GENE 121 Assignment 8 – Functions, Functions, Functions! Winter 2013

This assignment will be completed in pairs.

At the start of the lab session you should sit at the box of Lego parts corresponding to your team number. All parts should be returned at the end of the lab session in the same manner as you received them.

Question 1 (RobotC) (Partner A should code the functions, and partner B should code the main program).

Partner A

Write a function that:

- Receives a length of time in milliseconds and a motor power
- Runs the robot forward at the above power for the given amount of time
- Calculates the speed the robot was travelling, in cm/s, over that time interval using the **encoders** to measure distance. The radius of the robot's wheel is 1.5 cm
- Calculates the speed the robot was travelling, in cm/s, over that time interval using the **ultrasonic sensor**
- **Returns both speeds**

Partner B

Write a program that:

- Displays the day and your group number on the display, e.g. Tues 07
- For 4 button pushes:
  - Waits for the user to press and release the orange button
  - Calls the above function for 2 s, saves the returned speeds and displays them to the screen
  - Calculates the percent error in the ultrasonic sensor measurement. Assume that the encoder is more accurate. (%error = 100 * (ultrasonic-encoder)/encoder)
- The robot should be moving at 20% power for the first iteration, 40% power for the second, 60% power for the third, and 80% power for the fourth iteration of the above loop.
- Calculate and display the average percent error of the ultrasonic sensor speed measurement

Demonstrate your program to a TA. Print your code, and copy (by hand) the screen output onto your printout. Submit the code and output.

Question 2

A ship in trouble in the “Bermuda triangle” area of the Atlantic Ocean radios in its name and position. A file called **`ships.txt`** contains the names, positions (as latitude and longitude coordinates in degrees and minutes), and maximum speeds (in knots) of any ships that may be able to help. There are fewer than 50 ships listed in the file. For example,

```
America    30 17     66 57     2
```

indicates that the America is at latitude 30 degrees 17 minutes North and longitude 66 degrees 57 minutes West, and that its maximum speed is 2 knots (nautical miles per hour).

Useful Information:

- You can assume ships able to help are in the Atlantic Ocean and the northern hemisphere (i.e. 0° <= latitude <= 90° and 0°<= longitude <= 105°)
- 1° latitude ≈ 111 km
- 1° longitude ≈ 111 km x cos(latitude angle) km
- 60 minutes = 1 degree
- 1 knot = 1.8536 km/h
- Equator is 0° latitude and the North pole is 90°

Partner B

Write, test and debug the following functions:

(a) Write a function with prototype

```
void position(int latDeg, int latMin, int longDeg, int longMin, double & x, double & y);
```

that calculates the position of a ship in km

(b) Write a function with prototype

```
int getShipData(ifstream & finShip, string ship[], double x[], double y[],
                double speedK[]);
```

that reads ship data from the already opened file stream, stores it in the arrays, and then returns the number of ships in the file.

(c) Write a function that receives the positions of two ships (in km) and returns the distance between them.

(d) Write a function with prototype

```
double timeShip(double x1, double y1, double x2, double y2, double speedK);
```

that:

- Receives the positions of two ships (in km), the ship in trouble, and the ship that may be able to help
- Receives the speed of a ship (in knots) that may be able to help
- Returns the time (in hr) for the ship that may be able to help travel to the ship in trouble (Think: What if both ships have a speed of zero?)

Partner A

(e) Write a function that receives:

- The number of ships to check
- Ship information (position in km and speed) arrays
- Position (in km) of the ship in trouble

and returns the index in the array (i.e. the number of the location in the array) and the time to reach the ship in trouble of the ship that can arrive the fastest

Some times and distances to the **`Waterloo`** (located at 32 deg 17 min N and 64 deg 49 min W).
For debugging only.

```
               Name    x km    y km      km/h    Dist    Time
        Andrea_Doria 6089.15 3638.95       0 55.8998   NEVER
             Bismark 6163.69 3772.15 15.1995 205.434 13.5158
         Black_Pearl 6037.63 4467.75  6.4876 885.437 136.481
```

Write, test and debug a program that:

- Opens the files **`ships.txt`** and **`shipTrouble.txt`**, and checks whether they have opened
- Gets the ship data for any ships that may be able to help
- For each ship in trouble, read from the file **`shipTrouble.txt`**, the program should output the name and time for the ship that can reach the ship in trouble in the least amount of time

The program should call the above functions.

Submit your program with output for all of the ships in trouble.

Bonus (C++) **(This Bonus question is to be submitted individually).**

The file **`ids_in.txt`** contains an unknown number of ID numbers. Write a program that:

- Opens the file **`ids_in.txt`** and checks whether it opened properly
- Reads the ID numbers and determines the number of times that each unique ID number occurred in the file
- Outputs to the file **`ids_out.txt`** each ID number and its frequency, the ID number(s) that occurred the least number of times, and the ID number(s) that occurred the most number of times

You can assume that there are fewer than 100 unique ID numbers in the file.

A sample of the output file is:

```
ID   # occurrences

2032    479

3865    501

6074    503

  .       .

  .       .

  .       .

All IDs with 235 occurrences 2095 1151

All IDs with 524 occurrences 8329
```

Write at least two non-trivial functions (at least one of which has an array as a parameter) and a main program (which primarily just calls the functions) to solve this problem. Submit your code with output for the file given on UW-Learn.