

Practice Test

1. Name each kind of turn and give the required code to make the robot perform it

a) Name: _____

```
PULSOUT motorR, _____ 'assuming:
PULSOUT motorL, _____ 'assuming:
PAUSE 20
```

b) Name: _____

```
PULSOUT motorR, _____ 'assuming:
PULSOUT motorL, _____ 'assuming:
PAUSE 20
```

c) Name: _____

```
PULSOUT motorR, _____ 'assuming:
PULSOUT motorL, _____ 'assuming:
PAUSE 20
```

2. Given the following, what is the final value stored in the 'mysteryNum' variable?

mysterNum = _____

```
mysteryNum VAR Word
mysteryNum = 25

IF (mysteryNum < 25) THEN
  GOSUB subA
ELSEIF (mysteryNum >= 25) THEN
  GOSUB subB
ELSE
  GOSUB subC
ENDIF

END

subA:
  mysteryNum = 25
RETURN

subB:
  mysteryNum = 100
RETURN

subC:
  mysteryNum = 42
RETURN
```

3. Given the following, use the provided subroutines to make your robot drive continuously in a “plus” sign as shown in the diagram. It should take up two tiles in each direction.

Please note:

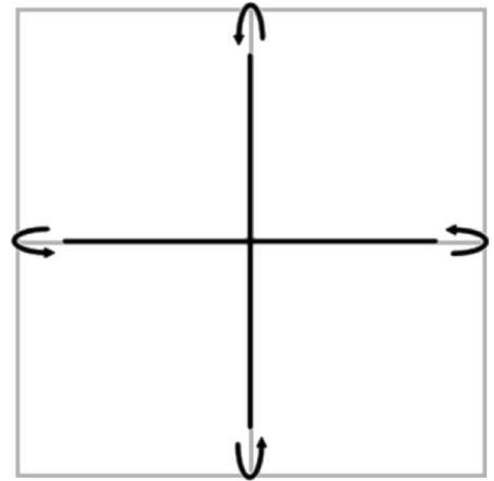
- The robot never drives backwards
- Do not directly control your motors (no ‘PULSOUT’)
- Do not use the PAUSE command
- Assume you are writing only the main code, no startup or subroutines
- Your code should be as efficient as possible

Available subroutines:

- FWD
- SPIN_L
- SPIN_R
- PIVOT_L
- PIVOT_R

Variables declared:

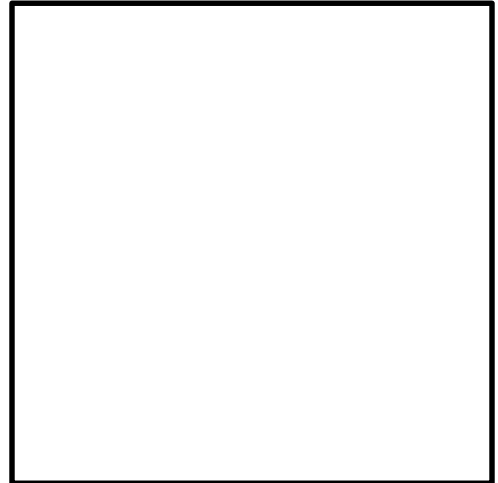
- loopCounter1 VAR Word
- loopCounter2 VAR Word



4. In the box on the right, sketch the path the following code would cause:
(assume logical behavior for subroutines: ex. RIGHT_90 = 90 degree turn to the right)

```
'---Assume normal definitions above here
switch VAR Word
switch = 0
DO
  IF (switch=0) THEN
    GOSUB RIGHT_90
    switch = 1
  ELSE
    GOSUB LEFT_90
    switch = 0
  ENDIF

  GOSUB FORWARD_1_TILE
LOOP
```



5. Examine the following code. Circle all errors.

```
' {$STAMP BS2}
' {$PBASIC 2.5}

myNumber VARIABLE Word

GOSUB STARTUP
myNumber = 1234
DO
  IF (myNumber = 42)
    GOSUB HITCH_HIKERS
  ELSE IF (myNumber <> 1234) THEN
    GOSUB EASY_NUMBER
  ELSE
    GOSUB HOW_DID_I_GET_HERE
  END
LOOP

STARTUP:
  FREAKOUT piezo, 250, 5000
  FREAKOUT piezo, 250, 4000
  FREAKOUT piezo, 500, 6000
RETURN

HITCH_HIKER:
  DO
    PULSOUT 12, 900
    PULSOUT 13, 900
    PAUSE 20
  LOOP

EASY_NUMBER:
  DO
    PULSOUT 12, 600
    PULSOUT 13, 600
    PAUSE 20
  LOOP

HOW_DID_I_GET_HERE:
  DO
    PULSOUT 12, 750
    PULSOUT 13, 750
    PAUSE 20
  LOOP
```

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 seven?

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