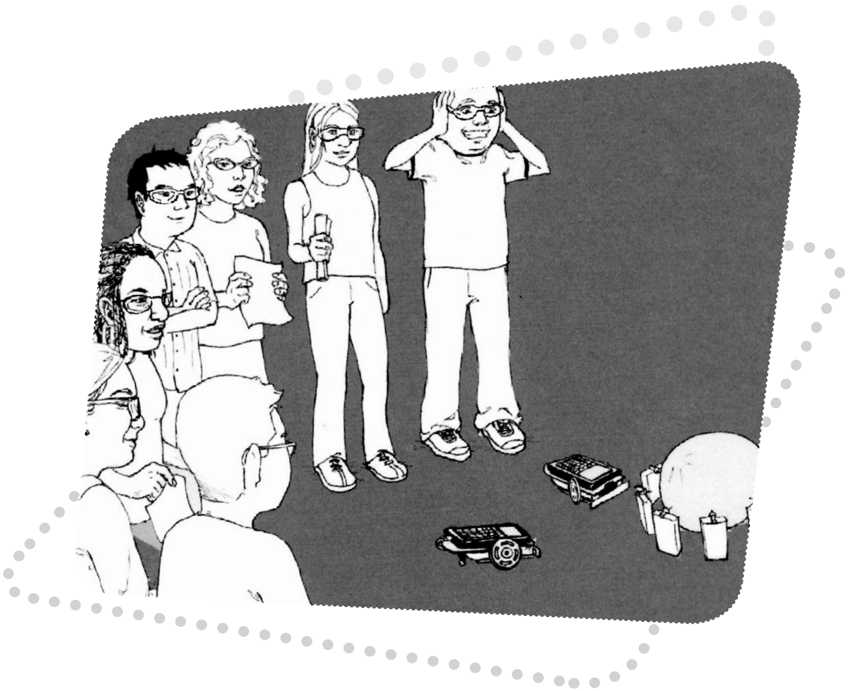
Name:			Date:
MISSION	2	Graph and Predict	*Materials*

Your second mission, should you decide to take it (and you know you will), is to come as close to crashing your robot into an object as possible, without actually hitting the object.

You need:

- ❏ 1 Norland Calculator Robot
- ❏ 1 Graphing Calculator
- ❏ 1 Meter Stick
- ❏ Graph Paper
- ❏ Safety Goggles



Name:	Date:
MISSION 2 Graph and Predict	*Instructions*

Use a meter stick with your robot and the **GO** program from **Mission 1** to obtain data for the table below:

Table 1

Time (In seconds)	**Distance** (In centimeters)
	50
	100
	150
	200
	250

Graph the data as points on graph paper with TIME on the horizontal or x-axis and DISTANCE on the vertical or y-axis. Draw the best-fitting line that most closely follows the pattern shown by your data points.

How Good Is Your Graph?

Write a simple program (see PROGRAMMING INSTRUCTIONS if needed) for your robot on a graphing calculator name your program **MISSION2**:

```
PROGRAM: MISSION2
    : randInt (1,10)->X

    : Disp X
    : Pause
    : X*100->T
    : Send ({122,T})
    : Get (R)
    : Stop
```

Name:			Date:
MISSION	2	Graph and Predict	*Instructions*

This program will randomly pick a number from 1 to 10. This number represents the time in seconds the robot will be instructed to travel forward. The program will pause while you use your graph to predict the distance the robot will travel. Record your prediction in the table below, then press $\boxed{\text{ENTER}}$ on the graphing device and measure the actual distance. Record your degree of error.

Table 2

Time (In seconds)	**Prediction** (In centimeters)	**Actual**	**Error**

Name:			Date:
MISSION	**2**	**Graph and Predict**	***Advanced***

Using your graph will let you predict how far the robot will travel forward for a given time. What if a time is given that exceeds the limitations of your graph?

Is there a more accurate way to predict or calculate the expected distance?

Find the slope of your best-fitting line on the graph. What does the slope represent?

Write an equation for your line using the slope-intercept form y=mx+b.
What does b equal?

Graph this equation on your calculator. Press $\boxed{\text{TRACE}}$ and confirm your values from table one. For any given Y value (distance) there will be a corresponding X value (time).

Rewrite your slope-intercept equation above, substituting d for y, r for m, and t for x. Does it look familiar? This equation can be used to predict the time needed for the robot to travel any given distance.

Name:			Date:
MISSION	2	Graph and Predict	*Challenge*

The mission is to instruct your robot to move as close as possible to a teacher designated object without actually hitting it. You may measure from the starting point to the object and then you must predict the time that your robot will need to complete the task. Using the **GO** program from **Mission 1** change code line 1 from ":Send ({222})" TO ":Send ({122,xxx})" where xxx represents the time in centiseconds you want your robot to travel forward, i.e. 850=8.5 seconds. (See EDITING INSTRUCTIONS on page 14 if needed.)

