

# Robo PACE Solutions

This document contains the solutions for the Robo PACE quizzes.

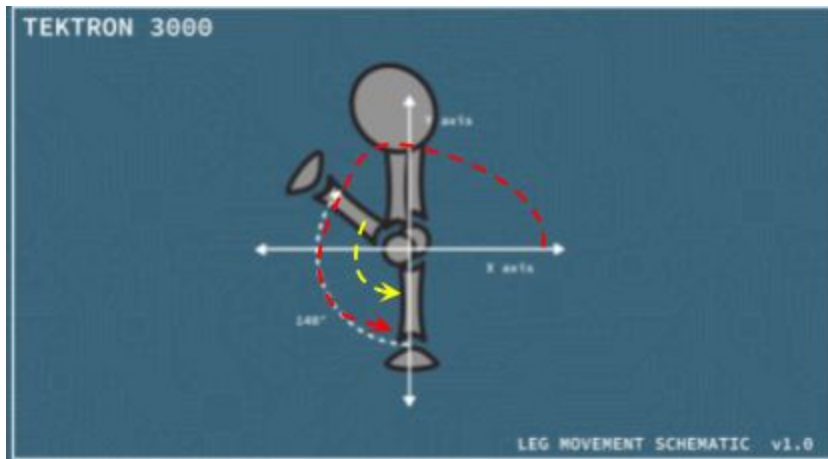
<b>Challenge: Range of Motion</b>	<b>1</b>
<b>Challenge: Physics</b>	<b>3</b>
<b>Challenge: Order of Tasks</b>	<b>4</b>
<b>Challenge: Rotational Matrices</b>	<b>5</b>
<b>Challenge: Unix/Linux Commands</b>	<b>7</b>
<b>Challenge: Planning</b>	<b>7</b>

# Challenge: Range of Motion

For this challenge we need to decide which of the given answers is a possible (x,y) coordinate for the robot's ankle.

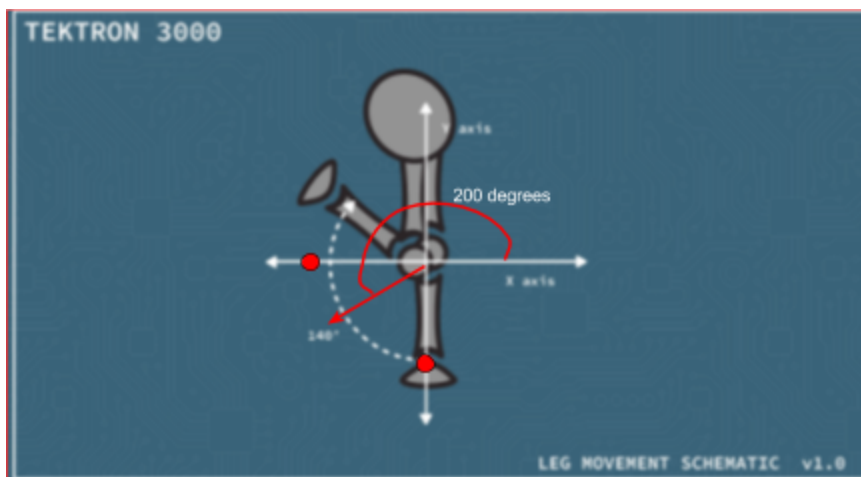
In the graphic below:

- The yellow arrow is the range of motion for the leg 140 degrees)
- The red arrow is the angle as measured from the traditional positive x axis.

