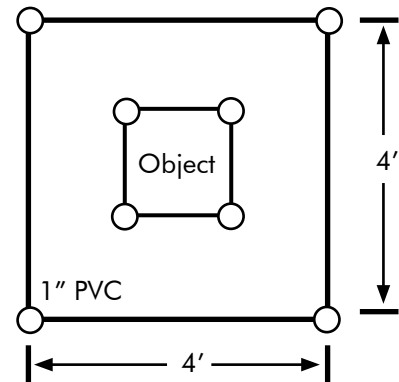# Optimizing Code

Challenge Description

The sample program below is made up of the same behaviors, used over and over. Write a program that uses structures such as functions and loops to minimize the number of lines of code necessary to perform the same functionality as the Sample Code below. To verify that you haven't changed the actual behavior of the robot, download and run both your program and the sample program on your robot. Note that you can and may need to modify the values to have the robot successfully move within the course.

Object

4'

1" PVC

4'

Sample Program

```
5  task main()
6  {
7      motor[leftMotor] = 63;
8      motor[rightMotor] = 63;
9      wait1Msec(2000);
10
11     motor[leftMotor] = -63;
12     motor[rightMotor] = 63;
13     wait1Msec(350);
14
15     motor[leftMotor] = 63;
16     motor[rightMotor] = 63;
17     wait1Msec(2000);
18
19     motor[leftMotor] = -63;
20     motor[rightMotor] = 63;
21     wait1Msec(350);
22
23     motor[leftMotor] = 63;
24     motor[rightMotor] = 63;
25     wait1Msec(2000);
26
27     motor[leftMotor] = -63;
28     motor[rightMotor] = 63;
29     wait1Msec(350);
30
31     motor[leftMotor] = 63;
32     motor[rightMotor] = 63;
33     wait1Msec(2000);
34
35     motor[leftMotor] = -63;
36     motor[rightMotor] = 63;
37     wait1Msec(350);
38
39     motor[leftMotor] = 63;
40     motor[rightMotor] = 63;
41     wait1Msec(2000);
42
43     motor[leftMotor] = -63;
44     motor[rightMotor] = 63;
45     wait1Msec(350);
46
47     motor[leftMotor] = 63;
48     motor[rightMotor] = 63;
49     wait1Msec(2000);
50
51     motor[leftMotor] = -63;
52     motor[rightMotor] = 63;
53     wait1Msec(350);
54
55     motor[leftMotor] = 63;
56     motor[rightMotor] = 63;
57     wait1Msec(2000);
58
59     motor[leftMotor] = -63;
60     motor[rightMotor] = 63;
61     wait1Msec(350);
62
63     motor[leftMotor] = -63;
64     motor[rightMotor] = 63;
65     wait1Msec(2000);
66
67     motor[leftMotor] = -63;
68     motor[rightMotor] = 63;
69     wait1Msec(350);
70  }
```