A Galaxy Far, Far Away...

M104 in Virgo, as seen by Hubble

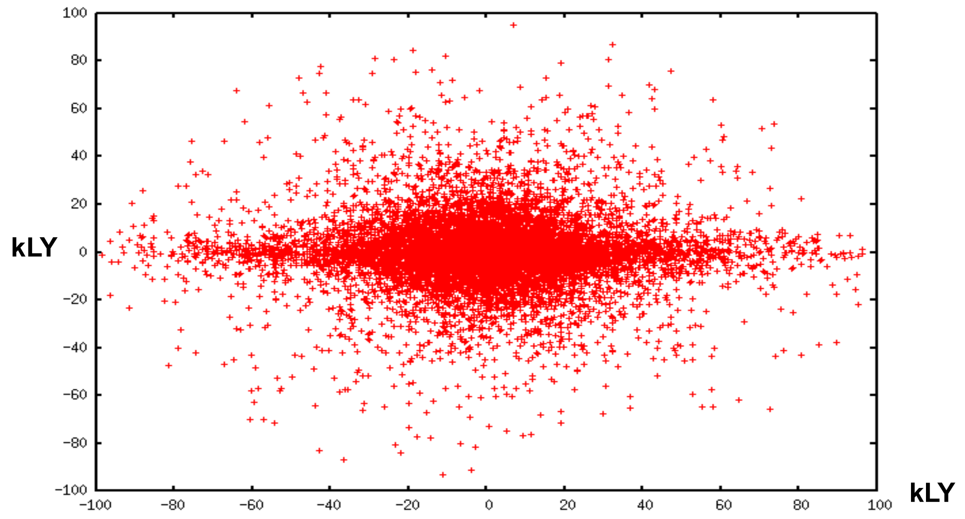
Galaxies like the “Sombrero Galaxy” above are beautiful, awe-inspiring objects. But the're also ready-made physics experiments that we can learn a lot from.

For example, astronomers can measure the speeds of stars in distant galaxies by looking at doppler shifts in the stars' spectral lines. These measured speeds let us test our knowledge of classical mechanics.

A Pet Galaxy, All Your Own:

Here's how you can get a file containing data for a small simulated galaxy. This galaxy contains only 10,000 stars. The data file give each star's coordinates and speed.

```
wget http://tinyurl.com/galaxy-dat
mv galaxy-dat galaxy.dat
```



Download the file above, and let's take a look at it.

The Contents of galaxy.dat:

Each row of the data file represents a single star. The first three columns are the star's x, y, and z coordinates, measured in kilo-light-years. The fourth column is the star's speed, as it orbits around the center of the galaxy.

```
     x          y          z       speed
 6.980059    2.361664  41.120018 22.050885
-0.152892    8.782960 -29.183423 20.863342
-16.502954  19.529388 -16.108144 20.828065
 28.622008  26.553047   1.133840 21.821036
  0.341239  -2.652743 -13.211824 16.668669
 14.841344 -55.441293   7.468305 23.124437
 -5.888803  25.012676 -10.480873 20.449301
 -7.096150  -7.318255  21.279209 19.713488
.
.
.
```

The stars have a distance from the center of the galaxy that ranges from zero to 100 kilo-lightyears.

(The speed is in arbitrary units. We'll only be interested in how it varies.)

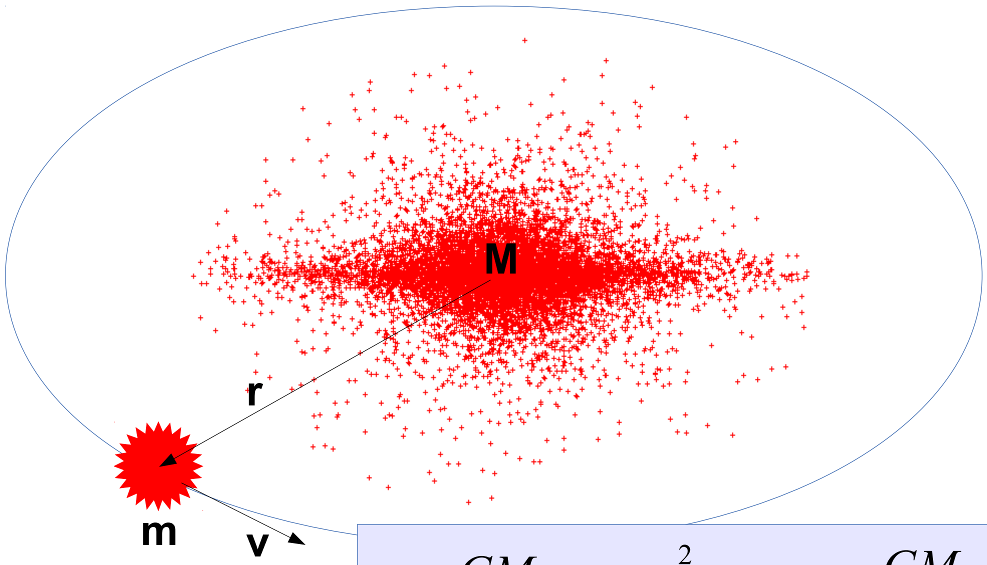
We can use these speed values to test our theories of physics. Each star's velocity should be determined by Newton's laws of motion and the law of gravitation (with possible corrections due to relativity).

We aren't given the mass of each star, but let's just assume they all have the same, average, mass.

One way to find each star's speed would be to add up the gravitational forces on that star due to all of the other stars. That's a big job, though, and it gets bigger quickly when we deal with even bigger galaxies. Still, we could do it with a computer program if we wanted to.

