Introduction to Programming in C

Summer Workshop, McGill University

We have provided a set of exercises that cover the lecture on basic programming in C. Once you are done with the exercises, you should be familiar enough with the syntax but also the different operations for control flow of the code, such as **for loops**, **while loops**, and **if-then-else statements**. Furthermore, you should be able to write code that will take some user input and storing it in various types of variables, depending on the usage. Finally, you should also be familiar with basic data structures available in C, such as arrays.

Trial Problem: Newton's Law of Gravity

Newton discovered that the gravity between two objects affect one another. He discovered that the force of gravity exerted on each is proportional to how far away they are from each other, squared. In other words:

$$F \sim 1/r^2$$

We can calculate the amount of force between the two objects if we know the mass of the two objects and how far away they are from each other, using this formula:

$$F = \frac{G * (mass1 * mass2)}{r^2}$$

Write a program that asks the user for the values of mass1, mass2 and r. Assume G = 0.0000000000667. The program should print the result for F. The following is some sample code for basic I/O in C:

```
#include <stdio.h>

main() {
   double mass1 = 0;
   double mass2 = 0;
   double G = 0.0000000000667;
   double r = 0;

   printf("Mass of object 1 = \n");
   scanf("%lf", &mass1);
   printf("Mass of object 2 = \n");
   scanf("%lf", &mass2);
   printf("How far apart from each other are they (in meters)? \n");
   scanf("%lf", &r);

   // Fill in the calculation and output of the gravitational force
}
```

The first line is required to import the standard C IO library. This gives us access to functions such as *printf()* and *scanf()*, both of which we use to interact with the user through the terminal and read input values for the object masses and the distance between them. You can copy this code in a file and compile it (using gcc). You can then run the executable through the terminal and see the code running for yourself. In this exercise, you are asked to complete this program to output the force F, based on the given equation.

Exercise 1

We can calculate the distance an accelerated object can travel with the formula:

$$distance = \frac{acceleration * time^2}{2}$$

Assume that we want to study how an object accelerates when it falls. We know that the acceleration due to gravity is

$$acceleration = gravity = 9.8\frac{m}{s^2}$$

Write a program that uses the above formula and value for gravitational acceleration, asks the user for three different time values (in seconds) and prints to the screen the result of applying each of these time values to the formula using gravity as the acceleration.

Exercise 2

A space shuttle around Earth burns its thrusters just enough to counter-act the gravitational acceleration of the Earth, i.e. the shuttle is floating in space. Suppose that the space shuttle experiences failure in all its thrusters and accelerates towards Earth at 9.8 m/s^2. Write a program that takes as input the orbit distance of the space shuttle from Earth and outputs the time available for the astronauts to fix the problem, i.e. the time *t* before the shuttle crashes into Earth. HINT: You should look into the C math library to find a function that calculates square root for you.

Exercise 3

Assume the sensors of a spacecraft function by depositing integer numbers into memory. These integer numbers are the measurements made by the instrument. Assume an array is used to record the information for analysis. Assume a spacecraft wants to determine where it is in space. It does this by locating a familiar object. In our solar system, that would be the sun, since it is the brightest object. Assume our sensor sweeps around recording light intensity. The cell of the array containing the largest number would be where the sun is. Write a program that uses an array of size 10. It asks the user to input ten integer values for the array. The program then scans the array to locate the largest value. The program then prints to the screen the largest value and the cell number where it was found.

Exercise 4

Using the information from Exercise #1, make a more advanced program that uses a function called CalcDistance that implements the distance formula from the Exercise #1 question. The function will look something like:

double CalcDistance(double acceleration, double time);

Write the program such that the main program reads the three values for time, but then the program calls the CalcDistance function to get the result in distance. This is then displayed for each time.

Exercise 5

This follows Exercise #3.
The sensor's array normally has a size equal to some logical purpose for the sensor. In our case, assume that the sensor scans a 360 degree circle around itself looking for the brightest object. However, asking the user to input 360 values is asking too much. Modify Exercise #3 to use a function that calls a random number generator to initialize all the array's 360 cells. Then the program will locate the largest value and in which cell it was found. This information is then printed. It is possible that more than one cell had this largest value. Your program should then count the number of times this largest value appears in the array. Print this number.