, которые описываются стартовыми центрами **O1** и **O2** и радиусом их действия **R1** и **R2** в декартовой системе координат. Радиус действия не изменяется в течении работы программы. Также дан массив точек, где лежат детали P1…Pn.

Задача:

Выведете две строки: какие точки обойдет каждый манипулятор.
Замечание: Важно помнить, что до детали должен дойти “оптимальный” манипулятор на текущей итерации процесса. Оптимальность можно считать по длине пути до точки. Радиус не учитывается. Пример ниже.

Пример:

Есть массив точек:
{1, 3}, {2, 1.41}, {0.2, -7} ,{-5, -1} , {0, 9}

Манипуляторы на старте:
M1: ({0, 0} , 4)
M2: ({2, 1}, 3)

Итерация	1	2	3	4	5
Манипулятор 1			{0.2, -7}		
Манипулятор 2	{1, 3}	{2, 1.41}		{-5, -1}	{0, 9}

Задача 3

Условие:

Есть GCODE Данного формата X(value);Y(value);Z(value).
Он используется для передачи команд микроконтроллеру, который в свою очередь управляет устройством, например манипулятором. Подробнее про GCODE можно прочитать тут.

Обычно GCODE выполняется построчно, поэтому если в скрипте оказалась ошибка, то только на ошибочной строке он остановится.

Входные данные:

GCODE

Задача:

Выведете конечные координаты устройства после выполнения комманд GCODE.
P.S. При написании кода обратите внимание на различные форматы GCODE команд.

Примеры:

10 20 0 X10;Y20;Z30; X40;Y-20;Z20; X-10;Y-20;Z-10;	50 0 40
10 20 0 X10;Y20;Z30; X40;Y-20;<Z20; X-10;>20;Z-10;	20 40 30 ERROR SCRIPT