Juan Ruelas SID 861116223

CS141 Assignment 3

due Friday, June 3

Solution 1: Consulting Firm

A

If our moving cost $M = 10$ and the number of operational months $n = 4$, then we have the table below to analyze.

	Month1	Month2	Month3	Month4
NY	1	3	20	30
SF	50	20	2	4

We are given the optimal plan already, however

B

C

D

Solution 2: Pretty Print

The entire basis of this problem is to be able to take some text that is "not balanced" and turn it into text whose right margin is as even as possible. Look below to see what I mean.

Call me Ishmael. Some years ago, never mind how long precisely, having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world.	$\rightarrow$	Call me Ishmael. Some years ago, never mind how long precisely, having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world.

In order to accomplish this we will need to make use of dynamic programming. Here is the overview of how we will make use of this programming technique:

- Find a recurrence relation for the optimal solution
- Based on our recurrence relation, create an algorithm to solve our problem

A: Recurrence Relation

In order to come up with a recurrence relation we need to understand what it is we are exactly computing. We are trying to re-arrange the text such that the "slack" or amount of spaces from the last word of every line are evenly distributed among the entirety of the text. This becomes a trivial task until we define what "even" means. Let us assume that "even" means to minimize the sum of all the "slacks". If we were to take this approach we would be left with several different viable solutions. See below for more details. Assume we have a max row width of 10.

```
0123456789                    0123456789
Ruelas    --> 4               Ruelas    --> 4
Juan is my--> 0       VS      Juan is   --> 3
name      --> 6               my name   --> 3

4 + 0 + 6 = 10                4 + 3 + 3 = 10
```

As you can see from the figure above, since we define "even" to be the minimization of all the slacks of every line, we will have multiple "optimal" solutions. This is a problem since we can reproduce identical slack summations with different text patterns and as we can visually see, the pattern on the right appears to be more "even" than the left one. Because of this, we will define "even" to mean the summation of all the $slacks^2$. This will enable us to be greedy with our spaces and will force us to minimize the amount of slack on *every* line.

Slack Squared

Since we can see that by calculating the summation of the slacks of each line will give us an "optimal" solution,

B: The Algorithm

In order to solve this problem we will divide this into two sub problems. The 1st subproblem will deal with using dynamic programming to create a table of slack lengths knows as our $slack-cost-table$. The second problem will deal with using the slack cost table to actually calculate the

---

Solution 3:
