Gravity Car Calendar (90 min block)

Week	Mon	Tues	Wed	Thur	Fri	
1	- Entry Doc - K/NtK/NS - Lab - Gravity Car - 8s video - Gravity Car Data Sheet	- Do Now: Average Rate of Change/Piecewise - Finish Gravity Car Data Sheet - Start Planning Document	- Do Now: What's My Function? - Finish Planning Document - Program cars - ch code	- Do Now: Graphing Data - The Twist: - 10s video - student data sheet	- Do Now: Fit a parabola to data? - Loops Workshop: Loops Ch - Program cars - Take data - Prepare for Code Review - Rubric for Code Review	
2	- Code Review - Group Quiz	- Individual Test				
3						

Gravity Car Graph Challenge:

Dear Students,

Now that we've had fun with my racecar, I wanted to try something that doesn't need batteries. This gravity car accelerates down the ramp through the power of gravity. Does your Linkbot car do the same thing? Can you make the Linkbot's motion mirror the motion of my gravity car? Your challenge is to have the graph of your Linkbot look exactly the same as the graph of my gravity car at one second intervals.

Your proof of this will not just be a graph, but well documented code. Well documented means every line or item that isn't self explanatory should have a comment that helps the reader understand what you did. Often the reader is you at a later date trying to reuse the code, the comments will help future you save time. And isn't that what life is, but trying to make sure future you is set up for awesomeness?

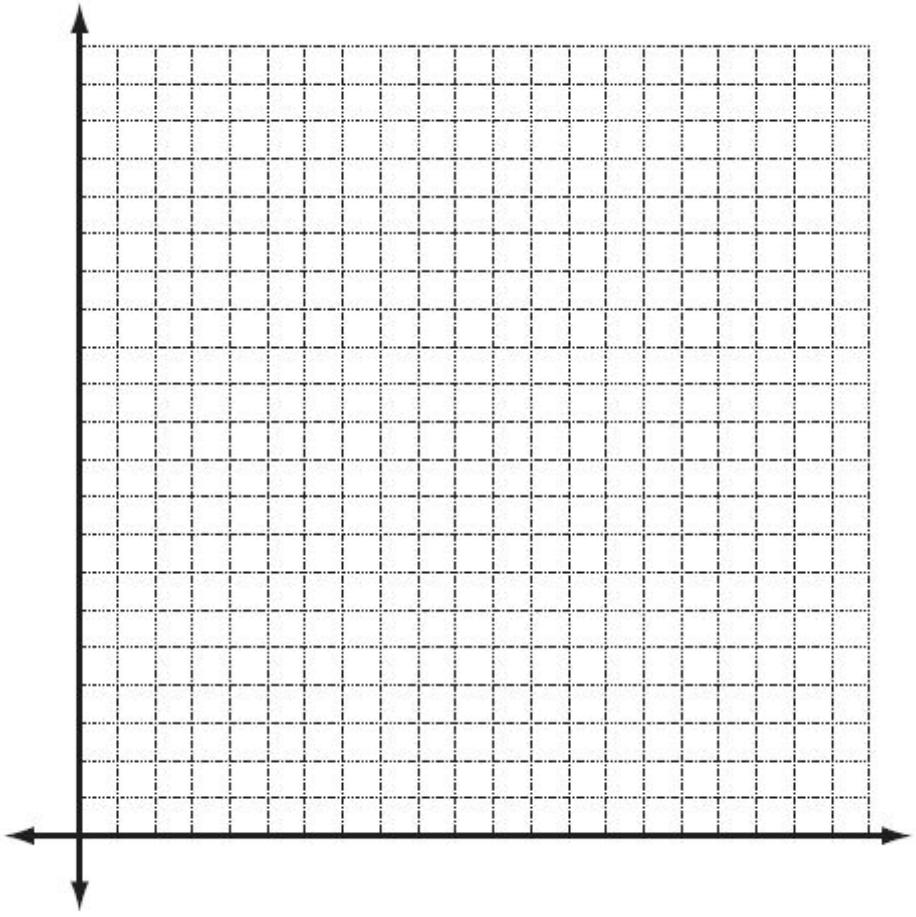
Sincerely,
Your Teacher

Gravity Car Data

IV: _____ DV:_____

Data:

Graph:



1. a. Does the car move at a constant speed down the ramp? Give evidence from the table to back up your answer.