Expectations about Speed:

Consider a single star orbiting far from the center of the galaxy.....



$$F=\frac{GMm}{r^2}=\frac{mv^2}{r}, \qquad v^2=\frac{GM}{r}$$

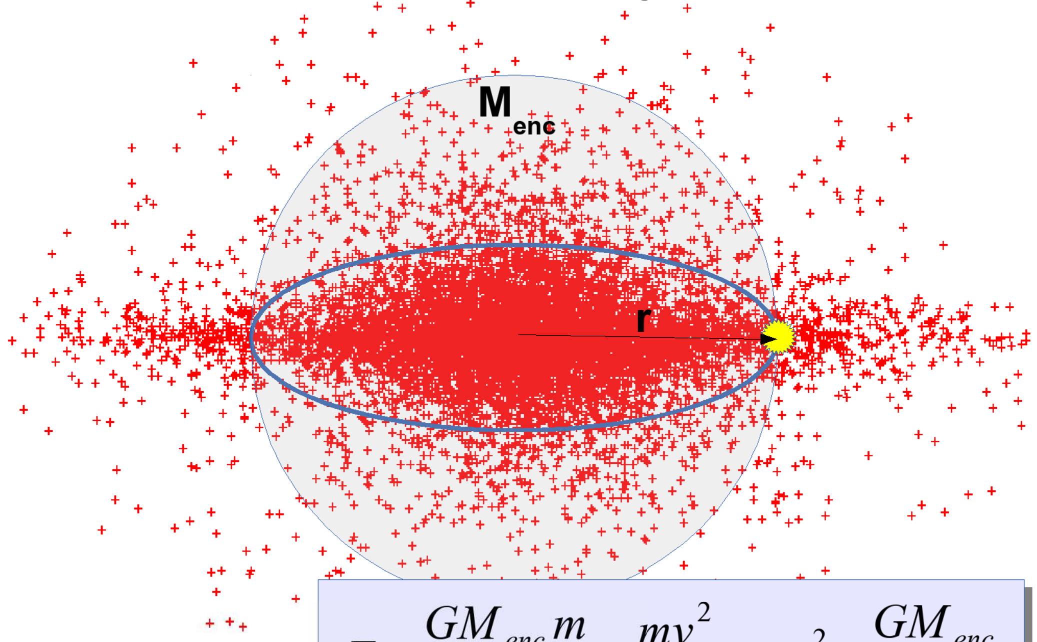
In reality, astronomers are seldom given positions of individual stars in distant galaxies. Instead, they look at the density of stars. So even if we developed a program to calculate forces due to individual stars, it would be of little use real-world use.

As an approximation, let's assume that the mass of the galaxy, M, acts like a single point-mass concentrated at the center. We can then estimate the speed of the star above by writing down the condition that the inward pull of gravity must be matched by the apparent “centrifugal force”, and then solving for the velocity, v.

(We're also assuming that the orbit is circular, so the velocity is constant.)

This tells us that v depends on only M and the star's distance from the center, r.

Speed of Stars Inside the Galaxy:



$$F = \frac{GM_{enc}\,m}{r^2} = \frac{mv^2}{r}\ , \qquad v^2 = \frac{GM_{enc}}{r}$$

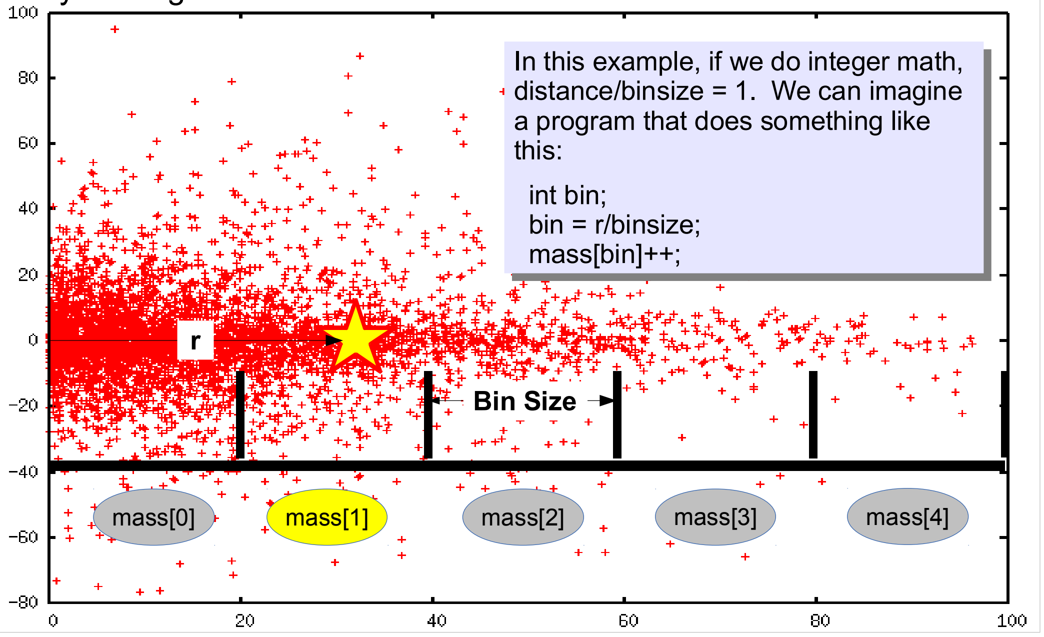
We can extend this approximation to stars *inside* the galaxy by assuming that the dominant contribution to the gravitational forces acting on a star will be due to the mass, M_{enc}, enclosed by the star's orbit. (The pulls of the masses *outside* the orbit will tend to cancel each other out.)

So, with these assumptions, we should be able to estimate the orbital speed of any star in the galaxy. We just need to find the mass enclosed by the star's orbit, and then apply the lower right-hand equation above.

It look's like we're going to want M_{enc} as a function of r, so let's see how we might find that.

Counting the Mass in “Shells”:

Let's have an array, mass[bin], where “bin” is the shell number. Then, for each star, we can find out which shell that star belongs to by looking at the star's distance from the core.



