Assignment 1: Vending Machine

Due: 20:00, Thu 23 Sep 2021 Full marks: 100

Introduction

The objective of this assignment is to learn how to use variables, operators, expressions, and console input/output. You will write a program that determines the number of coins of different denominations (面值) to be returned as change in a vending machine.

Suppose a country issues coins of six different denominations $\$d_1, \$d_2, \$d_3, \$d_4, \$d_5, \$d_6$ (in decreasing order). When you buy items (e.g., drinks) in a vending machine, you insert money into the machine, select an item, and the machine will dispense the item and return the change, if any. Given a particular change amount, a strategy to determine the combination of various coins to be returned is as follows. First, return the maximum number of the highest denomination ($\$d_1$). Then, from the remaining change, return the maximum number of the second highest denomination ($\$d_2$). The process continues for the remaining denominations $\$d_3, \$d_4, \$d_5, \$d_6$. The following shows two examples of returning coins combination.

Example 1. Suppose money inserted is $400 and item price is $57. (Thus change is $343.) Coin denominations are $100, $50, $10, $5, $2, and $1. The following combination of coins will be returned.

Coin Denomination	$100	$50	$10	$5	$2	$1
Number of Coins	3	0	4	0	1	1

Example 2. Suppose money inserted is $100 and item price is $35. (Thus change is $65.) Coin denominations are $48, $26, $12, $6, $2, and $1. The following combination of coins will be returned.

Coin Denomination	$48	$26	$12	$6	$2	$1
Number of Coins	1	0	1	0	2	1

Note that this strategy does not guarantee the fewest number of coins used. E.g., a change of $52 (say, $60 inserted to buy an $8 item) with the coin denominations in Example 2 can be formed optimally by two $26 coins, but this strategy would give one $48 coin and two $2 coins. (Totally three coins.) Besides, it even does not guarantee to find a combination of coins. E.g., a change of $8 (say, $10 inserted to buy a $2 item) with denominations $7, $6, $5, $4, $3, $2 can be formed by one $6 coin and one $2 coin, but this strategy cannot find a combination after returning one $7 coin.

Program Specification

The program first obtains the six coin denominations from the user. Then, it further obtains the amount of money inserted into the vending machine, and the price of the buying item. You can assume that (a) *all user inputs will always be positive integers*, (b) *the coin denominations are always entered in strictly decreasing order*, and (c) *the money inserted is always no smaller than the buying item price.* You can use `int` variables to store the inputs. Use the strategy described above to determine the number of coins of each denomination to be returned and print out the result.

However, if no coin combinations can be found, print out just the change amount and then the message *“Cannot find combination!”*

Sample Run

In the following sample runs, the blue text is user input and the other text is the program printout. You can try the provided sample program for other input. *Your program output should be exactly the same as the sample program* (same text, symbols, letter case, spacings, etc.). Note that *there is a space after the ‘:’ in the program printout*.

```
Enter six decreasing denominations: 100 50 10 5 2 1↵
Insert money ($): 400↵
Item price ($): 57↵
Change is $343
$100 x 3
$50 x 0
$10 x 4
$5 x 0
$2 x 1
$1 x 1
```

```
Enter six decreasing denominations: 48 26 12 6 2 1↵
Insert money ($): 100↵
Item price ($): 35↵
Change is $65
$48 x 1
$26 x 0
$12 x 1
$6 x 0
$2 x 2
$1 x 1
```

```
Enter six decreasing denominations: 48 26 12 6 2 1↵
Insert money ($): 160↵
Item price ($): 160↵
Change is $0
$48 x 0
$26 x 0
$12 x 0
$6 x 0
$2 x 0
$1 x 0
```

```
Enter six decreasing denominations: 7 6 5 4 3 2↵
Insert money ($): 10↵
Item price ($): 2↵
Change is $8
Cannot find combination!
```

Submission and Marking

- Your program file name should be vending.cpp. Submit the file in Blackboard (https://blackboard.cuhk.edu.hk/).
- Insert *your name*, *student ID*, and *e-mail* as comments at the beginning of your source file.
- You can submit your assignment multiple times. Only the latest submission counts.
- Your program should be *free of compilation errors and warnings*.
- Your program should *include suitable comments as documentation*.
- ***Do NOT plagiarize.*** Sending your work to others is subjected to the same penalty as the copier.