1. b. Describe the motion of the car.

2. How does your graph show the car's motion?

3. What is the average speed between 0 and 1 second?

Add your answer to the table below and fill in the rest of the table.

Seconds	Average Speed
0-1	
1-2	
2-3	
3-4	
4-5	
5-6	
6-7	
7-8	

4. How do your answers in the table in #3 support what you said in #1 and #2?

Warm-up: Average Rate of Change/Piecewise

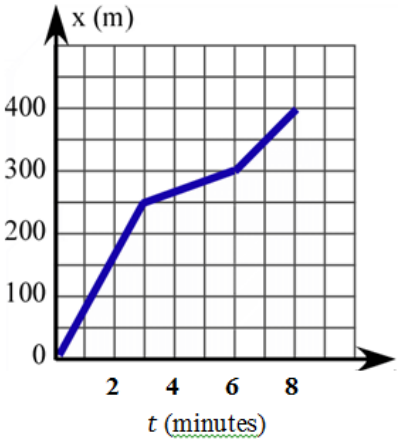
6. The graph below shows Glenn's distance from home as he rode his bicycle to school, which is just down his street. His next-door neighbor Pablo, who lives 100 m closer to the school, leaves his house at the same time as Glenn. He walks at a constant velocity, and they both arrive at school at the same time.

   a. Graph a linear function that represents Pablo's distance from Glenn's home as a function of time.

   b. Estimate when the two boys pass each other.

   c. Write piecewise-linear functions to represent each boy's distance and use them to verify your answer to part (b).

   d. What is the averagae rate of change for the two boys?



Linkbot Planning:

You will learn several new commands today to help you make the Linkbot's motion match the gravity car's motion. Everything after the two forward slashes is a comment that the computer will ignore, but the reader may find helpful. The following program makes the robot go at speed1 for one second, then speed2 for one second.

```
#include <linkbot.h> //loads linkbot module

CLinkbotI robot; //assigns the name robot to the robot

robot.connect(); //connects the robot

robot.setJointSpeeds(speed1,NaN,speed1); //changes the robot's speed:
degrees/second
robot.moveNB(x,NaN,-x);//it goes x degrees altogether
sleep(1); //sleeps for 1 second
robot.setJointSpeeds(speed2,NaN,speed2); //you should add your own comments
sleep(1);
```

How might this be useful for matching the motion of the gravity car?

Use the file gravityCar.ch to get your started. Remember to comment each line with things like, *what the line does* and *why you are doing that*.

Once you think your Linkbot is right, run your program, take data, then graph the data on the next page.

Does your graph match the gravity car graph? What needs to be fixed?

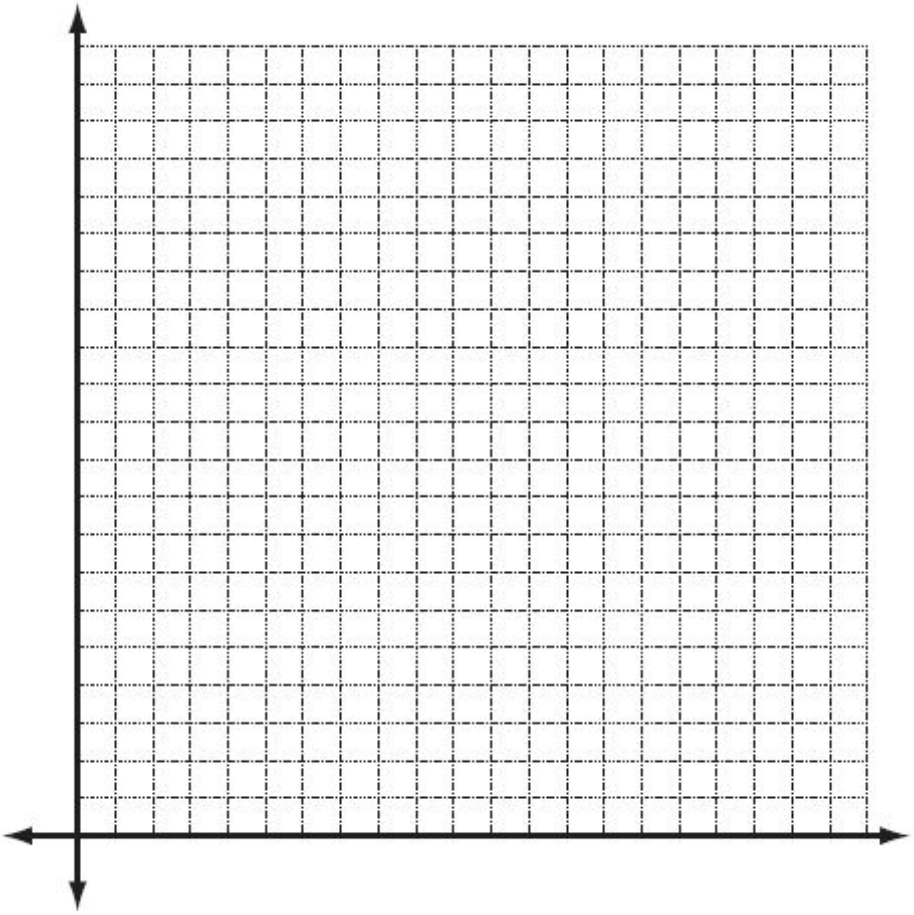
Are your graphs linear like the previous challenges or is it something else?

