**Jonathan Klein: arrayProj PROJECT OUTLINE**

INITIALIZATION:

* Began project by initializing three phrases as string that I want to use (which all happened to be physics-related)
* Then created a single method to convert each string to an array, each character of the string as a new element
  + Used this method to convert each phrase to an array
* Then organized all of my arrays in “phrase database”, a matrix that contained of the other phrase arrays
* Initialized scanner object to take in user input in the future
* Created variables to store user input and user score, as well as a variable to keep track of round, which I used to index the phrase database matrix
* Finally, I made a “guesses” array which would keep track of all guesses that the user could guess
  + Length 26 because user can only guess 26 different unique characters

MAIN LOOP:

* Created while loop with condition to run dependent on the round number
  + Did this because I wanted the program to terminate after all phrases (rounds) were completed
* Began by taking in user input to decide whether the user wanted to spin, solve, or quit
  + Used a switch statement to satisfy this

QUIT:

* Did this first because it was the easiest
* If user inputted “q”, the round number would be set to 4, which would exceed the while loop condition and would thus break out of the loop, ending the program gracefully

SPIN:

* Began by using Math.random() to generate a random number for the spin value that the user could be rewarded
* At this point I realized that I was going to need to perform many linear searches so I created an arrayContains() method to simplify the situation
  + Simply iterates over a certain array and checks if any element of the array equals the given target; returns a boolean based on this condition
* Then stored the user’s letter guess in the guess array (assuming it was a valid letter and it was not one already guessed)
* Then printed the current phrase array, by iteration
  + Only printed the letters that the user had successfully guessed
  + Otherwise, an underscore took that character’s place
* After this, I created an allCorrect() method which I used to check if the user had guessed the entire phrase correctly letter by letter
  + Simply checked that every character in the current phrase had already been in the guesses array, signifying that each letter had been correctly guessed by the user at some point
* Once the entire phrase had been guessed, the guesses array was reset and the phrase was changed to the next phrase

SOLVE:

* Began by creating a method named compareArrays(), which checked if two arrays were exactly the same
  + Simply iterates over one array and compares the element at one index to the element of the other array at that same index.
* Used this to compare the user’s input to the current phrase
* If the user guessed correctly, the round would proceed
  + Guesses array reset once again
* If the user failed to guess the phrase correctly, all points would be lost and the current round would continue