



#### Unit 6: Rover's Coordinates

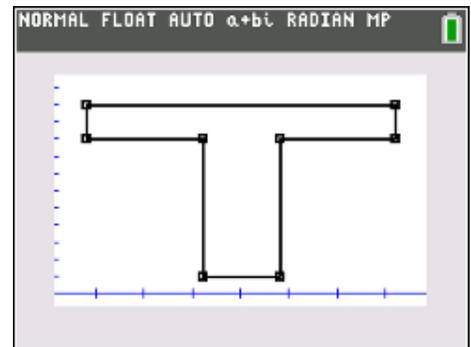
#### Skill Builder 3: Make the Shape

In this lesson, you will write a program to make a pre-designed two-dimensional shape. You will program with lists and use a loop to get Rover to draw (or just drive) the shape on paper.

#### Objectives:

- Create lists in the calculator's Stat Editor
- Use `recall_list` in Python
- Use a `for` loop to process elements in the lists
- Use a `'pause'` statement to control processing

This project requires you to make two lists that represent the x- and y-coordinates of the vertices of a shape of your own design. In this lesson, we will use the design of the block letter **T** (for Texas!) shown to the right. Your goal is to have the Rover make this design using a marker (or just follow the route if no marker is available).



1. Before writing the program, enter your shape's coordinates into two lists in the TI-84 Stat Editor: Quit Python, press `[2nd] [stat]` and select **Edit...** from the menu. All the coordinates of the 'T' shape are given in the image to the right. Note that the names of the two lists are  $L_1$  and  $L_2$ . These names are important when you write your Python program.

You can test your values by setting up a Scatter Plot (`[2nd] [y=]`) and an appropriate viewing window (`[zoom] ZoomStat`). You should see the plot shown in the previous step.

*Why do we start and end with the same coordinate pair (4, 1)? What would it draw without the last 4, 1?*

L1	L2
4	1
6	1
6	9
9	9
9	11
1	11
1	9
4	9
4	1
-----	-----



# 10 Minutes of Code – Python

## TI-84 PLUS CE PYTHON WITH THE TI-INNOVATOR™ ROVER

- Now begin a new Python Rover Coding project.

Your first two *new* statements will get the lists from the TI-84 variables and store them in two Python variables.

From **[math] ti\_system...** menu select the statement:

```
var=recall_list("name")
```

*Reminder: the blue text is not included in your code.*

You need *two* of these statements, so just copy and paste the statement using the **<Tools>** menu (or get the statement from the menu again).

- To complete the statements, insert variables to the left of the = signs and, inside the quotes, type 1 and 2 for the lists L<sub>1</sub> and L<sub>2</sub> in the calculator.

We are using the variables **xs** and **ys** for the two Python lists.

This special TI-developed function copies data from the calculator side of the TI-84 into the Python environment. Just the numbers of the lists are needed.

- You are now ready to program the Rover's route. Remember that Rover begins at the point (0,0). Your first point might not be the origin, so have Rover move to the first point before pausing to insert the marker. The first point is ( xs[0], ys[0] ), so use the statement:

```
rv.to_xy(xs[0], ys[0])
```

**[ ]** are found in several places:

- On the keypad (see **[2<sup>nd</sup>] [x]** (multiply) and **[2<sup>nd</sup>] [-]** (subtract).
- On the **[list]** menu (**[2<sup>nd</sup>] [stat]**)
- On the Python menu **<Fns...> List**
- On the **<a A #>** Character Map

```
EDITOR: RVCOORDC
PROGRAM LINE 0007
# Rover
from time import *
from ti_system import *
import ti_rover as rv

=recall_list("")
=recall_list("")
```

```
EDITOR: RVCOORDC
PROGRAM LINE 0008
# Rover
from time import *
from ti_system import *
import ti_rover as rv

xs=recall_list("1")
ys=recall_list("2")
-
```

```
EDITOR: RVCOORDC
PROGRAM LINE 0004
# Rover
from time import *
from ti_system import *
#import ti_rover as rv

xs=recall_list("1")
ys=recall_list("2")
rv.to_xy(xs[0],ys[0])
```



- Add a statement that pauses processing while you insert a marker (if you have one) *after* Rover reached the starting point:

**disp\_wait()**

```

EDITOR: RVCOORDC
PROGRAM LINE 0010
# Rover
from time import *
from ti_system import *
#import ti_rover as r

xs=recall_list("1")
ys=recall_list("2")
rv.to_xy(xs[0],ys[0])
disp_wait()

```

- Make a **for** loop to drive to the rest of the points:

**for i in range(1, len(xs)):**



**len(xs)** is the length (size) of the list **xs**. If the length is 12, then the loop ends with the value  $i = 11$ , the last element of the list.

Notes: use the **for i in range(start, stop)** structure on <Fns...> Ctl

**len()** is found on <Fns...> List

```

EDITOR: RVCOORDC
PROGRAM LINE 0011
# Rover
from time import *
from ti_system import *
#import ti_rover as r

xs=recall_list("1")
ys=recall_list("2")
rv.to_xy(xs[0],ys[0])
disp_wait()
for i in range(1,len(xs)):
    rv.to_xy(xs[i],ys[i])

```

- Challenge:** Use the **tiplotlib** tools to also plot the shape's coordinates on the screen *in sync with Rover*. Use **plt.auto\_window(xs, ys)** to set up the window for your points. At the end, you can use the function **plt.plot(xs,ys,"mark")** to draw a *connected* scatter plot as shown here.