Linkbot Data

IV: _____ DV:_____

Data:

Graph:

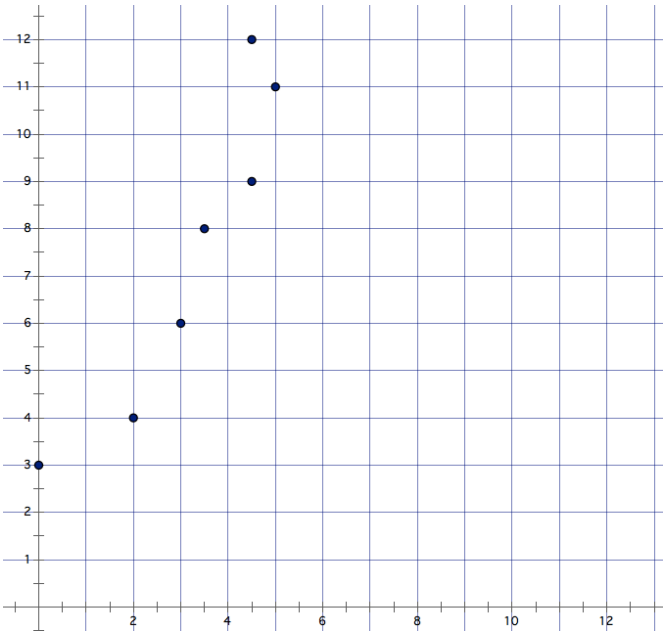


Set

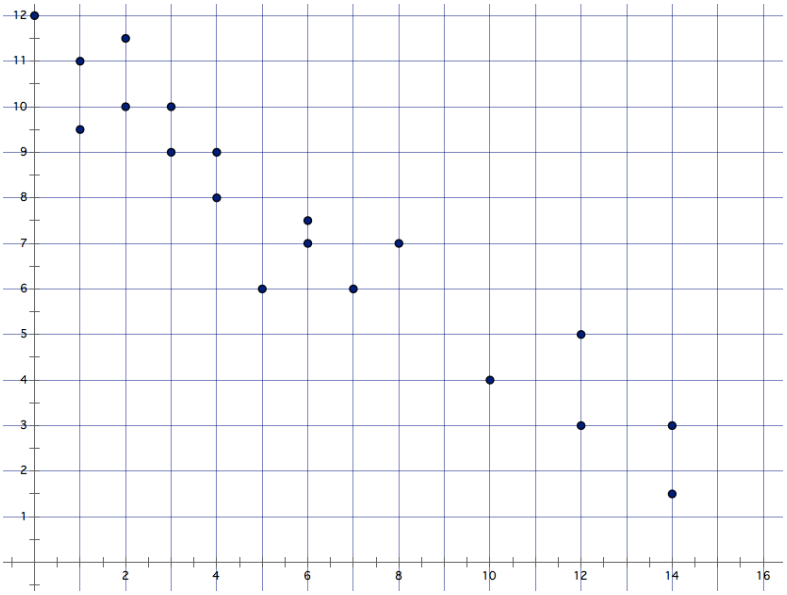
Topic: Creating and analyzing scatter plots.

