Project 2 -- Equipping Characters

Jumptastic Games has decided to develop a new [role-playing game (Links to an external site.)](https://en.wikipedia.org/wiki/Role-playing_game) in which characters will be able to enhance their basic abilities by "wearing" different items. They have asked you to help by designing and implementing a model to help keep track of this. In their game, characters can wear four different types of clothing:

* **Head gear:** These items go on the character’s head (hats/helmets/visors) and are only useful for defense.
* **Footwear:** These items go on the character’s feet (boots/sneakers/hoverboard) and are only used for attack. Since character's have 2 feet, they can have 2 of these items.
* **Hand gear:** These items go on the character’s fingers/hands and can be for either attack or defense. Since character's have 10 fingers, they can have 10 of these items.
* **Jewelry:** These items go around the character's neck and can be used for either attack or defense. There is no limit to the number of these items that the character can have.

The goal of the model is to be able to keep track of all of the items that a character is wearing at any given time during the game. it should also provide the following functionality:

1. In addition to current hit points, characters begin with a basic attack power and defensive strength (represented as numerical values). As they go through the game, they can pick up new items based on how many items of a particular type that they can wear.
2. The attack or defensive power of the items that a character is wearing *temporarily* adds (or subtracts in the case of a cursed item) to the players attack power and defensive strength.
3. When describing what a character is wearing, the names of items of the same type are combined. The new grammatically correct name is the full name of the first item, and the adjective of the others. For example (order does not matter):
   * To combine "Sandals of Speed" and "Hoverboard of Heft", you would get "Hoverboard of Heft and Speed".
   * To combine "Gloves of Holding", Ring of Invisibility", "Ring of Electrical Shock", you would get "Gloves of Holding, Invisibility, and Electrical Shock".
4. Some items wear out with each use and thus their benefit decreases each time a player uses them. Characters that continue to wear and use worn-out items find that these items decrease their basic attack power and defensive strength.

Rather than creating a whole game, create a **driver class** (as opposed to a full MVC application) that dresses two (2) characters and pits them in a battle. To do this:

* Populate a chest of gear gets populated with a minimum of 4 items of headgear, 8 items of footwear, 15 items of hand gear, and 15 items of jewelry
* "Dress" each character with a random 20 items from the chest of gear (duplicates allowed). When items are selected the character can chose to wear them based on:
  + Whether the item can be combined with what they are already wearing. If they cannot be combined with what they are already wearing, the character can choose to wear the new item discarding an item they are already wearing. Print out any items that are discarded along what item took its place.
  + Whether the item would improve their basic attack power or defense strength. Characters always choose a higher attack strength over a higher defense strength
* Print out each character in the battle along with what they are wearing and their attack and defense strength
* Predict which of the two characters would win in a battle by calculating the number of rounds a character would survive if attacked by the other player. For example, suppose that Alice and Bob have been properly equipped from the chest of gear. First calculate the potential damage that each player can inflict on the other in each round:

​ damageAlice = attackBob - defenseAlice

​ damageBob = attackAlice - defenseBob

Assuming that the calculated potential damage is inflicted each round, calculate the number of rounds each player would last based on their hit points. Then predict the winner as the player that would last the most number of rounds by printing this information to the screen. If the players would last the same number of rounds, you should predict a tie.

* Provide the option for the characters to play a rematch in which case they should be allowed to "dress" in a new selection of items. This should be the only keyboard input required to use your driver program.