____

Name

Recitation Assignment # 5

Boo.

For this especially spooky Halloween recitation, you will be applying the following new concepts:

Physical

- Special Relativity

Computational

- Subroutines
- Packing Data

This week, the idea is that I will provide you with a "simulation" series of events: a space station of length, 1ls, with a person running through it at speed 0.05c (pretty fast), and light beam bouncing off of two mirrors on the interior of the station. Meanwhile, a spaceship flies past at a speed specified by the user. Your goal: Transform these observations to the perspective of the spaceship.

1. Log on, and download the following file to your "Contemporary1" directory (or whatever you've chosen to call it):

   `http://www.physics.drexel.edu/∼goldberg/prog5_goldberg.py`

   Take a look at the file. It's an example of a subroutine. The first line is a def statement:

   ```
   def goldberg_frame(t,vs):
           # t is an input array
           # vs is the speed of the spaceship (in units of the speed of
           #   light)
           # the output is an array of the format: [xb,xf,xl,xp,xs]
   ```

   which means that you need to provide two pieces of input:

   - A list of "times" (a "[ ]" List)
   - The speed of the spaceship (a single number, expressed as a fraction of the speed of light)

   Bear in mind that everything in this exercise is in units of c. You not need to put $3 \times 10^8 m/s$ anywhere in your code.

   The output of this code is a little complicated (and I'll describe it shortly).

   Note: You will not edit "prog5_goldberg.py" for this assignment.

2. Now, create a program called “prog5.py” which uses my subroutine. The top lines should read:

   ```
   from pylab import *
   from prog5_goldberg import *
   ```

   This basically says that you should be allowed to use my subroutine in your program.

3. Within prog5.py, create a “list” (t=[ ]) of times from 0 to 10 seconds. You can use whatever dt you like. Also, set a variable called “vs” to be the speed of your ship. Express it in units of speed of light. The “goldberg” code (as it will henceforth be called) can be called via:

   ```
   data_g=goldberg_frame(t,vs)
   ```

   What is data_g? It’s a list of lists. The first is the position of the back of my space station at various times, the second is the front of my station, and then the position of the lightbeam, the position of a person “walking” through the station, and the position of your ship, respectively. Thus:

   ```
   xl=data_g[2]
   ```

   sets the variable “xl” to the list generated by my subroutine.

4. Use your notes from the very first recitation. Set your code to create a space-time diagram (positions on the x-axis, time on the y), containing

   (a) The back of the ship at all times.
   (b) The front of the ship at all times,
   (c) The light beam bouncing back and forth,
   (d) The person walking through the station.

   If possible, make the axes have equal scaling (1ls=1s).

5. Now is your turn to make a subroutine.

   First, some practice. Create a file called “prog5_testsub.py”. In it, you will create a simple subroutine which computes the average speed of a particular point (as measured by the observer in a particular frame). It should be:

   ```
   def getspeed(x,t) :
         v=x/t
         return(v)
   ```

   Notice that the indenting is the same as with loops. That’s it! This code takes two numbers, and computes the speed of a particle (again, in units of the speed of light), which travels from the origin (0,0), to (x,t).

   In the main program, you’ll need the line “from prog5_testsub import *” at the very top. Later, you could compute something like:

   ```
   vtest=getspeed(xl[5],t[5])
   ```

vtest is equal to the “return” value. You can print it, use it for a calculation, whatever.

6. Now I want you to make a more complicated subroutine in a file called “prog5_boost.py”. This subroutine will take a set of positions and times (as measured by me), and turn them into a set of positions and times as measured by you.

   It should have the definition:

   ```
   def boost(x,t,v) :
   ```

   where v is the speed of the moving observed, in fractions of the speed of light. You should begin by creating two lists:

   ```
   x1=[]
   t1=[]
   ```

   And, for each point you now need to compute:

   $$\begin{aligned} x1[i] &= \gamma x[i] - v\gamma t[i] \\ t1[i] &= \gamma t[i] - v\gamma x[i] \end{aligned}$$

   and append them to the lists?

   Why no “c” in our transforms? Well, everything is in lightseconds and seconds, so in our units, c=1!

   How do we return the data? Well, see my example in “prog5_goldberg.py”. Return x1 and t1 (in that order) as a list of lists!

7. Transform the the times/positions of the front and back of the space station, the person walking through the station, and the lightbeam into the coordinates as measured by you on the ship, and plot them!

8. Using your speed subroutine, verify that the speed of light as measured by you is still 1.

9. Make sure you submit all three of your codes (prog5.py, prog5_boost.py, prog5_testsub.py) to Travis, with your name in a comment in all 3.

10. Of course, to win the recitation cup, you may want to figure out some interesting ways to make your code stand out from the rest. I am looking forward to seeing what you come up with.