Determine whether a linear or a quadratic model would be best for the given scatter plots. Then sketch a model on the graph that could be used to make predictions.

6.

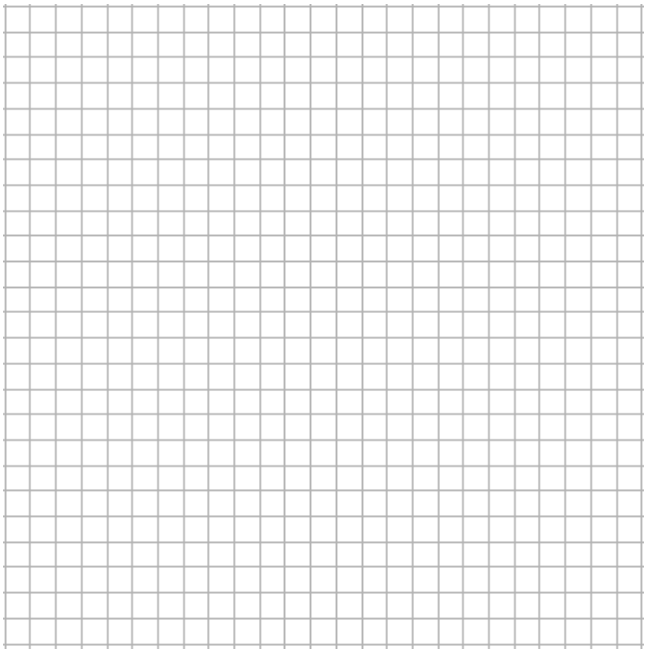


7.



8. Use the data to make a scatter plot. Then answer the questions.

Weeks since school started	Money in savings
1	200
3	175
4	162
7	120
10	87
13	57
20	5



9. Is the correlation of the graph positive or negative? Why?

10. What would you estimate the correlation coefficient to be? Why?
(If you have a calculator or software that can calculate it precisely then do so.)

11. Create a regression line and find the regression equation. What is the regression equation?

12. What does the slope of the regression equation mean in terms of the variables?

13. Most school years are 36 weeks. If the rate of spending is kept the same how much more money needs to be saved during the summer in order for there to be money to last all 36 weeks.

Warm-up: Fit a Parabola to Data

2. The table shows the average sale price, p, of a house in New York City, for various years, t, since 1960.

Years since 1960, t	0	1	2	3	4	5	6
Average sale price (in thousands of dollars), p	45	36	29	24	21	20	21

a. What type of function most appropriately represents this set of data? Explain your reasoning.

b. In what year is the price at the lowest? Explain how you know.

c. Write a function to represent the data. Show your work.

d. Can this function ever be equal to zero? Explain why or why not.

e. Mr. Samuels bought his house in New York City in 1970. If the trend continued, how much was he likely to have paid? Explain and provide mathematical evidence to support your answer.

For loops:

```
for (i = 0; i < 10; i++) {...
```

In Ch, the **for** keyword is followed by a set of parentheses containing three parts separated by semicolons.

for (*init*; *test*; *update*)

The intent of C's **for** loop is to enable stepping a variable through a series of numbers, like counting from 0 to 9. The part before the first semicolon (*init*) is performed as soon as the **for** statement is reached; it is for initializing the variable that will count. The part between the two semicolons (*test*) is evaluated before each iteration to determine whether the iteration should be repeated. And the part following the final semicolon (*update*) is evaluated at the end of each iteration to update the counting variable for the following iteration.

In practice, **for** loops are used most often for counting out n iterations. The standard idiom for this is the following.

```
for (i = 0; i < n; i++) {
   body
}
```

Here we have a counter variable i whose value starts at 0. With each iteration, we test whether i has reached n or not; and if it hasn't, then we execute the **for** statement's body and then perform the i++ update so that i goes to the following integer. The result is that the body is executed for each value of i from 0 up to n – 1.

