# Problem 1 – Uncle Scrooge’s Bag +

Scrooge McDuck has worked his way up the financial ladder from humble immigrant roots. As a young boy, he took up a job polishing and shining boots in his native Glasgow. His turning point was when a ditchdigger paid him with 1875 rotten coins, which was useless. Enraged, Scrooge vowed to never be taken advantage of again.

For that reason he knows all the time how much money he has in his bag. Write a **function** that will help Uncle Scrooge to find the exact amount of **gold**, **silver** and **bronze** in his bag.

Think of the input **array** as Uncle Scrooge’s bag. Each element is an **item** from the bag. The **type** of the element must be **а coin** and the amount must be a **positive number** with **0 fraction** in order to be a valid coin. Every **100** coins make **1** gold, every **10** coins make **1** silver and **1** coin is **1** bronze.

For example a **valid** coin elements are the strings “coin 111.00” and “COIN 1001”. “Coin 111.23” on the other side is **not a valid** string because 111.23 is **not an integer value**.

### Input

### The input data will be received as an array. It contains a couple of strings. Each string is containing a type and a value separated by a whitespace. The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

**The output consists of three lines. The** first **line is for the** gold**, the** second **line is for the** silver **and the** third **line is for the** bronze**.** See the examples for better understanding.

### Constraints

* The amount of coins will be a number no greater than **100000**.
* Time limit: 0.3 sec. Memory limit: 16 MB.

**function** *solve*(input) {  
  
 **let** sum = 0;  
 **let** coins = 0,  
 bronzeCoins = 0,  
 silverCoins = 0,  
 goldCoins = 0;  
  
 **for** (**let** i = 0; i < input.length; i++) {  
  
 **let** tokens = input[i].**split**(**' '**);  
 **let** type = tokens[0].toLowerCase();  
 **let** value = tokens[1];  
  
 **if**(value > 0 && *isInt*(value) && type === **'coin'**){  
 coins += parseInt(value);  
 }  
 }  
  
 goldCoins += ***Math***.floor(coins/100);  
 coins = coins - goldCoins\*100;  
 silverCoins += ***Math***.floor(coins/10);  
 bronzeCoins += coins - silverCoins\*10;  
  
 **console**.log(**'gold : '** + goldCoins);  
 **console**.log(**'silver : '** + silverCoins);  
 **console**.log(**'bronze : '** + bronzeCoins);  
  
 **function** *isInt*(n) {  
 **return** n% 1 === 0;  
 }  
}  
  
*solve*([  
 **'coin 1'**,**'coin 2'**, **'coin 5'**, **'coin 10'**, **'coin 20'**, **'coin 50'**, **'coin 100'**, **'coin 200'**, **'coin 500'**,**'cigars 1'**]  
);

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| [**'coin 1'**,**'coin 2'**, **'coin 5'**, **'coin 10'**, **'coin 20'**, **'coin 50'**, **'coin 100'**, **'coin 200'**, **'coin 500'**,**'cigars 1'**] | gold : 8  silver : 8  bronze : 8 |
| [**'coin one'**, **'coin two'**, **'coin five'**, **'coin ten'**, **'coin twenty'**, **'coin fifty'**, **'coin hundred'**, **'cigars 1'**] | gold : 0  silver : 0  bronze : 0 |
| [**'coin 1'**, **'coin two'**, **'coin 5'**, **'coin 10.50'**, **'coin 20'**, **'coin 50'**, **'coin hundred'**, **'cigars 1'**] | gold : 0  silver : 7  bronze : 6 |