In this example, if we do integer math, distance/binsize = 1. We can imagine a program that does something like this:

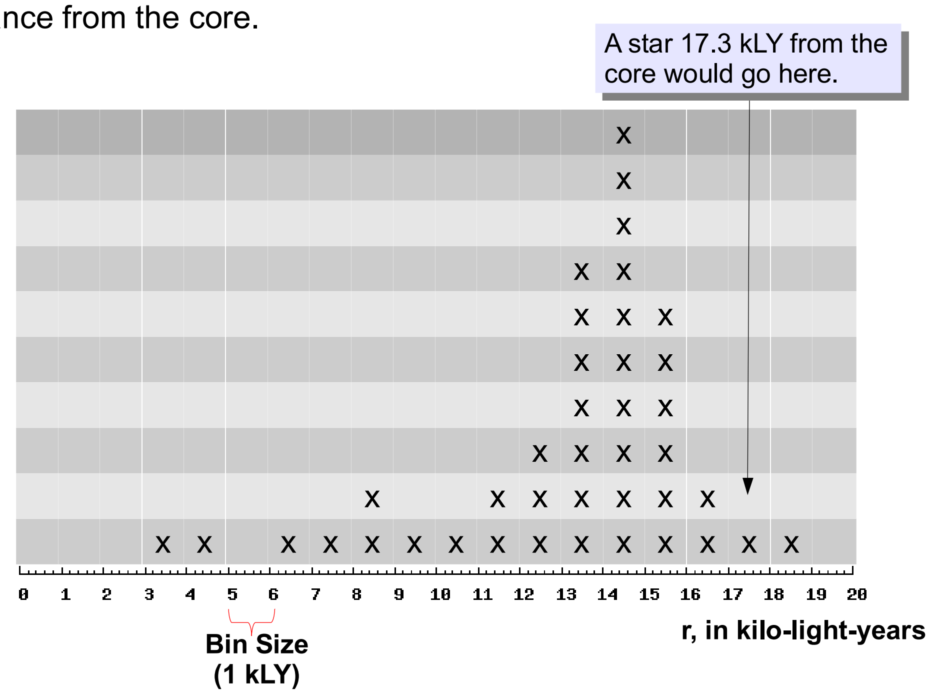
```
int bin;
bin = r/binsize;
mass[bin]++;
```

One way to look at the mass distribution is by slicing the galaxy up into layers like an onion. We can find how many stars lie in each of these layers, or “shells”. We do this by using an array and making a histogram, just as we did last week with the energies of gamma rays. In this case, we're making a histogram of the distance, r, at which stars orbit the galaxy.

Note that we're measuring mass in units of “stars”, and that we're still assuming every star has the same mass.

Constructing the Histogram:

For each star, we add 1 the appropriate bin, chosen by to the star's distance from the core.

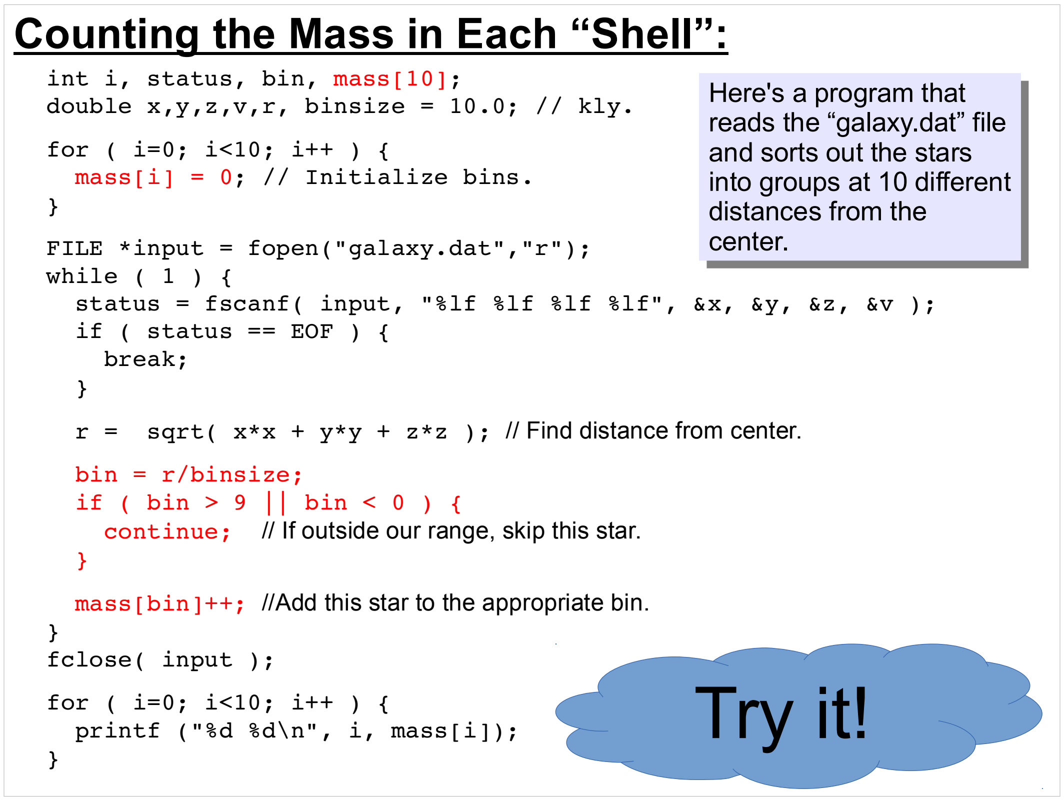


We could use any bin size and any distance range. It's up to us to choose something appropriate.

This is just like the gamma-ray example we looked at last week.

How would you go about writing a program that reads the “galaxy.dat” data file and constructs such a histogram? Remember that we were told that these stars have distances between zero and 100 kLY from the core. Let's divide that range into ten 10 kLY-wide bins.

Think about it before you look at the next page.



Counting the Mass in Each "Shell":

```
int i, status, bin, mass[10];
double x,y,z,v,r, binsize = 10.0; // kly.

for ( i=0; i<10; i++ ) {
  mass[i] = 0; // Initialize bins.
}

FILE *input = fopen("galaxy.dat","r");
while ( 1 ) {
  status = fscanf( input, "%lf %lf %lf %lf", &x, &y, &z, &v );
  if ( status == EOF ) {
    break;
  }

  r =  sqrt( x*x + y*y + z*z ); // Find distance from center.

  bin = r/binsize;
  if ( bin > 9 || bin < 0 ) {
    continue;   // If outside our range, skip this star.
  }

  mass[bin]++;  //Add this star to the appropriate bin.
}
fclose( input );

for ( i=0; i<10; i++ ) {
  printf ("%d %d\n", i, mass[i]);
}
```

Here's a program that reads the "galaxy.dat" file and sorts out the stars into groups at 10 different distances from the center.

Try it!

