Programming Design, Spring 2015
Bonus problem

Instructor: Ling-Chieh Kung
Department of Information Management
National Taiwan University

Note. The bonus problem is designed by someone else. The credit does not belong to the instructor.

1 The story

It's a cold, hearty Canadian morning and you need a coffee. So you saunter over to the coolest hipster coffee-shop in town, OR Cafe, and plop yourself down on one of the massively comfy couches. You barely get yourself comfortable when you feel a soft tap on your shoulder. Turning, you see Curtiss and Shefali, the friendly store managers. You smile at them, but you quickly realize that all is not well.

"It's the penny!" Curtiss exclaims grimly, "They're taking it away!" You think back to this week's news:

> The penny's days of living large are nearly over – with many businesses unprepared and the federal government having only limited success informing Canadians of the looming change to, well, change.
>
> The Royal Canadian Mint will stop distributing pennies to financial institutions on Feb. 4, 2013 – the same day businesses are encouraged to begin rounding cash transactions to the nearest five-cent increment in a "fair and transparent manner."
>
> For example, a transaction totaling \$1.02 would round down to \$1, but a purchase of \$1.03 would round up to \$1.05, while \$1.07 would round down to \$1.05 and a purchase of \$1.08 would round up to \$1.10.

He continues, "Call me Canadian, but I can't stand the idea of rounding prices up for my customers! It's terrible! I've been up all night trying to find new prices that round nicely, but I can't figure out how! It's really quite difficult..." Thinking about your coffee, you realize that the \$1.24 you paid for it plus the 13% Ontario sales tax comes out to exactly \$1.40. However, had you purchased a doughnut with your coffee, the final price would be \$2.47. And apparently that's somehow unacceptable.

Curtiss rambles on, "Your mission, should you choose to accept it, is to determine the menu that minimizes the number of times that we have to round a customer's purchase, up or down. Of course, we don't want to change prices by too much, so you may only move prices up or down by up to 2 cents." Shefali, surprisingly silent up to now, slides over a USB key and says, "We've been keeping track of customer purchases; here is what people have been buying. I hear you're good at this operations research stuff, so you know how to use this, right?" You sigh, roll your eyes, pull out a pen and paper and start jotting down some ideas. As crazy as these people are, they definitely come up with some interesting questions...

Due to the retirement of the Canadian penny, small businesses will now round purchases to the nearest 5 cents. However, Canadian businesses generally advertise prices before tax, making it incredibly difficult to put together a menu that ends in 0 or 5 without rounding so that pennies are not needed.

Today's challenge is to put together a menu for the fictional OR Cafe (in Ontario, Canada) that rounds cleanly to a 0 or a 5 for the most transactions. Ontario sales tax is 13%. The current menu and a record of historical transactions are here. The only restriction is that current prices cannot be changed by more than 2 cents (up or down).

2 Data

The plain text file "PD-Sp15_bonus.txt" contains three sections:

- The item section: The first line contains the twelve item names, separated by commas.
- The price section: The second line contains twelve fractional numbers, separated by commas, as the original prices of the twelve items.
- The purchase section: The third to the 202th lines record 200 purchases randomly sampled from a typical day. In each line, there are twelve nonnegative integers, separated by commas, indicating the purchase quantities of the twelve items.

3 Tasks

1. Please write a C++ program that can find an optimal way to adjust the prices of the twelve items. Your objective is to maximize the number of purchases (out of the 200 given to you) whose after-tax price will be rounded to a multiple of \$0.05 after your adjustments. Your program should first read (`cin`) the given data set and then print out (`cout`) the adjustments. You should print out twelve price adjustments in cents, where -2 means that the price of that item should be decreased by 2 cents, etc. The twelve numbers should be printed in twelve lines, one in each line.
2. Write a formal report to explain your algorithm and the final suggestion. You may get a higher grade if your algorithm is correct and more efficient than a complete enumeration.

4 Submission rules and grading

1. This is an individual work. ***DO NOT*** discuss with anyone!
2. Limit your report to at most four pages (i.e., two double-sided sheets).
3. Send your report in the PDF format and your program in CPP format to one of the TAs at their e-mail addresses. Indicate the file names in the report. Do not send to the instructor.
4. The deadline is ***23:59 pm, May 17, 2015***. Works submitted late for less than one hour will get 1 point off as a penalty. Submissions late for more than one hour will not be accepted.
5. Based on the quality of your program and report, you will get your grades as an integer between 0 and 5. That number will be added into your final semester grades.
6. If you also attend the PDAO contest, you will only get bonus points from one of the two activities.