Due: Thursday, July ~~5th~~ 12th, 6PM
Worth: 4%.
submissions: Use CDF submit command. Assignment name is PA1.
File name: PA1sol.c, PA1sol.cc, PA1sol.java or PA1sol.py
Any submission that:

- does not compile,
- produces a runtime error,
- does not adhere to the precise input/output format,

will not get any marks. Marks are earned based on passing (producing correct answer for) 20 test cases. For each test case that your program produces the correct output in the allocated runtime you get 5% of the assignment's mark.
You are free to write your answers in Python, Java, or C/C++ but be warned that Python programs might be at a disadvantage in terms of speed.
The runtime limit of the problem is there to test if you have implemented the fastest algorithm for the problem and is designed to be tight. A suboptimal algorithm, or a particularly poor implementation is not supposed to pass all the tests. (But if you can't figure out or implement the fastest algorithm feel free to submit a slower implementation; you will get the marks from the easier test cases.)

Question #1: Paying in Change

You are visiting a popular grocery store to buy grocery. Unfortunately, you forgot to bring your credit card and all you have in your pocket is a number of coins. You have to pay x cents for your groceries using these coins. Of course, you might not have the exact change so you can pay more than x cents in which case the cashier will give you back some money. What you want to do is *minimize the total number of coins that change hands*, that is, minimize the number of coins that you give to the cashier plus the number of coins that they give back to you.
You can assume that neither you nor the cashier has any bills so the whole transaction is done using coins and that the cashier has a very big supply of all the canadian coins (1¢, 5¢, 10¢, 25¢, $1 and $2.) Furthermore, you can assume that the cashier is really smart so they will use the minimum number of coins to return your change.
Devise an algorithm that gets x, and the content of your pocket as input and outputs the minimum number of coins that change hands in the optimal way of paying for your groceries.

Input format: You algorithm should read its input from the standard input. The first line of input will have a single number, this will be the integer x the amount of money you have to pay in cents. The second line of input will have 6 numbers, the first is the number of 1¢ coins you have, the second the number of 5¢ coins you have, the third the number of 10¢ coins, then 25¢ coins, then $1 coins you have and finally the sixth and final number is the number of $2 coins you have.

Conditions: The number of coins of each kind that you have is at most 1000 and the number x is at most 10^5. You can assume that you have enough money to pay x cents.

Runtime (tentative): Your algorithm should take no more than 3 seconds on each input on the cdf servers. This is the "cpu time" which you can see by using the "time" command on the cdf machine.

Output format: Your output should be written on the standard output and should have only one line with a single number, the minimum number of coins that change hands.

Sample input and output: Here are 4 sample inputs with correct output for each.

Sample Input	Correct output
7 5 5 5 5 5 5	3
126 100 3 0 1 0 1	4
30 5 0 5 1 0 0	3
99 99 0 0 0 1 0	2

In the second input Here is a description of the second input and its correct. In this example $x = 126$ and we have one hundred 1¢coins, three 5¢coins, a 25¢coin and a 2$ coin. You should give the cashier $2.26 using a 1¢, a 25¢ and a $2 coin and they will give back a $1 coin.
In the last input $x = 99$ and we have ninety nine 1¢coins and one 1$ coin the answer is 2. You should pay the cashier a $1 coin and they will give you back a 1¢ coin. Notice that in this example you *could* pay the exact change with a lot of pennies but that is not the optimal solution.