

O problema do troco Suponha um país que tem uma certa configuração de moedas. No Brasil: {1, 5, 10, 25, 50 e 100} centavos. Nos EUA: {1, 5, 10, 25, 50}. Em qualquer um desses países, máquinas automáticas de venda de alguma coisa precisam calcular e entregar em eventual troco ao freguês. A combinação de moedas que totaliza o troco não é única. Por exemplo, em uma país cujas moedas são {1, 6, 10}, o troco de 19 centavos pode ser 10,6,1,1,1 ou 6,6,6,1 ou outras combinações ainda, por exemplo, 19 moedas de 1.

Usa-se um algoritmo guloso aqui. O resultado quase sempre é o ótimo, embora em algumas vezes, ele entregue um resultado pior. O critério de otimização é a menor quantidade de moedas (para preservar a máquina e minimizar as visitas de reposição de numerário).

Eis o algoritmo:

```
1: função troco (moedas[n], valor : inteiro)
2: ordenar as moedas em ordem decrescente
3: enquanto valor > 0
4:   i ← 1
5:   enquanto i ≤ tamanho(moeda)
6:     enquanto valor ≥ moeda[i]
7:       escreva (moeda[i])
8:       valor ← valor - moeda[i]
9:     fim{enquanto}
10:    i ← i + 1
11:  fim{enquanto}
12: fim{função}
13: fimfuncao
```

Para testar:

```
1 5 10 25 50 100 troco 98 é: 50,25,10,10,1,1,1
1 5 10 25 50 100 troco 51 é: 50,1
1 5 10 25 50 100 troco 77 é: 50,25,1,1
1 2 5 35 50 troco 59 é: 50,5,2,2
1 7 11 20 66 troco 78 é: 66,11,1
```

PL998c 11/02/2019 12:36:14.3

nome	código	MOEDAS	VALOR	RESPOSTA
	1	11 25 35 7 1	75	