(Note that I've omitted the #include statements and so forth to save space. You'll need to add these back in to get a complete program.)

This program just counts up how many stars lie within each distance range. (Remember: we're counting mass in units of "stars").

We should also add an "overflow/underflow" counter above the "continue" statement, as we did in our gamma-ray program, and print out the number of over/underflows at the end of the program. Can you figure out how to do this?

We might also want to count the total number of stars. A future data file might contain an unknown number of them. How would you modify the program to count them?

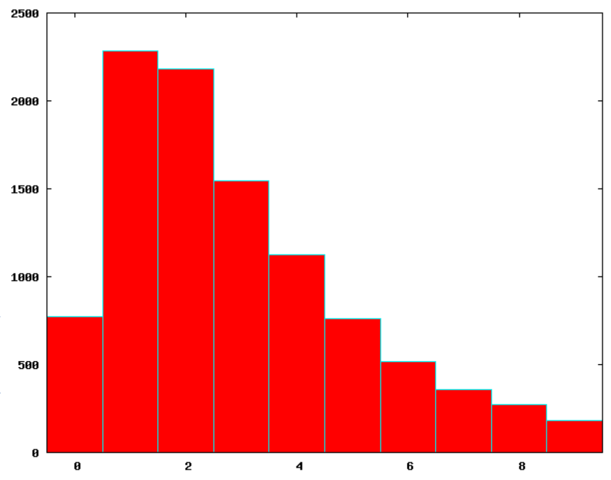
The Mass Distribution:

Here's what the program should print out, and a graph of the results.

Notice that the first shell (the innermost one) doesn't have much mass. This is expected: as r goes toward zero, the enclosing sphere gets smaller and smaller, so the mass enclosed should go toward zero, too.

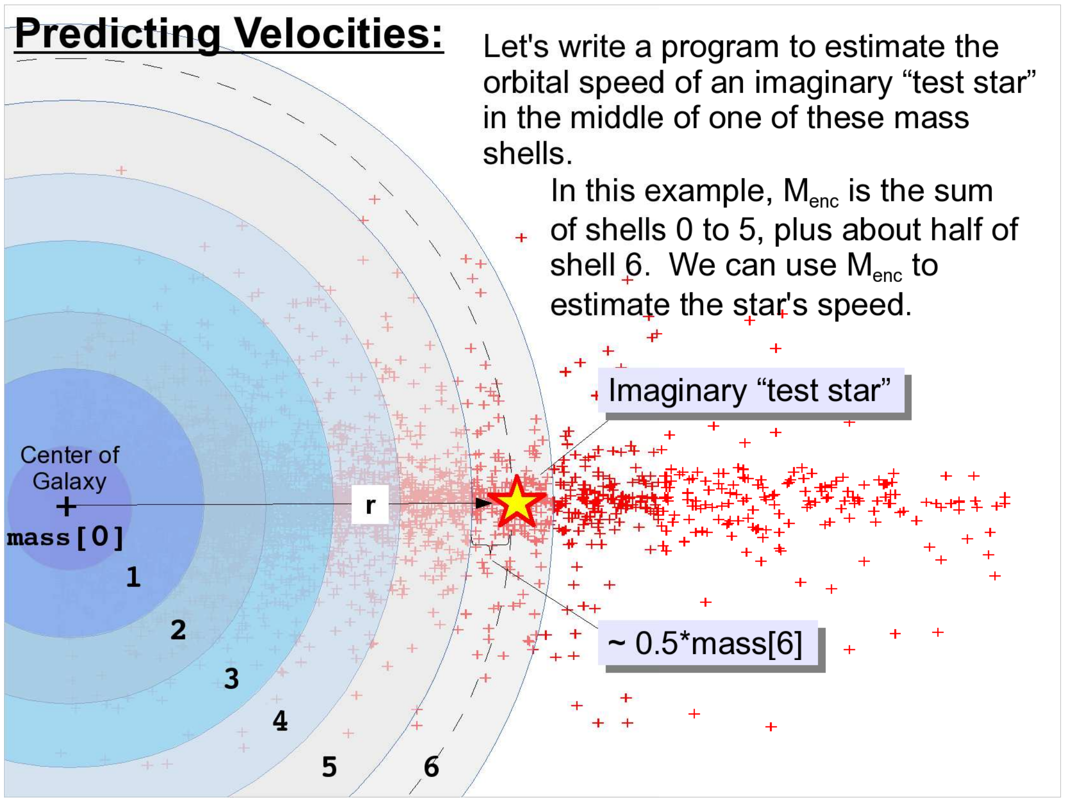
Shell Mass

```
0  772
1  2285
2  2182
3  1546
4  1124
5  764
6  516
7  356
8  272
9  182
```



The mass distribution has a hump near where the galaxy's central bulge ends, and it trails off toward zero as we get very far from the center.

When astronomers collect data from a distant galaxy, they usually get something like this. Although we may not be able to see the galaxy's individual stars with a telescope, we can see how the brightness of the galaxy varies with distance from the core, and we can use that brightness to estimate how the mass varies with distance. (More brightness implies more stars, which implies more mass.)



Predicting Velocities:

Let's write a program to estimate the orbital speed of an imaginary "test star" in the middle of one of these mass shells.

In this example, M_{enc} is the sum of shells 0 to 5, plus about half of shell 6. We can use M_{enc} to estimate the star's speed.

We place our "test star" in the middle of a shell. The distance from the star to the center of the galaxy will be:

r = binsize*(bin + 1.5)

where "bin" is the shell number and "binsize" is the thickness of each shell (10 kLY in our previous program).