But you can use a **for** loop for other purposes, too. In the following example, we display the powers of 2 up to 512. Notice how the update portion of the **for** statement has changed to "p *= 2".

```
for (p = 1; p <= 512; p *= 2) {
   printf("%d\n", p);
}
```

While loops:

```
while (test) {
body
 }
```

The **while** statement works by checking the test condition, if it is true then the body will be executed. After the body is executed it will check the test condition again and repeat until the test condition is no longer true.

```
i=10
while (i >= 0) {
   printf("%d\n", i);
   i--;
}
```

Again, the test expression requires a set of parentheses around it, there is no colon, and we use braces to surround the loop's body.

CODE REVIEW RUBRIC

STUDENT: _____

EVALUATOR: _____ DATE: _____

CRITERIA	UNSATISFACTORY (Below Performance Standards)	PROFICIENT (Meets Standards)	ADVANCED (Demonstrates Exceptional Performance)
Description Documented Code (40%)	● Fails to meet any of the Proficient Descriptors. ● Common Pitfall: - Assuming a line is self explanatory that actually isn't (when in doubt, add a comment) 0 - - - - - - - - - - - - 7 - - - - - - - - - - - - 13	● Each line is either self explanatory or comment helps understand it ● Code runs without errors and makes the Linkbot move in a quadratic fashion. 14 - - - - - - - - - - - - 16 - - - - - - - - - - - - 18	In addition to meeting the PROFICIENT criteria... ● Code is parameterized 19 - 20
Description Justification of Motion (60%)	● Fails to meet any of the Proficient Descriptors ● Common Pitfalls: - Graph doesn't start at 0 - Graph scales are inconsistent, too big, or too small 0 - - - - - - - - - - - -11 - - - - - - - - - - - - 22	● Student can justify how their car is moving quadratically by referencing the graph, equation, and table. ● Student can explain how the motion of the Linkbot is an approximation of a quadratic, and is actually a piecewise linear graph. ● Student references the average rate of change between two measured points during their explanation. 23 - - - - - - - - - - - - 25 - - - - - - - - - - - - 27	In addition to meeting the PROFICIENT criteria ... ● Student can explain the equation of each piece of the piecewise graph 28 - - - - - - - - - - - -29 - - - - - - - - - - - - 30

COMMENTS:

Group Quiz

Coefficients of 3 and 5

If we list first few values greater than zero for quadratics with coefficients of three and five below 50 we get 3, 5, 12, 20, 27, and 45. The sum of these values is 112. Find the sum of all values of quadratics with coefficients of 3 or 5 below 1000.

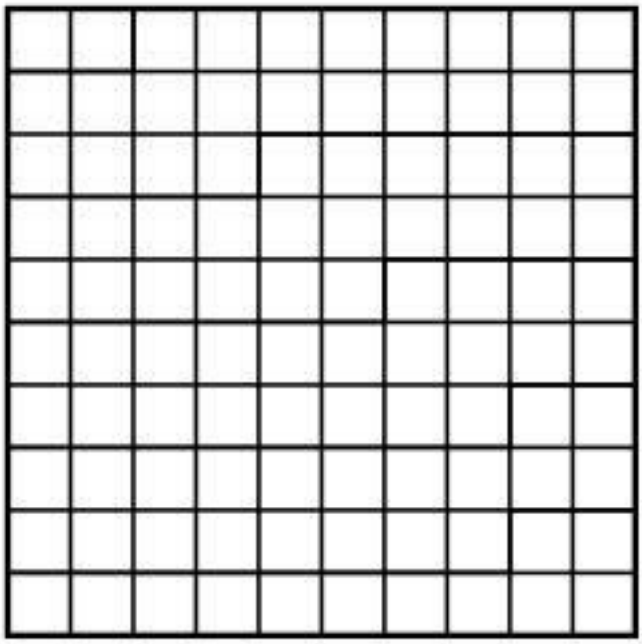
Robots & Math **Name**_____

Gravity Car Assessment

1) You have been selected through a contest to win one million dollars. You must shoot a basketball from the top of a building so that it goes over a billboard and lands in a hoop on the ground. The following table compares the height of the basketball from the ground to the amount of time it is in the air.

IV:	0	1	2	3
DV:	80	128	144	128

a) Graph the data. Be sure to label your axes and pick a good scale.



b) Write the recursive rule for the function.

c) Use the recursive rule to find f(4) (show all calculations).

d) Write the explicit rule of the function.

e) Use the explicit rule to determine how many seconds it takes the ball to get to the rim (show all calculations).