Imagine that you are sending a $125 million satellite to Mars and not doing a trial and error exercise. You have only one shot. Make sure your estimates are accurate and that you have accounted for all variables.

Accuracy of Prediction Grading Scale:

Within 0 to ≤10 cm	**A**
>10 to ≤20 cm	**B**
>20 to ≤30 cm	**C**
>30 cm	**Try again**

Name:			Date:
MISSION	2	Graph and Predict	*Results*

1. In centimeters, how close did you get?

2. What could you do to improve your results?

3. What other designated object would be interesting to use in this mission besides the one given by your teacher? (Perhaps one that would show a definite reaction if a robot hit it.)

4. How did you predict the travel time needed for the robot?

Name:	Date:

MISSION 2 Graph and Predict

Programming Instructions

Turn on your graphing calculator. Press PRGM, then use the arrow to highlight **NEW**. Press ENTER, then spell out **MISSION2** by pressing the appropriate keys. (Press ALPHA to switch from letters back to numbers for the 2.) Press ENTER and you're ready to enter the first command for the program.

Line 1: Press MATH, then use the arrow to highlight **PRB**. Use the arrow to scroll down to **5: randInt(**. Press ENTER. Type in 1,10 and). Press STO▶. Press the X,T,Θ,*n* button. Press ENTER. The first line should appear as:
:randInt (1, 10) -> X

Line 2: Is blank

Line 3: Press PRGM, then use the arrow to highlight **I/O**. Use the arrow to scroll down to **3: Disp**. Press ENTER. Press the X,T,Θ,*n* button. Press ENTER. The third line should appear as:
:Disp X

Line 4: Press PRGM and **CTL** will be highlighted. Use the arrow to scroll down to **: Pause**. Press ENTER twice. The fourth line should appear as:
:Pause

Line 5: Press X,T,Θ,*n*. Press × and then type in 100. Press STO▶. Press ALPHA, then press [T]. Press ENTER. The fifth line should appear as:
X*100->T

Line 6: Press PRGM, then use the arrow to highlight **I/O**. Use the arrow to scroll down to **B: Send(**. Press ENTER. Press 2nd and then press [{]. Type in 122, then press ,. Press ALPHA, then press [T]. Close by pressing 2nd, the [}] button, and then). Press ENTER. The seventh line should appear as:
:Send ({122,T})

Line 7: Press PRGM, then use the arrow to highlight **I/O**. Use the arrow to scroll down to **A: Get (**. Press ENTER. Press ALPHA, then press [R]. Press) then ENTER. The second line should appear as:
:Get (R)

Line 8: Press PRGM and **CTL** will be highlighted. Use the arrow to scroll down to **F: Stop**. Press ENTER. The fourth line should appear as:
:Stop
Press 2nd, then [QUIT].

To run the program, attach the calculator to your robot and connect link cable. Make sure the robot and calculator are both switched on. Press PRGM and use the arrow to scroll down to **: MISSION2**. Press ENTER. Press ENTER again and the program will randomly pick a number from 1 to 15. This number represents the time in seconds the robot will be instructed to travel forward. The program will pause while you predict the distance the robot will travel. Place the robot on the floor, then press ENTER and robot will travel forward for the displayed number of seconds.

Editing Instructions:

Press PRGM, then use the arrow to highlight **EDIT**. Use the arrow to scroll down to **:GO**. Press ENTER. Change 222 to 122 then press ,. Enter the number of centiseconds you want the robot to run. Close the braces and parentheses by pressing 2nd, the [}] button, and then). The new first line (if you want the robot to run 8.5 seconds) should appear as:
:Send ({122,850})
Press 2nd, then [QUIT]. Follow instructions above to run the program; after the second ENTER the robot will move forward.

The graph of TIME verses DISTANCE for this exercise should be a straight line that can be modeled by a linear equation. Students can make predictions by reading from the graph. Algebra 1 students or higher can use the slope formula and the slope–intercept model ($y=mx+b$) to find an equation for the best-fitting line. The y-intercept point (b) is zero. The slope of the line is determined by the speed or rate of the robot. From their slope–intercept equation, students can discover the formula for distance, $d=rt$.

The object that student robots come close to, but never touch, could be many things. For example: a wall, a house of cards, a small board with a tack that can fall and pop a balloon, or a line of dominos. The board and balloon challenge is exciting and works well. Students come to a testing area with only their robots and graphs. They are tested individually and given a randomly picked distance to start from in front of the board. Grades are given according to the scale on page 11. Whoever gets closest to the board for the class period can pop the balloon at the end. Participants should wear safety goggles.

Another approach is to simulate a Mars Rover doing planetary exploration. The robot has to go to the edge of a crater rim (tabletop) and peer down with a mock up bumper mounted camera to take pictures. The bumper must hang over the edge without the robot falling.