The enclosed mass will be approximately:

$$M_{enc} = 0.5 * mass[bin] + \sum_{i=0}^{bin-1} mass[i]$$

and the predicted speed (squared) would be:

$$v^2 = \frac{GM_{enc}}{r}$$

A Programming Challenge:

OK, so given all of this, can you modify the previous program to make it print out the predicted velocity for an imaginary test star in the middle of each mass shell? Here are some tips:

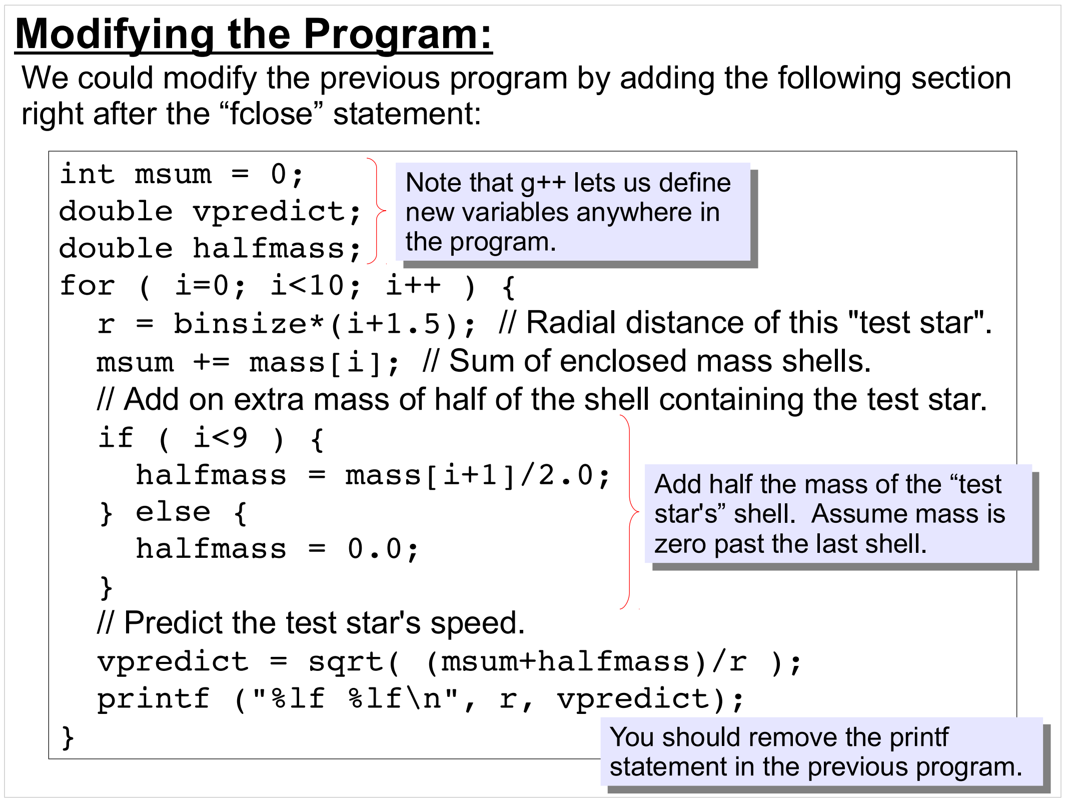
* Ignore the measured velocities that are in “galaxy.dat”. Later we'll compare our predicted velocities with these.

* Add a section at the bottom of your program that loops through the shells, calculating r and M_{enc} for test stars.

* For each shell, print out the test star's “r” and predicted velocity.

Don't worry about units. Assume G=1, and use “stars” as the mass unit and kLY as the distance unit.

Think about how you'd do it, before looking at the next page.



Modifying the Program:

We could modify the previous program by adding the following section right after the “fclose” statement:

```
int msum = 0;
double vpredict;
double halfmass;
for ( i=0; i<10; i++ ) {
  r = binsize*(i+1.5); // Radial distance of this "test star".
  msum += mass[i]; // Sum of enclosed mass shells.
  // Add on extra mass of half of the shell containing the test star.
  if ( i<9 ) {
    halfmass = mass[i+1]/2.0;
  } else {
    halfmass = 0.0;
  }
  // Predict the test star's speed.
  vpredict = sqrt( (msum+halfmass)/r );
  printf ("%lf %lf\n", r, vpredict);
}
```

Note that g++ lets us define new variables anywhere in the program.

Add half the mass of the “test star's” shell. Assume mass is zero past the last shell.

You should remove the printf statement in the previous program.

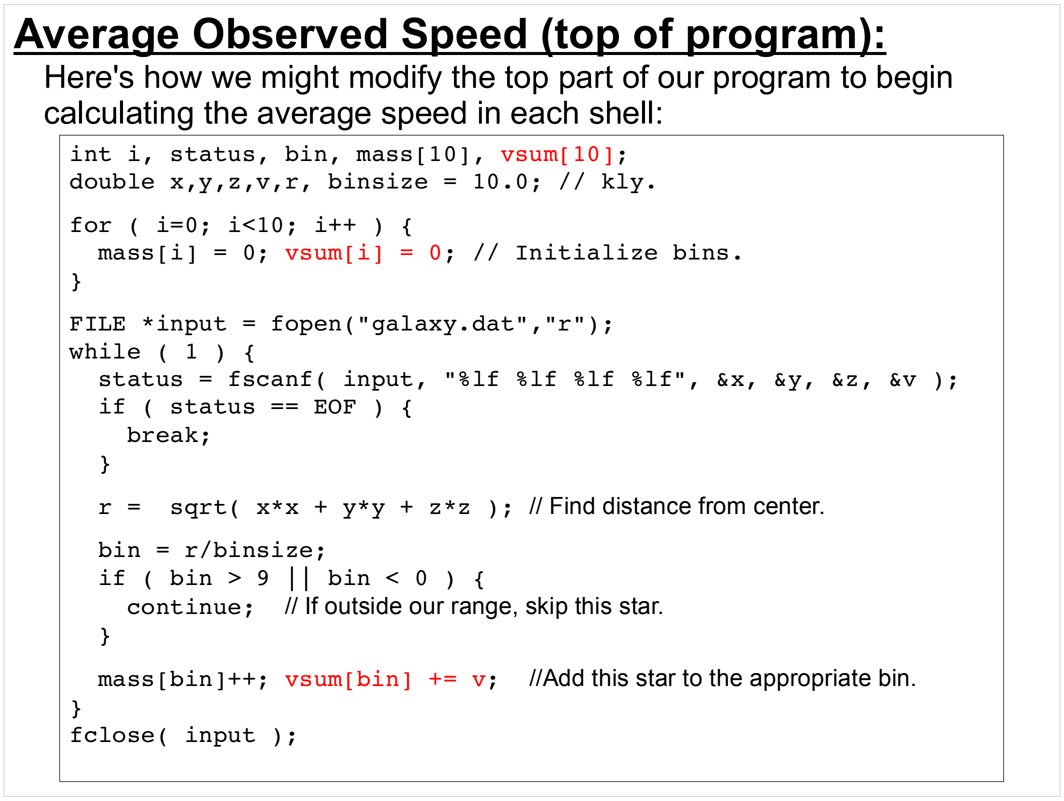
Our choice of units (G=1, “stars” as the mass unit, and kLY as the distance unit) will turn out to be convenient because these are the same units that were used for the measured velocities in “galaxy.dat”, so we'll be able to directly compare our predicted velocities with those.

