

## Robotics Practice

1. Given the following, use the provided subroutines to make your robot drive a SINGLE time as shown in the diagram. It should be two tiles wide in all directions.

Please note:

- The robot never drives backwards
- Do not directly control your motors (no 'PULSOUT')
- Do not use the PAUSE command
- Assume you are writing only the main code, no startup or subroutines
- Your code should be as efficient as possible
- Remember to STATE ALL ASSUMPTIONS

Available subroutines:

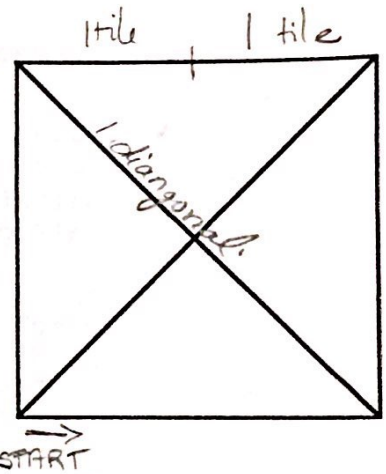
- FWD (Single pulse forward)
- SPIN\_LEFT\_45 (45° turn)
- SPIN\_LEFT\_90 (90° turn)
- SPIN\_LEFT\_135 (135° turn)

Variables declared:

- counter1 VAR Word
- counter2 VAR Word

```

FOR counter 1 = 1 to 4
  FOR counter 2 = 1 to 200 ' 2 tiles
    GO SUB FWD
  NEXT
  GOSUB SPIN-LEFT-135
  FOR counter 2 = 1 to 300 ' 1 diagonal
    GOSUB FWD
  NEXT
  GOSUB SPIN-LEFT-135
NEXT
    
```



assume :

- 2 tiles wide (200)
- 1 diagonal is 300 (actual 283)

## Robotics Practice

1. Given the following, use the provided subroutines to make your robot drive a SINGLE time as shown in the diagram. It should be two tiles wide in all directions.

Please note:

- The robot never drives backwards
- Do not directly control your motors (no 'PULSOUT')
- Do not use the PAUSE command
- Assume you are writing only the main code, no startup or subroutines
- Your code should be as efficient as possible
- Remember to STATE ALL ASSUMPTIONS

Available subroutines:

- FWD (Single pulse forward)
- SPIN\_LEFT\_45 (45° turn)
- SPIN\_LEFT\_90 (90° turn)
- SPIN\_LEFT\_135 (135° turn)

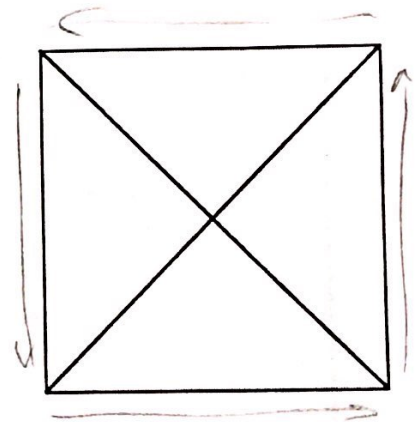
Variables declared:

- counter1 VAR Word
- counter2 VAR Word

```

FOR counter1 = 1 TO 4
  FOR counter2 = 1 TO 200 '2 tiles
    GOSUB FWD
  NEXT
  GOSUB SPIN_LEFT_90
NEXT
GOSUB SPIN_LEFT_45
FOR counter1 = 1 TO 300 '1 diagonal
  GOSUB FWD
NEXT
GOSUB SPIN_LEFT_135
FOR counter1 = 1 TO 200 '2 tiles
  GOSUB FWD
NEXT
GOSUB SPIN_LEFT_135
FOR counter1 = 1 TO 300 '1 diagonal
  GOSUB FWD
NEXT

```



```

FOR 1 TO 4
  ✓ FWD 200
  ✓ LEFT_90
  NEXT
  ✓ LEFT_135
  ✓ FWD_300
  ✓ LEFT_135
  ✓ FWD_200
  ✓ LEFT_135
  ✓ FWD_300

```

2. In the box on the right, sketch the path the following code would produce:

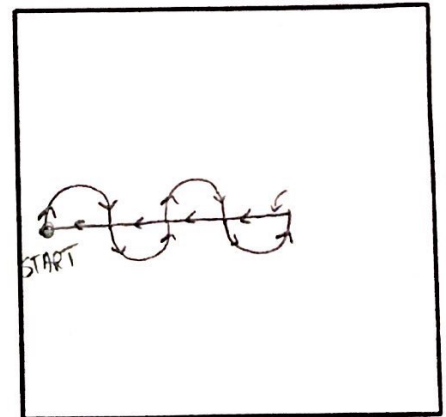
(Assume logical behavior for

subroutines: ex. RIGHT\_90 = 90 degree turn to the right)

```

counter VAR Word
flip VAR Byte

flip = 0
FOR counter = 1 TO 4
  IF flip = 0 THEN
    GOSUB curveRightSemiCircle 'diameter of 1 tile
    flip = 1
  ELSE
    GOSUB curveLeftSemiCircle 'diameter of 1 tile
    flip = 0
  ENDIF
NEXT
GOSUB spinLeft_90
FOR counter = 1 TO 200 'assume enough for 4 tiles
  GOSUB forward
NEXT
  
```



3. Examine the following code. Circle all errors.

```

' -----[ Variables ]-----
speedLeft    VAR Bite Byte
speedRight   VAR Bite
goalLeft     VAR Bite
goalRight    VAR Bite

' -----[ Main Code ]-----
speedLeft = 0
speedRight = 0
goalRight = 100
goalLeft = 100

DO
  GOSUB changeSpeed adjustSpeed
  GOSUB drive
NEXT LOOP

' -----[ Subroutines ]-----
adjustSpeed:
  IF FOR speedLeft =< goalLeft THEN
    speedLeft = speedLeft + 1
  ELSEIF speedLeft > goalLeft THEN
    speedLeft = speedLeft - 1
  ENDIF
  IF speedRight =< goalRight THEN
    speedRight = speedRight + 1
  ELSEIF speedRight > goalRight THEN
    speedRight = speedRight - 1
  ENDIF
END RETURN

drive:
  PULSOUT speedLeft, 12
  PULSOUT speedRight, 13
  PAUSE 20
RETURN
  
```

For this question, this is the whole code. If this were pasted into the editor, what problems would it find?

If any, treat any repeated error as a single error. Can you find all of them?

This is a really silly program. Don't worry about why it does what it does; just find the problems.