Coin Counter

Project Explanation

Suppose you have a coin counting machine that provides input to your program. A bunch of mixed coins are placed into a hopper and are sorted one-by-one.

The machine sends a string of characters to your program. When a penny is sorted, the character 'p' is concatenated with the string. When a nickel is sorted 'n' is concatenated with the string. Other coins: dimes, quarters, half-dollars all work the same way: the first letter of the name identifies the coin. Your program will read the string using the input() function and calculate the number of each type of coin, the value of each type of coin and the total monetary value represented.

Coin values are: penny = $0.01, nickel= $0.05, dime= $0.10, quarter= $0.25 and half-dollar= $0.50. Your output should have a neat report format as shown in the example below.

Total amount calculation example:
If the string is: "dqpppnqhppqd" the monetary value will be
0.10+0.25+0.01+0.01+0.01+0.05+0.25+0.50+0.01+0.01+0.25+0.10 = $1.55

Of course, we are not going to connect your program to a machine. A user at the console will type in strings. Here are some example program runs …

Here the input string is: dqpppnqhppqdpp …

```
Program to count coins and calculate values

Enter coin string: dqpppnqhppqdpp

====================================
        Coin Counter Report
====================================

Coin           Value  Number  Amount
====           =====  ======  ======
Pennies        $0.01     7     $ 0.07
Nickels        $0.05     1     $ 0.05
Dimes          $0.10     2     $ 0.20
Quarters       $0.25     3     $ 0.75
Half-dollars   $0.50     1     $ 0.50
                Total amount: $ 1.57
```

Coin Counter

Here, the input string is: dqpppnqhppqdppdqpppnqhppqdpp …

```
Program to count coins and calculate values

Enter coin string: dqpppnqhppqdppdqpppnqhppqdpp

===================================
        Coin Counter Report
===================================

Coin           Value  Number  Amount
====           =====  ======  ======
Pennies        $0.01    14    $ 0.14
Nickels        $0.05    2     $ 0.10
Dimes          $0.10    4     $ 0.40
Quarters       $0.25    6     $ 1.50
Half-dollars   $0.50    2     $ 1.00
               Total amount: $ 3.14
```

Use the input strings and the results shown above to test your own program.

This programming problem involves lists, loops, f-strings, and accumulation. There are many Pythonic ways to solve the problem. It can be solved without using **`if … elif`** but that's not a requirement. All solutions should traverse the user response string without using **`range()`** or **`len()`**. The problem can be solved in as little as 17 lines of source code (neglecting comments) if a suitable strategy is used. In my version, six of the 17 lines are print statements.

TIPS:

1. There are four aspects of each coin: the character code used to represent each coin, the name of each coin, the value of each coin, and the number of each coin. If you use a separate variable for each of these, you will have 20 variables … which is quite a large number for a simple program. Think about how you would use the data structures to group these aspects to minimize the number of variables you create. It's ok to use several lists and index into them. It's ok to use the **`index()`** method to locate an item in a list. But avoid using **`range()`** and **`len()`**.
2. Counting the characters
    a. One way would be to use a for loop to traverse the string …
    ```
    for c in s:
      if c=='p':
        etc.
    ```
    b. Another way would be to use the string method: **`count()`**
3. Once you know the number of each character (i.e. coin) in the input string, you can compute the dollar amount of each coin. And once you know the dollar amount for each coin, you can compute the total.
4. The final thing you should do is work on getting the report formatted into columns. By the final thing, I mean save it for last … after you have figured out how to do the calculations correctly. Formatted print takes planning, testing, and tweaking. Study the f-string document. Study the report format shown above. How is each column justified? How many characters are used for each column? How are the "$" characters placed?