# 换零钱

## 1 题目描述

考虑仅用1分、5分、10分、25分和50分这5种硬币支付某一个给定的金额。例如需要支付11分钱，有一个1分和一个10分、一个1分和一个5分、六个1分和一个5分、十一个1分这4种方式。请写一个程序，计算一个给定的金额有几种支付方式。注：假定支付0元有1种方式。

### 1.1 输入描述:

输入包含多组数据。每组数据包含一个正整数n（1≤n≤10000），即需要支付的金额。

### 1.2 输出描述:

对应每一组数据，输出一个正整数，表示替换方式的种数。

### 1.3 输入例子:

11

26

### 1.4　输出例子:

4

13

## 2 解题思路

假设硬币的种类数组t={1,5,10,25,50}，按大小排序。m表示选择有0~m种硬币可以选择，面值是t[0]、···、t[m-1]。要换的钱的数目是n。本题可以使用动态规划算法解决。

### 2.1 递归方式

假设有n钱待找零，当前可以供选择的方式为m种，f(n,m)表示共的待找零方案，则有递推公式：

当n=0，f(n,m)=1，表示已经找零完毕，再找0元只有一种方案。

当n<0表示这种方案找零不合理，不能完成找零操作，而m0说明找零还没有完成，但是已经没有可以供选择的硬币了。所以f(n,m)=0。

对于可以找零，并且还有硬币选择的情况找零有两种方案。第一种是：选择一个可以选择的最大面值的，剩下的钱再进行找零操作，同时硬币种类的选择方案没有变化，即为：f(n-t[m-1],m)。第二种是：现在和以后都不选择本次可以选择的最大的硬币面值，然后再进行找零操作。即f(n,m-1)。

### 2.2 非递归方式

假设有n钱待找零，当前可以供选择的方式为m种，创建一个长度为n+1的数组r，r[i]表示找零为i的找零方法为r[i]。初始时r的第一个元素为1，其它元素都为0，即r[0]=1，r[i≠0]=0。

步骤一、因为硬币的面值都按大小排序，从最小的面值开始选择，先选择最小的一个t[0]。对于找零为大于0的情况只有从t[0]开始才可能有找零的情况。对于i≥t[0]有r[i]=r[i]+r[i-t[0]]。这是只有一种硬币可以选择的情况。

步骤二、当有2种硬币可以选择，在步骤一已经求出了只有一种硬币可供选择的情况，现在可以选择第二种硬币，那么只有找零数i≥t[0]时才可以选择第二种硬币，所以有r[i]=r[i]+r[i-t[1]]。

同理可以求得有3、4、···。具体实现详见代码。