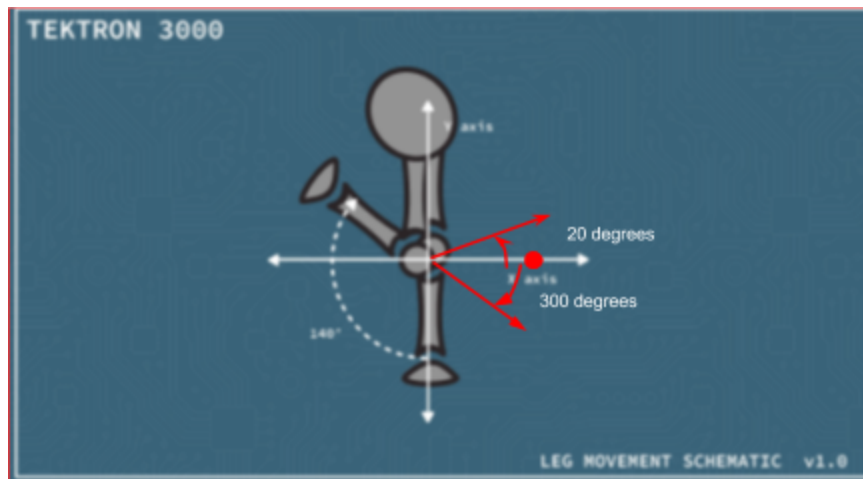


We can see all possible (x,y) coordinates for the ankle will be in two quadrants (-x,y) and (-x,-y). So we just need to look for answers that fit this criteria.

The correct answers are plotted below. We can see the correct answers fall within the range of the valid range of motion of the leg.



The incorrect answers are plotted below. We can see the answers do not fall within the valid range of motion for the ankle.



# Challenge: Physics

This is one solution to the programming quiz.

This is a sample program for utils.py

#Write a python function for the following declarations:

import re

#do not alter the printLocation function

def printLocation(location):

print "Location = (%i, %i)" %(location[0],location[1])

def moveForwardX(distance, startLocation):

x = startLocation[0] + distance

y = startLocation[1]

new = (x,y)

return new

def moveForwardY(distance, startLocation):

x = startLocation[0]

y = startLocation[1] + distance

new = (x,y)

return new

def calcDistance(distanceX, distanceY):

return ((distanceX)\*\*2 + (distanceY)\*\*2)\*\*0.5

def robotVelocity(distance, time):

velocity=distance/time

return distance/time

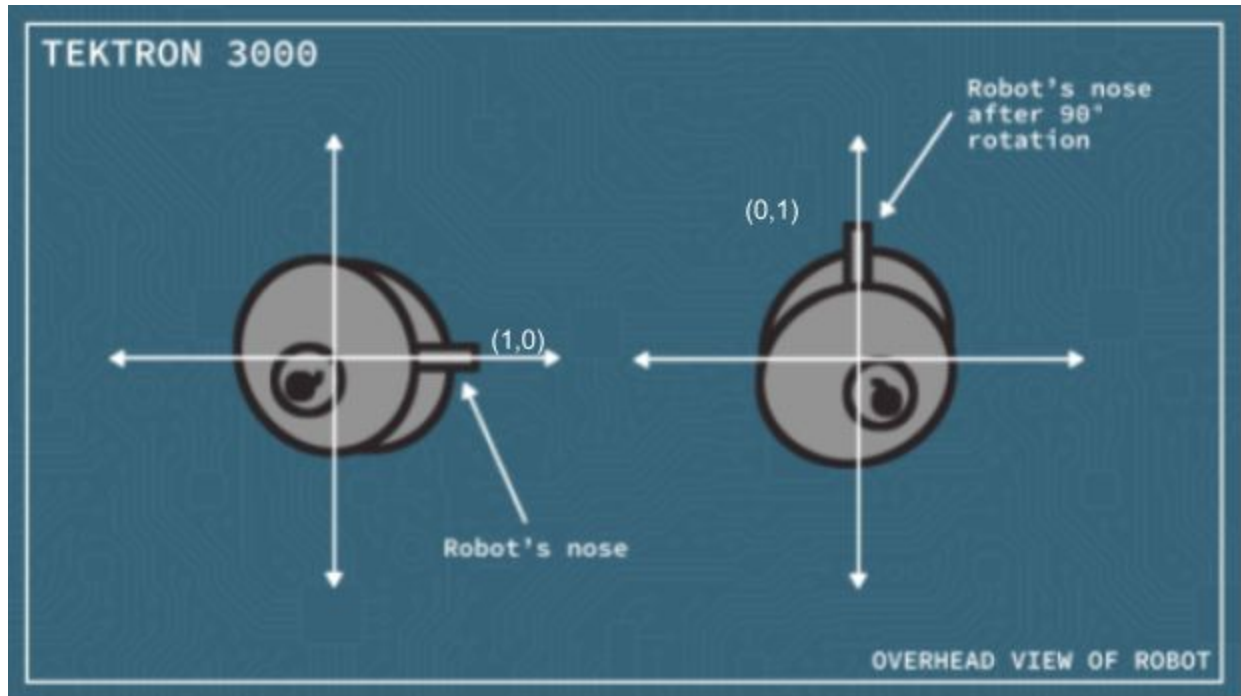
## Challenge: Order of Tasks

TASK NUMBER	TASK
Task #1	Scan the room
Task #2	Identify exit sign
Task #3	Plan the route to the exit sign
Task #4	Move toward exit

# Challenge: Rotational Matrices

Question 1/3

The two positions have (x,y) coordinates as shown:



Question 2/3

We know after answering the first question in this challenge, that a vector (1,0) on the x-axis rotated 90 degrees will result in a vector (0,1) on the y-axis.

