In [5]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

from astropy import units as u
from astropy.io import ascii, fits
from astropy.table import Table, Column, MaskedColumn, join, unique
from astropy.cosmology import WMAP9 as cosmo
from astropy.coordinates import SkyCoord

from scipy.io.idl import readsav
from scipy.stats import gaussian_kde
from scipy.optimize import fsolve
from scipy import constants
```

In [6]:

```
%%html
<style>
body {
    font-family: "Comic Sans MS", cursive, sans-serif;
}
</style>
```

3) A passenger is running at her maximum velocity of 8 m/s to catch a train. When she is a dis-tance d from the nearest entry to the train, the train starts from rest with a constant acceleration a = 1.0 m/s2 away from her.

a) If d = 30 m and the passenger keeps running, will she be able to jump onto the train?

b) The critical separation distance is that at which the passenger can just catch the train. De-termine its value. What is the speed of the train when the passenger catches it? What is the train's average speed for the time interval from t = 0 until she catches it?

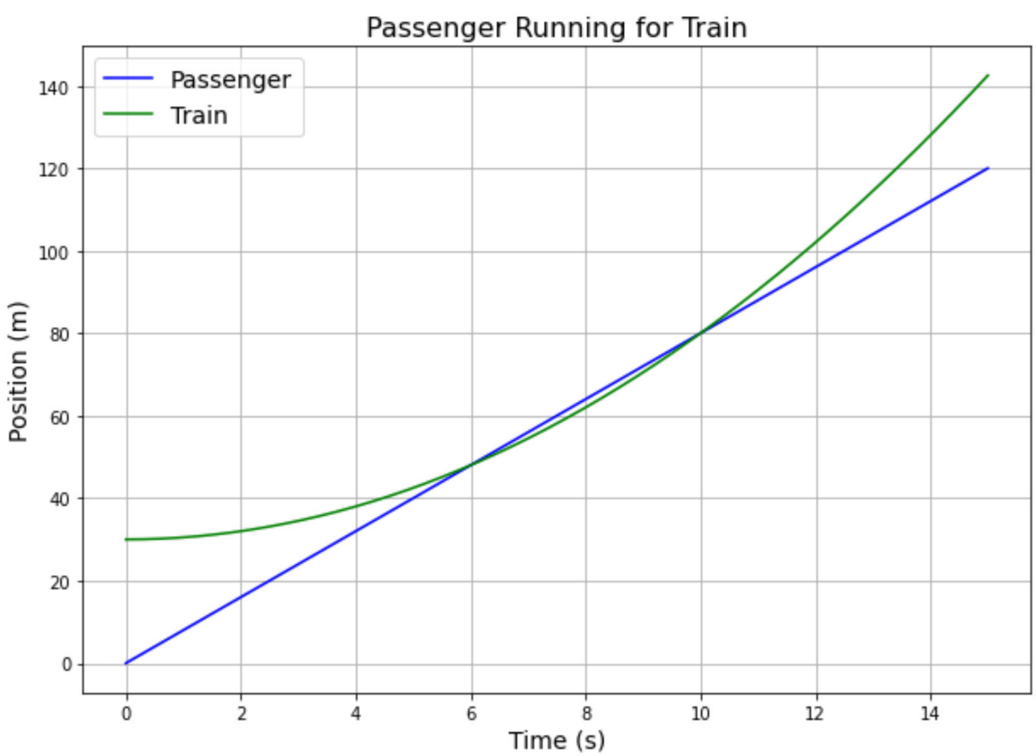
c With Python (or other plotting program), sketch the position function x(t) for the train, choosing x = 0 at t = 0. On the same graph sketch x(t) for the passenger for various initial separa-tion distances d, including d = 30m and the critical separation distance dc such that she just catches the train.

Answers: a) t_{catch} = 6 s or 10 s (explain each), b) $d_{critical}$ = 32 m, $v_{T,catch}$ = 8 m/s, v_T = 4 m/s

First just plot the situation where the passenger starts 30 m behind to tran to see if they're ever in the same positions at the same time so the passenger can catch the train.

In [15]:

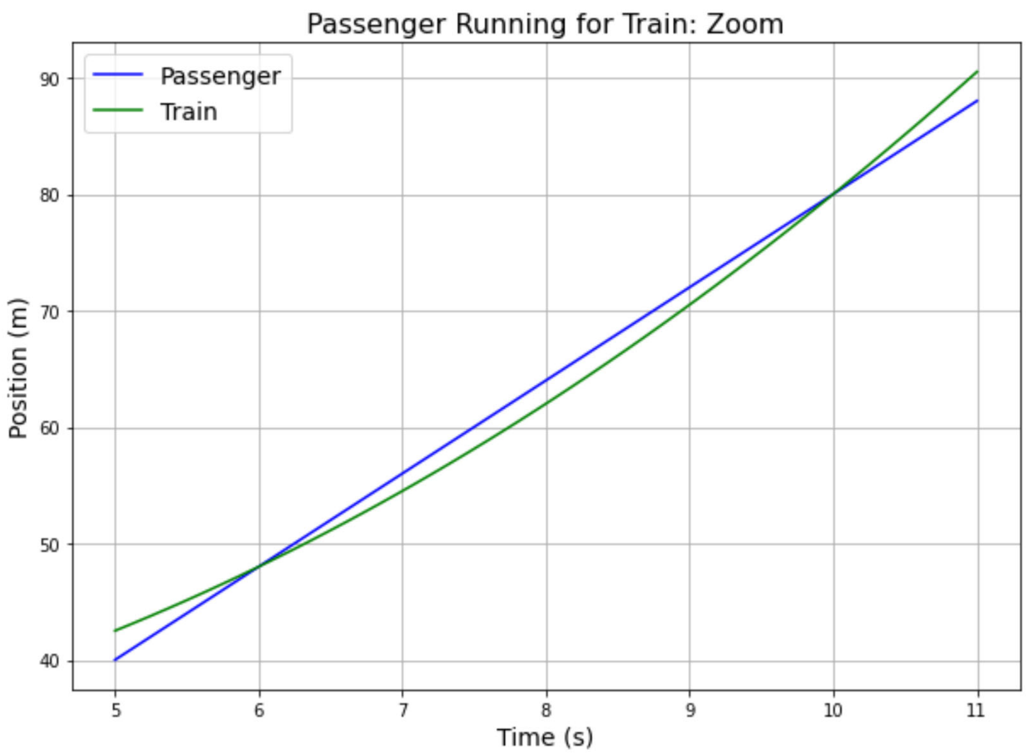
```
#Define the variables (floating point allows decimals)
#The first lines that are commmented out allow live input of the initial positions.

