

题目描述

记账本上记录了若干条多国货币金额，需要转换成人民币分（fen），汇总后输出。
每行记录一条金额，金额带有货币单位，格式为数字+单位，可能是单独元，或者单独分，或者元与分的组合。
要求将这些货币全部换算成人民币分（fen）后进行汇总，汇总结果仅保留整数，小数部分舍弃。
元和分的换算关系都是1:100，如下：

1CNY=100fen（1元=100分）
1HKD=100cents（1港元=100港分）
1JPY=100sen（1日元=100仙）
1EUR=100eurocents（1欧元=100欧分）
1GBP=100pence（1英镑=100便士）

汇率表如下：

CNY	JPY	HKD	EUR	GBP
100	1825	123	14	12

即：100CNY = 1825JPY = 123HKD = 14EUR = 12GBP

输入描述

第一行输入为N，N表示记录数。0<N<100
之后N行，每行表示一条货币记录，且该行只会是一种货币。

输出描述

将每行货币转换成人民币分（fen）后汇总求和，只保留整数部分。
输出格式只有整数数字，不带小数，不带单位。

用例

输入	1 100CNY
输出	10000
说明	100CNY转换后是10000fen，所以输出结果为10000

输入	1 3000fen
输出	3000
说明	3000fen，结果就是3000

输入	1 123HKD
输出	10000
说明	HKD与CNY的汇率关系是123:100，所以换算后，输出结果为10000

输入	2 20CNY53fen 53HKD87cents
输出	6432
说明	20元53分 + 53港元87港分，换算成人民币分后汇总，为6432

题目解析

纯逻辑题。

本题唯一的难度应该在于如何解析输入字符串中的金额和单位，这里最好是使用正则表达式：

题目解析

纯逻辑题。

本题唯一的难度应该在于如何解析输入字符串中的金额和单位，这里最好是使用正则表达式：

```
> const regExp =  
  /(\d+)((CNY)|(JPY)|(HKD)|(EUR)|(GBP)|(fen)|(cents)|(sen)|(eurocents)|(pence))/g;  
: undefined  
> regExp.exec('20CNY53fen')  
: (13) ['20CNY', '20', 'CNY', 'CNY', undefined, undefined, undefined, undefined, undefined, undefi  
ned, undefined, undefined, undefined, index: 0, input: '20CNY53fen', groups: undefined]  
> regExp.exec('20CNY53fen')  
: (13) ['53fen', '53', 'fen', undefined, undefined, undefined, undefined, undefined, undefined, 'fen', undefi  
ned, undefined, undefined, undefined, index: 5, input: '20CNY53fen', groups: undefined]  
> |
```

剩下的就是将不同单位转换成人民币分的汇率计算工作了，如下源码中exchange记录的不同单位转成人民币分的汇率。

JavaScript算法源码

```
1  /* JavaScript Node ACM模式 控制台输入获取 */  
2  const readline = require("readline");  
3  
4  const rl = readline.createInterface({  
5    input: process.stdin,  
6    output: process.stdout,  
7  });  
8  
9  const lines = [];  
10 let n;  
11 rl.on("line", (line) => {  
12   lines.push(line);  
13  
14   if (lines.length === 1) {  
15     n = lines[0] - 0;  
16   }  
17  
18   if (n && lines.length === n + 1) {  
19     lines.shift();  
20     console.log(getResult(lines));  
21     lines.length = 0;  
22   }  
23 });  
24  
25 function getResult(arr) {  
26   const regExp =  
27     /(\d+)((CNY)|(JPY)|(HKD)|(EUR)|(GBP)|(fen)|(cents)|(sen)|(eurocents)|(pence))/g;  
28  
29   const exchange = {  
30     CNY: 100,  
31     JPY: (100 / 1825) * 100,  
32     HKD: (100 / 123) * 100,  
33     EUR: (100 / 14) * 100,  
34     GBP: (100 / 12) * 100,  
35     fen: 1,  
36     cents: 100 / 123,  
37     sen: 100 / 1825,  
38     eurocents: 100 / 14,  
39     pence: 100 / 12,  
40   };  
41  
42   let ans = 0;  
43   arr.forEach((str) => {  
44     while (true) {  
45       const res = regExp.exec(str);  
46       if (!res) break;  
47       const [, amount, unit] = res;  
48  
49       ans += amount * exchange[unit];  
50     }  
51   });  
52  
53   return Math.floor(ans);  
54 }
```

Java算法源码

```
1  import java.util.*;  
2  import java.util.regex.Matcher;  
3  import java.util.regex.Pattern;  
4  
5  public class Main {  
6    public static void main(String[] args) {  
7      Scanner sc = new Scanner(System.in);  
8      int n = sc.nextInt();  
9  
10     String[] arr = new String[n];  
11     for (int i = 0; i < n; i++) {  
12       arr[i] = sc.next();  
13     }  
14   }
```

Java算法源码

```
1 import java.util.*;
2 import java.util.regex.Matcher;
3 import java.util.regex.Pattern;
4
5 public class Main {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int n = sc.nextInt();
9
10        String[] arr = new String[n];
11        for (int i = 0; i < n; i++) {
12            arr[i] = sc.next();
13        }
14
15        System.out.println(getResult(arr));
16    }
17
18    public static int getResult(String[] arr) {
19        String reg = "\\d+((CNY)|(JPY)|(HKD)|(EUR)|(GBP)|(fen)|(cents)|(sen)|(eurocents)|(pence))";
20
21        HashMap<String, Double> exchange = new HashMap<>();
22        exchange.put("CNY", 100.0);
23        exchange.put("JPY", 100.0 / 1825 * 100);
24        exchange.put("HKD", 100.0 / 123 * 100);
25        exchange.put("EUR", 100.0 / 14 * 100);
26        exchange.put("GBP", 100.0 / 12 * 100);
27        exchange.put("fen", 1.0);
28        exchange.put("cents", 100.0 / 123);
29        exchange.put("sen", 100.0 / 1825);
30        exchange.put("eurocents", 100.0 / 14);
31        exchange.put("pence", 100.0 / 12);
32
33        double ans = 0.0;
34        Pattern p = Pattern.compile(reg);
35
36        for (String s : arr) {
37            Matcher m = p.matcher(s);
38            while (true) {
39                if (m.find()) {
40                    Double amount = Double.parseDouble(m.group(1));
41                    String unit = m.group(2);
42                    ans += amount * exchange.get(unit);
43                } else {
44                    break;
45                }
46            }
47        }
48
49        return (int) ans;
50    }
51 }
52 }
```

Python算法源码

```
1 import math
2 import re
3
4 # 输入获取
5 n = int(input())
6 arr = [input() for i in range(n)]
7
8
9 # 算法入口
10 def getResult(arr):
11     pattern = r"\d+((CNY)|(JPY)|(HKD)|(EUR)|(GBP)|(fen)|(cents)|(sen)|(eurocents)|(pence))"
12
13     exchage = {
14         "CNY": 100,
15         "JPY": 100 / 1825 * 100,
16         "HKD": 100 / 123 * 100,
17         "EUR": 100 / 14 * 100,
18         "GBP": 100 / 12 * 100,
19         "fen": 1,
20         "cents": 100 / 123,
21         "sen": 100 / 1825,
22         "eurocents": 100 / 14,
23         "pence": 100 / 12
24     }
25
26     ans = 0
27
28     for s in arr:
29         result = re.findall(pattern, s)
30         for item in result:
31             amount = item[0]
32             unit = item[1]
33             ans += int(amount) * exchage[unit]
34
35     return math.floor(ans)
```



伏城之外 已关注

👍 0 🗨 4



📁 2 📁

专栏目录

已订阅

Python算法源码

```
1 import math
2 import re
3
4 # 输入获取
5 n = int(input())
6 arr = [input() for i in range(n)]
7
8
9 # 算法入口
10 def getResult(arr):
11     pattern = r"(\d+)((CNY)|(JPY)|(HKD)|(EUR)|(GBP)|(fen)|(cents)|(sen)|(eurocents)|(pence))"
12
13     exchange = {
14         "CNY": 100,
15         "JPY": 100 / 1825 * 100,
16         "HKD": 100 / 123 * 100,
17         "EUR": 100 / 14 * 100,
18         "GBP": 100 / 12 * 100,
19         "fen": 1,
20         "cents": 100 / 123,
21         "sen": 100 / 1825,
22         "eurocents": 100 / 14,
23         "pence": 100 / 12
24     }
25
26     ans = 0
27
28     for s in arr:
29         result = re.findall(pattern, s)
30         for item in result:
31             amount = item[0]
32             unit = item[1]
33             ans += int(amount) * exchange[unit]
34
35     return math.floor(ans)
36
37
38 # 算法调用
39 print(getResult(arr))
```

[复制](#)