Multiplying the matrix by the vector:

$$0 = 1 * M[0,0] + 0 * M[1,0] \rightarrow 0 = 1 * 0 + 0 * M[1,0] \rightarrow 0 = 1 * \cos(90) + 0$$

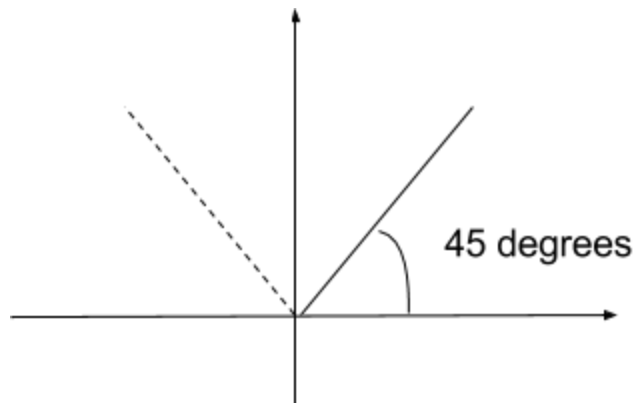
$$1 = 1 * M[0,1] + 0 * M[1,1] \rightarrow 1 = 1 * 1 + 0 * M[1,1] \rightarrow 1 = 1 * \sin(90) + 0$$

This reduces the number of possible right answers to 1.... Selection 'A'.

Let's try another example to make convince ourselves this is the correct answer.

Begin with a point along a line at 45 degrees from the x axis.  $(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$

Let's rotate it 90 degrees. The coordinates will then be  $(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$



Substitute in the each solution option and we see only A will satisfy the equations.

$$-\frac{\sqrt{2}}{2} = \frac{\sqrt{2}}{2} * M[0,0] + \frac{\sqrt{2}}{2} * M[0,1] \rightarrow -\frac{\sqrt{2}}{2} = \frac{\sqrt{2}}{2} * 0 + \frac{\sqrt{2}}{2} * -1 \rightarrow -\frac{\sqrt{2}}{2} = \frac{\sqrt{2}}{2} \cos(90) + \frac{\sqrt{2}}{2} (-\sin 90)$$

$$\frac{\sqrt{2}}{2} = \frac{\sqrt{2}}{2} * M[1,0] + \frac{\sqrt{2}}{2} * M[1,1] \rightarrow \frac{\sqrt{2}}{2} = \frac{\sqrt{2}}{2} * 1 + \frac{\sqrt{2}}{2} * 0 \rightarrow \frac{\sqrt{2}}{2} = \frac{\sqrt{2}}{2} \sin(90) + \frac{\sqrt{2}}{2} \cos(90)$$

Question3/3

Using the same substitution method, we see that answer 'D' is the correct option.

## Challenge: Unix/Linux Commands

To search for words in a file:

```
grep "search key" fileName  
grep "nodes" file.txt
```

To create a directory and change its permissions and list the directory contents with permissions:

```
mkdir Tektron3000
```

```
chmod 700 ./Tektron3000
```

```
ls -al
```

# Challenge: Planning

A sample solution of the Robot class:

#Write a python function for the following declarations:

#Note: the location tuple contains (x-coordinate, y-coordinate, orientation)

#orientation is N,S,E,W

```
class Robot:
    def __init__(self, sLocation):
        self.locationX = sLocation[0]
        self.locationY = sLocation[1]
        self.orientation = sLocation[2]

    def moveForward(self,sLocation,distance):
        x = sLocation[0]
        y = sLocation[1]
        if(sLocation[2] == 'N' or sLocation[2] == 'S'):
            y = sLocation[1] + distance
        else:
            x = sLocation[0] + distance
        newLocation = (x,y, sLocation[2])
        return newLocation

    def rotate(self,slocation, orientation):
        newLocation = (slocation[0],slocation[1], orientation)
        return newLocation
```