Day 6: Let's Review | HackerRank

Terms you'll find helpful in completing today's challenge are outlined below, along with sample Java code (where appropriate).

Strings and Characters

As we've mentioned previously, a String is a sequence of characters. In the same way that words inside double quotes signify a String, a single letter inside single quotes signifies a character. Each character has an ASCII value associated with it, which is essentially a numeric identifier. The code below creates a char variable with the value *c*, and then prints its ASCII value.

```
char myChar = 'c'; // create char c
System.out.println("The ASCII value of " + myChar + " is: " + (int) myChar);
```

Output:

```
The ASCII value of c is: 99
```

Observe the `(int)` before the variable name in the code above. This is called *explicit casting*, which is a method of representing one thing as another. Putting a data type inside parentheses right before a variable is essentially saying: "The next thing after this should be represented as this data type". Casting only works for certain types of relationships, such as between primitives or objects that inherit from another class.

To break a String down into its component characters, you can use the String.toCharArray method. For example, this code:

```
String myString = "This is String example.";
char[] myCharArray = myString.toCharArray();
for(int i = 0; i < myString.length(); i++){
   // Print each sequential character on the same line
   System.out.print(myCharArray[i]);
}
// Print a newline
System.out.println();
```

produces this output:

```
This is String example.
```

Notice that we were able to simulate printing *myString* by instead printing each individual character in the character array, *myCharArray*, created from *myString*.

Solve Problem