Introduction to Computer Science
HW #4
Due: 2015/05/13

Homework Rules:

Hand-written homework can be handed in in class. Otherwise, you may contact the TA in advance and then bring the hardcopy to the TA in BL421.

As for the programming part, you need to upload it to CEIBA before the deadline. The file you upload must be a **.zip** file that contains the following files:

README.txt

HW01_b03901XXX (a folder that contains all .cpp & .h as required),

1. Do not submit executable files (.exe) or objective files (.o, .obj). Files with names in wrong format will not be graded. You must **remove any system calls**, such as system ("pause"), in your code if you use it.
2. In README.txt, you need to describe which compiler you used in this homework and how to compile it (if it is in a "project" form).
3. In your .cpp files, we suggest you write comments as detailed as you can. If your code does not work properly, code with comments earns you more partial credits.

Chapter 5 Review Problems (10% each):

17. The following is an addition problem in binary notation. Each letter represents a unique binary digit. Which letter represents 1 and which represents 0? Design an algorithm for solving problems like this.

```
  YXX
+ XYX
 XYYY
```

46. Design an algorithm that, given a list of five or more numbers, finds the five smallest and five largest numbers in the list without sorting the entire list.

18. Four prospectors with only one lantern must walk through a mine shaft. At most, two prospectors can travel together and any prospector in the shaft must be with the lantern. The prospectors, named Andrews, Blake, Johnson, and Kelly, can walk through the shaft in one minute, two minutes, four minutes, and eight minutes, respectively. When two walk together they travel at the speed of the slower prospector. How can all four prospectors get through the mine shaft in only 15 minutes? After you have solved this problem, explain how you got your foot in the door.

Introduction to Computer Science
HW #4
Due: 2015/05/13

49. Use big-theta notation to classify the traditional grade school algorithms for addition and multiplication. That is, if asked to add two numbers each having n digits, how many individual additions must be performed. If requested to multiply two n-digit numbers, how many individual multiplications are required?

57. Based on the preconditions that `X` and `Y` are assigned nonnegative integers, identify a

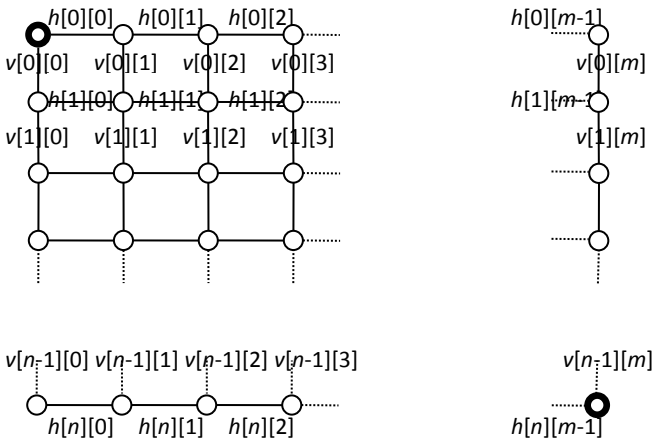
loop invariant for the following `while` structure that, when combined with the termination condition, implies that the value associated with `Z` upon loop termination must be `X – Y`.

```
Z ← X;
J ← 0;
while (J < Y) do
  (Z ← Z – 1;
   J ← J + 1)
```

Note: Try to use pseudo-code to present your algorithms.

Programming Problem (50%): Shortest Path

Consider a graph of *m* by *n* grids:



Use dynamic programming to find the shortest path from the upper-left node to the lower-right node with **right and down moves only**. Costs of vertical edges are stored in an 2D array *v*[0..*n*-1][0..*m*]; costs of horizontal edges are stored in another 2D array *h*[0..*n*][0..*m*-1]. These costs are **positive real-values**.

You can use readParameters() to read all parameters (*m*, *n*, *v*[][], and *h*[][]) from **input**. Remember to call release() when done. Check out **hw4.cpp** for more information.

Your program should print out 2 lines (on screen). The 1st line is the total cost of the shortest right-down path. The 2nd line is a string of (*m*+*n*) characters of 'v' or 'h', standing for vertical (down) or horizontal (right) respectively.

Bonus (5%)

Your program outputs a 3rd line, which is the cost of actual shortest path (no restriction of right or down). To earn full credit, your program should finish within a few seconds for 300x200 grids (input3).