This project is a two part programming assignment. For each part read and follow the instructions carefully. You should turn in a printout of both of your programs and you should submit your source code as usual to the TA via email at `pliu2@umbc.edu`. All work is due on November 7, 2002 at 5:30 pm.

Part 1: Position and Gravity (10 Points)

To help Isaac Netwon with his gravity experiments, you must write a program called **grav.c** for calculating the position of a falling object at given time intervals. The program should take as input from Sir Isaac the initial height from which the object is dropped, and the change in time for printing the position.

The physics of the system are simple. Let h_0 be the initial height of the object, let Δt be the size of the time intervals for calculating the new height, and let g be a constant acceleration due to gravity. Then,

$$\begin{aligned} h_1 &= h(\Delta t) = h_0 - g(\Delta t)^2 \\ h_2 &= h(2\Delta t) = h_0 - g(2\Delta t)^2 \\ h_3 &= h(3\Delta t) = h_0 - g(3\Delta t)^2 \\ &\vdots \\ h_{t+1} &= h((t+1)\Delta t) = h_0 - g((t+1)\Delta t)^2. \end{aligned}$$

The program should prompt the user for a **positive** starting height and a **positive** change in time (i.e. Δt). The constant of acceleration should be

```
#define G 9.8
```

The program should output the starting time (0.0), the starting height, and then all subsequent times (Δt, $2\Delta t$, $3\Delta t$, $\ldots$) and the heights at each time. The program should terminate once the height reaches 0 or passes below zero, printing the ending position of 0.0. You do not need to worry about units. As an example, the following is the output for $h_0 = 10$ and $\Delta t = 0.2$.

```
0.0   10.00
0.2    9.60
0.4    8.43
0.6    6.47
0.8    3.72
1.0    0.20
1.2    0.00
```

Part 2: Arithmetic Table (10 Points)

In this part of the assignment you will create a program, **arith.c** to calculate arithmetic tables for the operations `+`, `-`, `*`, and `/`. Using a menu, have the user select the operator of choice. Once the operator is selected, the user should be prompted for an operand range in $[1, 100]$. The user will need to input both a minimum value (min) for the range and a maximum value (max) for the range. Error checking should be used to ensure that $min < max$ and that $min, max \in [1, 100]$. Once the user input is collected, the program should calculate the arithmetic table for the selected operator on the range of **integer** values and print the results in a nicely formatted table.

As an example, for the operand `+` and an input range of [3,7], the nicely formatted output table would be

```
+   3   4   5   6   7
3   6   7   8   9  10
4   7   8   9  10  11
5   8   9  10  11  12
6   9  10  11  12  13
7  10  11  12  13  14
```

You **MUST** employ a `switch` instruction for using the various operators (to be covered in class on 11/5, read ahead)! Be sure to test your program for all operators and various ranges of values. Note: the `/` operator requires the output to be **non-integer**.