#xpassenger0 = float(input('xpassenger0 = '))
xpassenger0 = +0.0
vpassenger0 = +8.0
apassenger0 = +0.0

#xtrain0 = float(input('xtrain0 = '))
xtrain0 = +30.0
vtrain0 = +0.0
atrain0 = +1.0

#Set linear time steps from 0 to 15 seconds (default number of steps is 50)
time = np.linspace(0.,15.)

#Write the equations for the positions using the defined varialbes
xpassenger = xpassenger0 + vpassenger0*time + 0.5*apassenger0*time**2
xtrain = xtrain0 + vtrain0*time + 0.5*atrain0*time**2

#Plot the positions with respect to time in blue and green
plt.rcParams["figure.figsize"] = [10,7]
plt.plot(time, xpassenger, 'b', time, xtrain, 'g')

#make the plots pretty
plt.title('Passenger Running for Train', fontsize=16)
plt.xlabel('Time (s)', fontsize=14)
plt.ylabel('Position (m)', fontsize=14)
plt.legend(["Passenger", "Train"], loc ="upper left", fontsize=14)
plt.grid(True)
plt.show()
```



Zoom in on the span including the two places where the curves intersect and the passenger and train are at the same position. Confirm that they're at 6 s and 10 s as calculated.

In [16]:

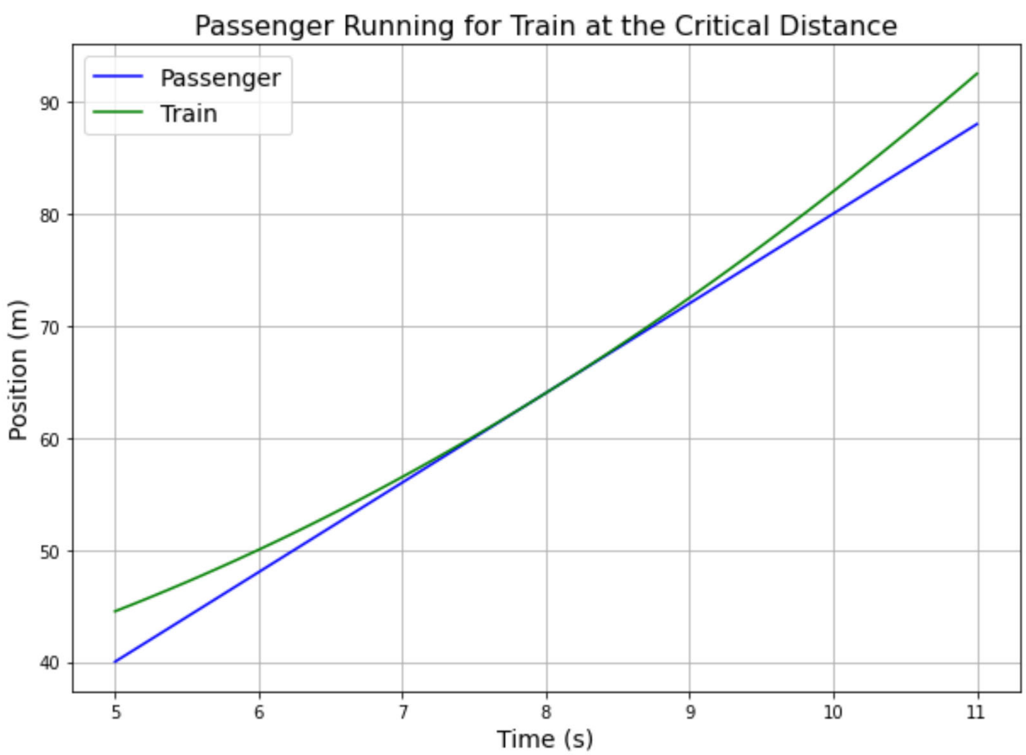
```
time = np.linspace(5.,11.)
xpassenger = xpassenger0 + vpassenger0*time + 0.5*apassenger0*time**2
xtrain = xtrain0 + vtrain0*time + 0.5*atrain0*time**2

plt.rcParams["figure.figsize"] = [10,7]
plt.plot(time, xpassenger, 'b', time, xtrain, 'g')

plt.title('Passenger Running for Train: Zoom', fontsize=16)
plt.xlabel('Time (s)', fontsize=14)
plt.ylabel('Position (m)', fontsize=14)
plt.legend(["Passenger", "Train"], loc ="upper left", fontsize=14)
plt.grid(True)
plt.show()
```



Since the critical distance was calculated to be $d_{critical}$ = 32 m, plot the curves for this as the initial distance from the passenger to the train to confirm that it results in a single intersection (actually, the curves are tangent, but both at the same distance at the same time).

In [18]:

```
#Define the critical distance
xtraincritical = +32.0
vtrain0 = +0.0
atrain0 = +1.0

time = np.linspace(5.,11.)
xpassenger = xpassenger0 + vpassenger0*time + 0.5*apassenger0*time**2

#change the initial position of the train
xtrain = xtraincritical + vtrain0*time + 0.5*atrain0*time**2

plt.rcParams["figure.figsize"] = [10,7]
plt.plot(time, xpassenger, 'b', time, xtrain, 'g')

plt.title('Passenger Running for Train at the Critical Distance', fontsize=16)
plt.xlabel('Time (s)', fontsize=14)
plt.ylabel('Position (m)', fontsize=14)
plt.legend(["Passenger", "Train"], loc ="upper left", fontsize=14)
plt.grid(True)
plt.show()
```



Thus the passenger has one opportunity to catch the train when she starts $d_{critical}$ = 32 m behind it. Now confirm the speed of the train ... the passenger's speed is a constant 8 m/s. Since the curves are parallel at the point where they overlap, the train should be at the same speed.

In [ ]: