

OJ 147

Dollars

Prof. Edson Alves – UnB/FGA

New Zealand currency consists of \$100, \$50, \$20, \$10, and \$5 notes and \$2, \$1, 50c, 20c, 10c and 5c coins. Write a program that will determine, for any given amount, in how many ways that amount may be made up. Changing the order of listing does not increase the count. Thus 20c may be made up in 4 ways: $1 \times 20c$, $2 \times 10c$, $10c + 2 \times 5c$, and $4 \times 5c$.

Input

Input will consist of a series of real numbers no greater than \$300.00 each on a separate line. Each amount will be valid, that is will be a multiple of 5c. The file will be terminated by a line containing zero (0.00).

Output

Output will consist of a line for each of the amounts in the input, each line consisting of the amount of money (with two decimal places and right justified in a field of width 6), followed by the number of ways in which that amount may be made up, right justified in a field of width 17.

Exemplo de entradas e saídas

Sample Input

0.20
2.00
0.00

Sample Output

0.20 4
2.00 293

Solução $O(NM)$

- Este problema é uma variante do problema do troco, onde se deseja obter o número de maneiras distintas de se dar um mesmo troco m utilizando-se as moedas disponíveis
- Deste modo, é preciso adaptar o algoritmo de programação dinâmica para obter este resultado
- Importante notar que a ordem não é levada em consideração, de modo que $\{0.25, 0.50, 0.25\}$ e $\{0.50, 0.25, 0.25\}$ são consideradas uma mesma maneira de dar o troco de \$1
- Seja $dp(m, i)$ o número de maneiras distintas de se dar o troco m utilizando-se apenas as i menores moedas
- São dois casos-base: $dp(0, 0) = 1$ e $dp(m, 0) = 0$, se $m > 0$

Solução $O(NM)$

- São duas transições possíveis: utilizar a i -ésima menor moeda ou não, sendo que ambas alternativas devem ser totalizadas
- Assim,

$$dp(m, i) = dp(m - c_i, i) + dp(m, i - 1),$$

se $c_i \leq m$, ou $dp(m, i) = dp(m, i - 1)$, caso contrário

- Se as moedas forem processadas em ordem crescente, é possível deixar ímplicita a segunda dimensão do estado, reduzindo o uso de memória e simplificando a implementação
- A tabela dos estados pode ser preenchida em $O(NM)$, e após este preenchimento cada caso de teste pode ser respondido em $O(1)$, onde M é o maior troco possível e N o número de moedas distintas

Solução $O(NM)$

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4 using ll = long long;
5
6 const int MAX { 30010 };
7 const vector<int> cs { 10000, 5000, 2000, 1000, 500, 200, 100, 50, 20, 10, 5 };
8
9 ll st[MAX];
10
11 void precomp()
12 {
13     st[0] = 1;
14
15     for (auto c : cs)
16         for (int m = c; m < MAX; m += 5)
17             st[m] += st[m - c];
18 }
```

Solução $O(NM)$

```
20 int main()
21 {
22     precomp();
23
24     int d, c;
25
26     while (scanf("%d.%d", &d, &c) == 2 and d + c)
27         printf("%3d.%02d%17lld\n", d, c, st[100*d + c]);
28
29     return 0;
30 }
```