How would we modify our program to do that? Here's a possible strategy:

* While we're reading “galaxy.dat”, let's keep track of the sum of the measured star velocities in each of our shells. We can make an array like vsum[i] to hold these numbers.

* Then, when we get to the section shown in the slide above, we can print out the average measured velocity for stars in each slice by just dividing vsum[i] by mass[i], since mass[i] contains the number of stars in that slice.

Think about how you'd write that before looking at the next slide.



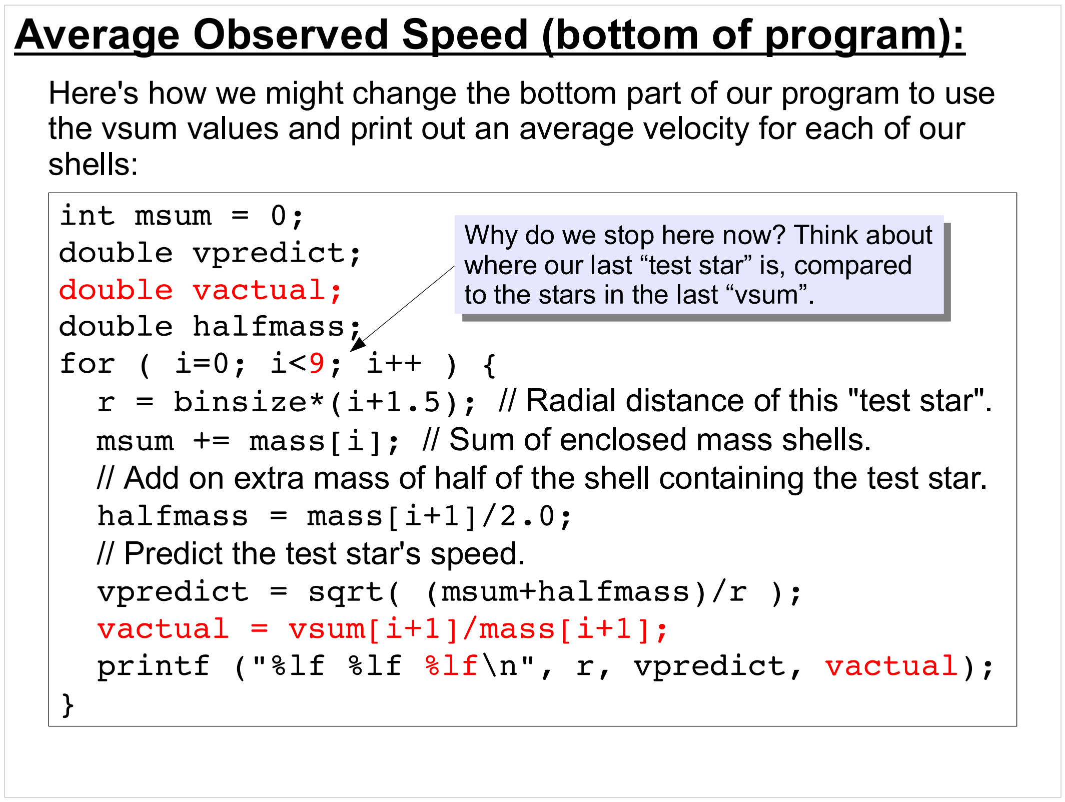
Average Observed Speed (top of program):

Here's how we might modify the top part of our program to begin calculating the average speed in each shell:

```
int i, status, bin, mass[10], vsum[10];
double x,y,z,v,r, binsize = 10.0; // kly.

for ( i=0; i<10; i++ ) {
  mass[i] = 0; vsum[i] = 0; // Initialize bins.
}

FILE *input = fopen("galaxy.dat","r");
while ( 1 ) {
  status = fscanf( input, "%lf %lf %lf %lf", &x, &y, &z, &v );
  if ( status == EOF ) {
    break;
  }

  r =  sqrt( x*x + y*y + z*z ); // Find distance from center.

  bin = r/binsize;
  if ( bin > 9 || bin < 0 ) {
    continue;   // If outside our range, skip this star.
  }

  mass[bin]++; vsum[bin] += v;   //Add this star to the appropriate bin.
}
fclose( input );
```

Here's one way to start. Define a new array, vsum[i]. Initialize all of its element to zero. Then, as we loop through all of the stars in “galaxy.dat”, add up the velocities of all of the stars in each bin.

On the next page, we'll look at how to use these sums to write out the average speeds. Think about it before you look.



Average Observed Speed (bottom of program):

Here's how we might change the bottom part of our program to use the vsum values and print out an average velocity for each of our shells:

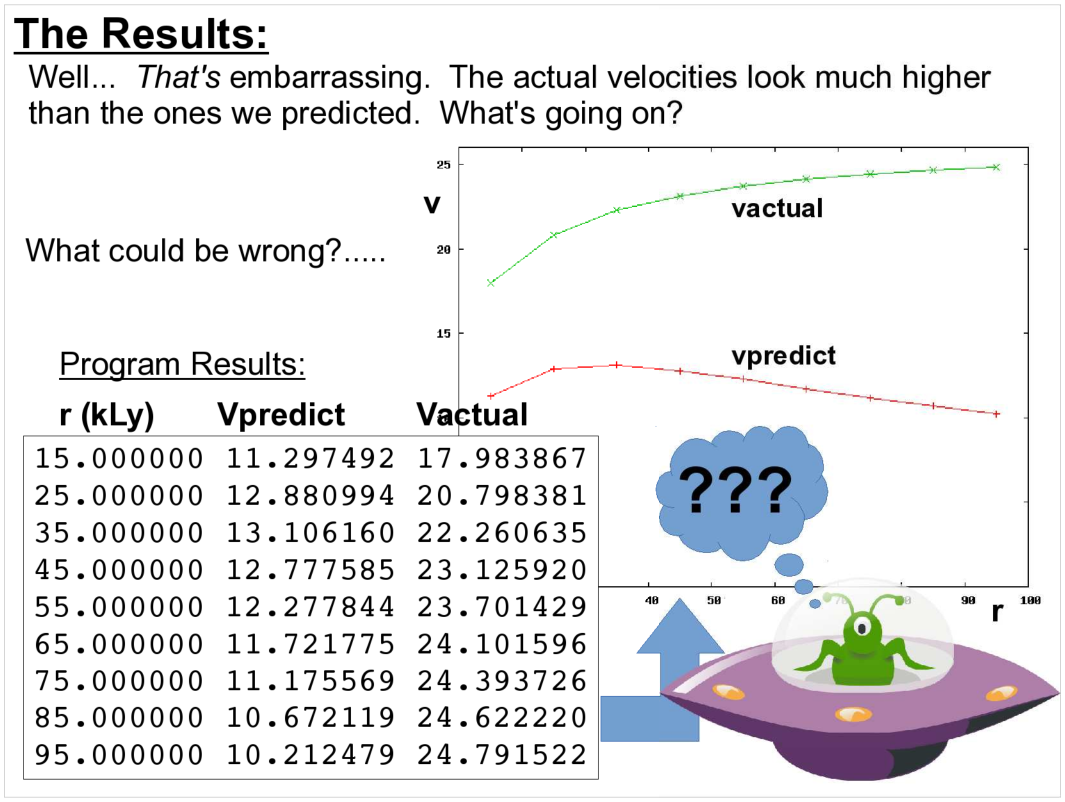
```
int msum = 0;
double vpredict;
double vactual;
double halfmass;
for ( i=0; i<9; i++ ) {
  r = binsize*(i+1.5); // Radial distance of this "test star".
  msum += mass[i]; // Sum of enclosed mass shells.
  // Add on extra mass of half of the shell containing the test star.
  halfmass = mass[i+1]/2.0;
  // Predict the test star's speed.
  vpredict = sqrt( (msum+halfmass)/r );
  vactual = vsum[i+1]/mass[i+1];
  printf ("%lf %lf %lf\n", r, vpredict, vactual);
}
```

Why do we stop here now? Think about where our last "test star" is, compared to the stars in the last "vsum".

So, now, our program should print out three columns of numbers: r, vpredict and vactual.

If all of our assumptions are reasonable, and if we know how physics works, then the last two columns should approximately match.

Do they?



The Results:

Well... *That's* embarrassing. The actual velocities look much higher than the ones we predicted. What's going on?

What could be wrong?.....

Program Results:

r (kLy) Vpredict Vactual

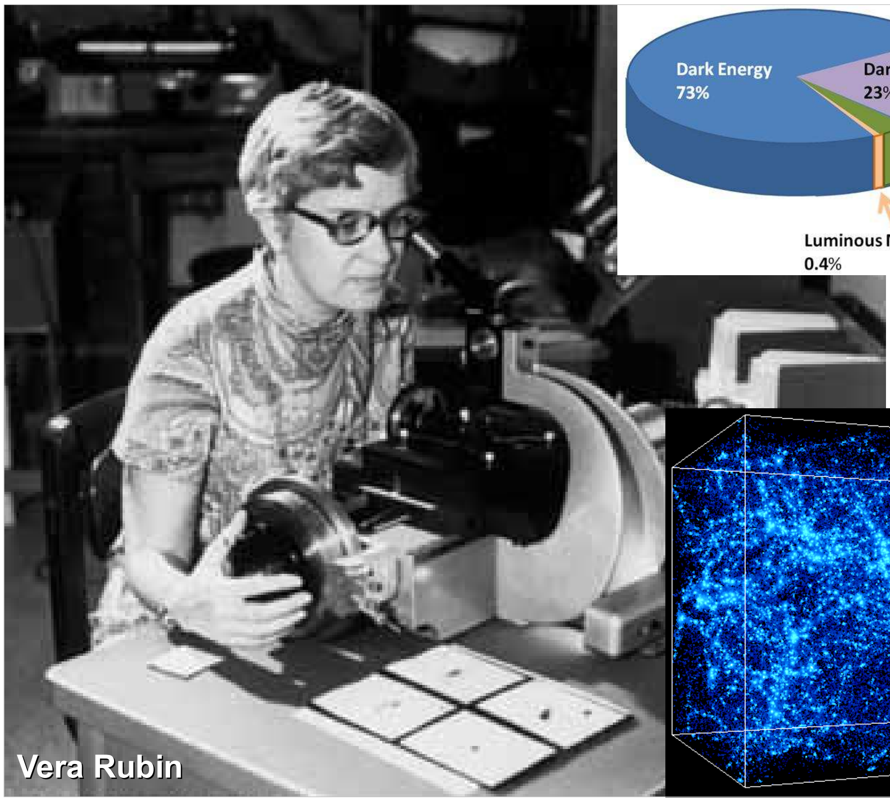
```
15.000000 11.297492 17.983867
25.000000 12.880994 20.798381
35.000000 13.106160 22.260635
45.000000 12.777585 23.125920
55.000000 12.277844 23.701429
65.000000 11.721775 24.101596
75.000000 11.175569 24.393726
85.000000 10.672119 24.622220
95.000000 10.212479 24.791522
```

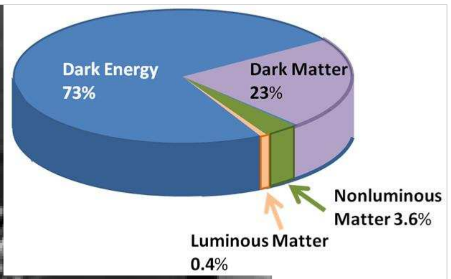
What could have gone wrong?

