HW3—Weather

10 points

Assignment: Write a program that graphically displays the current wind strength and direction in Newton, as explained below.

Due: Wednesday 9/18, 5PM

Turn in: Submit the source files for your programs to Blackboard Vista.

On BBV, you'll find a Python module called "bcWeather.py". Get a copy of it, and put it into a folder where you'll write HW#3. If you *`import bcWeather`*, you'll have several functions available to you:

`bcWeather.getWindSpeed()`	Gets the wind speed, mph
`bcWeather.getWindDirection()`	Gets the direction that the wind is coming from, in degrees (North=0°, East=90°, etc.)
`bcWeather.getTemperature()`	Gets the temperature, in degrees Fahrenheit
`bcWeather.getPressure()`	Gets the barometric pressure, in inches of mercury (normally around 30)
`bcWeather.getHumidity()`	Gets the relative humidity, in %
`bcWeather.getLastUpdate()`	Gets the last date/time the weather station was data was updated, as a text string

Of these, you'll only need `getWindDirection()` and `getWindSpeed()`.

The weather data really is correct, as measured by the weather station in Newton MA (you can view the data directly at http://www.weatherlink.com/user/gulottacomm if you wish). However, I did add a small random jitter to the wind info, to make the graphical display more interesting.

Since this homework is really about writing functions and drawing in the TurtleWorld, I'll guide you through the functions that you should write. Pick meaningful names for your functions.

1. First, write a function that simply displays an arrow pointing upwards, as shown here. Notice that the turtle ends up looking to the right, just like he started.

Test this function! Don't proceed until this one works.

2. Modify the function above to add one parameter: the length of the arrow. Again, test this before proceeding. Try calling your function several times with different lengths to test it.

3. Write a second function, that gets the current wind speed and direction into variables. From this second function, call the first function using `20+20*windSpeed` as the arrow length. Before calling the function, rotate your turtle by the value of the current wind direction. Then call your arrow function, then rotate the turtle back to his starting direction. (You can reduce that second 20 if the wind around here picks up.)

4. Write a third function that does the following repeatedly: print the wind speed and direction, and call the function from step 3 above, then call `time.sleep(1)` to pause for one second. Repeat this a large number of times, 10,000 or more. Also: call `window.clear()` inside your loop to erase all previously drawn arrows.

Your finished program should call this last function.