

Loot Bag Organizer Challenge

Challenge Preamble

This challenge is designed to help you practice more advanced Python concepts such as:

- string parsing,
- file reading,
- regular expressions,
- nested data structures, and
- arithmetic conversions.

You will develop a program that processes a text file containing a list of loot items collected by a player. Each item has a value expressed in a mix of **gold**, **silver**, and **copper** coins, where each type of coin has a specific exchange rate. The program will read each loot entry, interpret the coin values, and calculate the total value of **all items** obtained during the adventure.

Part 1 – Parsing and Value Conversion: Theory

In this program, each line of the input text file represents a treasure found by the adventurer. Each treasure is described by its name and its monetary value expressed using a combination of coins: gold, silver, and copper.

The three types of coins have the following exchange rate:

- 1 gold coin = 10 silver coins
- 1 silver coin = 10 copper coins
- Therefore, 1 gold coin = 100 copper coins

The task is to design a way to interpret these values numerically. Since every line in the loot file contains both a name and one or more coin amounts, you must first separate the name of the item from its value. The value itself is a string that includes both numbers and coin types, such as '2 gold, 5 silver'.

A good approach to extract the numerical values for each coin type is to use string searching techniques. You can use methods that detect specific words like 'gold', 'silver', and 'copper' and extract the numbers that precede them. Alternatively, a more flexible approach is to use regular expressions (regex) to find every occurrence of a number followed by a coin type. This allows your program to handle cases where not all coin types are present in a line.

Once the values have been identified, they should be converted into a single unit for easier calculation. Copper is a convenient base unit since every higher coin can be expressed as a multiple of copper. By converting each amount to copper, you can perform all additions and comparisons without worrying about mixed units. Later, you can convert the total back into gold, silver, and copper for a readable result.

Because the same type of item may appear more than once, a data structure is needed to group identical items together. A dictionary is perfect for this, where each key represents the item name and the value is a collection that stores both the count of how many times it appears and the list of coin values it was found with.

Part 2 – Loot Log Parsing: Code Walk-through

The overall program can be separated into smaller, reusable parts by defining functions. Each function should handle a single specific task, making the code easier to read, maintain, and test.

1. First, you will need a function to convert between gold, silver, and copper. This function should receive the amount of each type of coin and calculate its equivalent value in copper. This conversion allows all computations to be done in a single base unit.
2. A second function can be created to perform the opposite operation. After summing up all the values in copper, this function will convert the final amount back into gold, silver, and copper. This makes the result easier to read and display to the player.
3. A third part of the code should focus on reading and interpreting each line of the loot log. When the file is opened and read line by line, the first word or phrase on the line represents the item name, and the following words describe its value. The goal is to extract the numbers associated with each coin type. A regular expression can be used to identify these patterns and return a tuple representing the gold, silver, and copper values found.
4. Once the coin values have been extracted, they can be stored in a dictionary. Each key in the dictionary will be the item name, while the value will contain the number of times the item was collected and the list of all its individual values. Each new item found will either be added as a new key or appended to an existing one.
5. The next step involves calculating the total value of all loot items combined. This can be done by looping through each item in the dictionary, converting every stored gold, silver, and copper triplet into copper, and summing them up to obtain the overall value. The result can then be converted back into gold, silver, and copper for display.
6. The result value should give you:

1447 gold, 1 silver and 2 copper coins.

Part 3 – Bonus Challenge: Extended Analysis

Now that your program can calculate both the total values, you can extend it with an additional feature.

Imagine that the buyer of your loot is only interested in buying the single best valued item from each item. That means that if you collected 15 “Enchanted Thread” each of the following values :

[(1, 3, 1), (1, 4, 3), (1, 3, 9), (2, 2, 5), (0, 0, 1), (4, 5, 2), (2, 7, 9), (2, 1, 4), (0, 2, 7), (0, 8, 6), (0, 0, 6), **(4, 9, 5)**, (0, 4, 5), (0, 9, 4), (0, 2, 8)]

The highest value one would be the one that is worth 4 gold, 9 silver and 5 copper coins.

Do the sum of all the highest valued items. The result for that challenge should be :

220 gold, 0 silver and 2 copper coins.