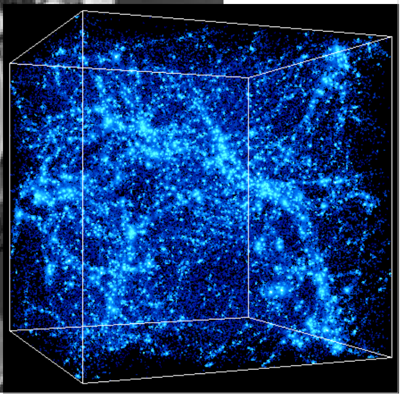
* One of our approximations is just way off.

* The data in galaxy.dat is wrong, because of some measurement error.

* We don't understand the physics.







In the 1970s Vera Rubin went through a procedure similar to what we've just done, meticulously measuring the velocities of stars in other galaxies and comparing those velocities with predictions based on what we know about physics. She found the same anomaly we saw in our data.

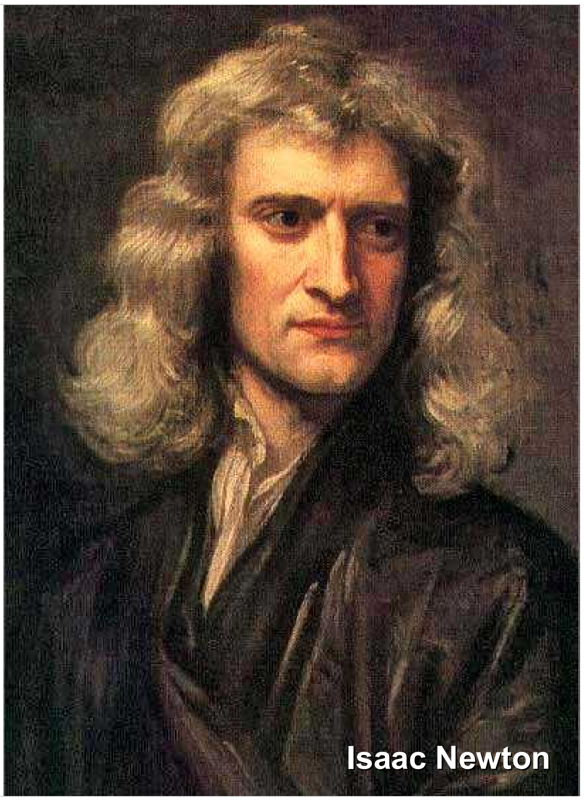
Today, after decades of further research, this anomaly persists. Many physicists believe that the velocity mismatch is due to the existence of a new kind of matter, dubbed “Dark Matter”, that we have not yet detected. In order to account for the observed velocities, galaxies must have about five times as much Dark Matter as regular matter!

Current models of the history of our universe lead us to believe that galaxies are laid out along a web of dark matter, the gravitational pull of which controlled the evolution of galaxies and galaxy clusters.

$$F = ?\,\frac{GMm}{r^2}$$

Although many things lead us to believe that Dark Matter exists, we might also be able to explain our observations by modifying the law of gravity or Newton's laws of motion. Maybe things behave differently at large distances or at very low accelerations?

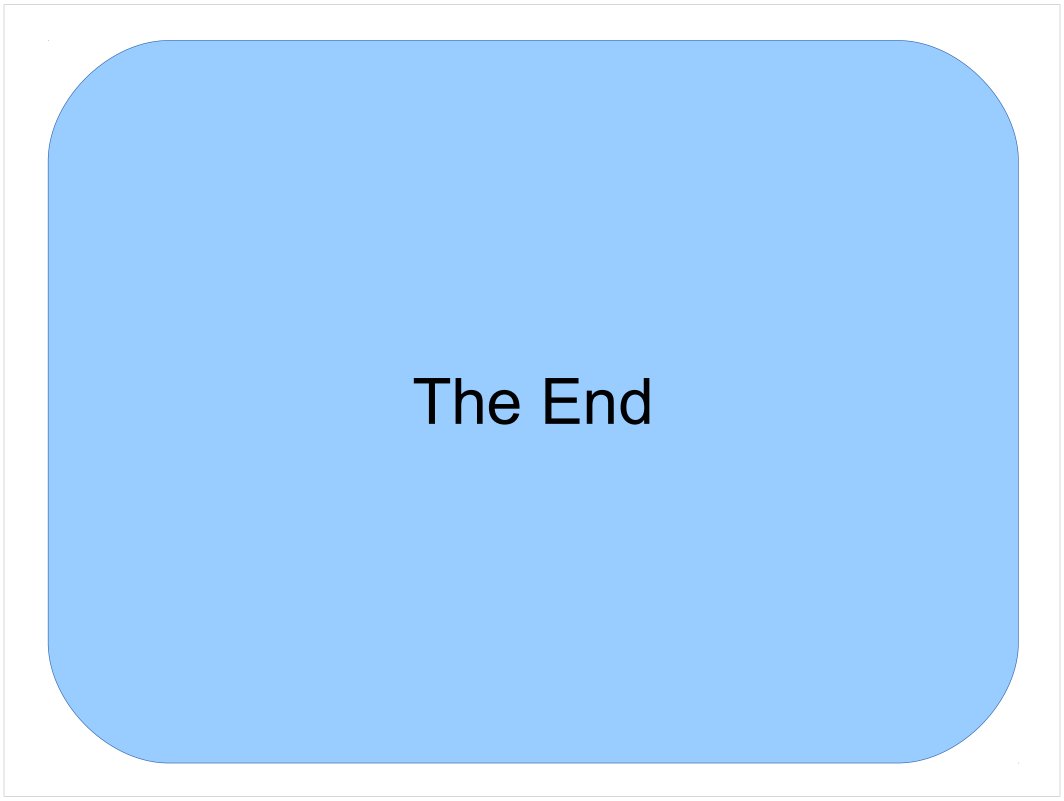
Nobody knows the answer yet, but I'm hoping that new scientists like you will someday find the answer.



Dark Matter, and the equally mysterious Dark Energy, are two of the biggest puzzles in physics today, but we seem to be right on the edge of understanding both of them.

Searches for possible Dark Matter particles are ramping up rapidly, and theorists are building testable theories that might explain Dark Energy, a force that's causing the expansion of our universe to accelerate.

This is a very exciting time in physics, and I'm glad you're here to participate in these discoveries.



The End

Thanks!