CS23710 Assessed Assignment 2013-2014
“For those in Peril on the Sea”.

David Price and Fred Long

November 11, 2013

1 Introduction

PLEASE NOTE: We require that your programs are compliant with a recognised version of ANSI ‘C’.

Our “official environment” in which all your user programs must run, and in which we will test them, is that provided by the Computer Science Linux Mint Olivia computers as installed in C56.

You **must** use NetBeans and the compilers and debuggers accessible from it to develop and debug your program.

2 Monitoring Carnsore Point to Braich-y-Pwll

You have been contracted to produce a software system to co-ordinate the deployment of rescue assets (e.g. helicopters and lifeboats) to assist mariners who are in peril at sea.

The initial “test” deployment will be used to co-ordinate rescues in the waters between Wales and Ireland known (mostly) as St Georges’s Channel. This area is depicted on the Admiralty Chart number 1410 titled “Carnsore Point to Braich-y-Pwll”.

All the ships travelling in this area will be using a simplified version of the new internationally deployed system known as Automatic Identification System (AIS). Your system is required to be fairly basic and will only have a textual interface.

Several “real” systems do exist using this information, some carried on the control bridge of ships and others, collecting AIS transmissions, but presenting them as web sites.

One such site is

`http://www.shipais.com/currentmap.php?map=SouthIrishSea`

We stress however, that you are **ABSOLUTELY NOT** being asked to produce a complex system like that, your system has to process our “simplified” AIS-like data and just provide text-based solutions to the tasks we present below.

3 Simplifications and Warning

NOTE: While much of the information supplied in this assignment specification is closely related to real issues of monitoring the movement of ships at sea, and co-ordinating rescues, various simplifications have been made to reduce the complexity of the problem.

1. The Lack of Land!

   For instance, while our ship co-ordinates will be (mostly) within the Celtic Sea, St Georges's Channel and the Irish Sea, we have conveniently decided to ignore the presence of any land. Thus, you are not expected to identify any ships about to run aground onto Borth Beach for instance.

   If anyone is interested in an "amusing tale" in this regard, (which is actually claimed to be an urban myth...) try reading...

   ```
   http://joi.ito.com/weblog/2006/01/16/lighthouse-vs-u.html
   or
   http://www.snopes.com/military/lighthouse.asp
   ```

2. The Lack of Tides and Winds

   For the purposes of this assignment we have decided to ignore the effect of tidal flows and of wind direction and speed.

Our information, and any software you produce, should **NOT** be used for co-ordinating the rescue of any real mariners who may be "in Peril on the Sea"!

4 The Programs you must Create

You are required to design, write and test two ANSI C programs to help solve various tasks that could assist a maritime planner to decide how best to deploy rescue assets (i.e. lifeboats and helicopters) to rescue mariners from ships in danger.

Both programs are required to deploy a helicopter and a lifeboat to assist the mariners in distress in response to the reception of a *Mayday call* [1].

The first program, specified in more detail in section 9.1, requires that only a single Mayday call has to be handled and that the ship remains stationary after issuing the Mayday.

The second program, specified in more detail in section 9.2, requires that multiple Mayday calls must be handled. It still assumes that any ship that sends a Mayday remains stationary after issuing the Mayday.

NOTE: if you calculate that the current location of a ship in peril, which has thus issued a Mayday call, is *outside of our shipping area* (as described in section 6.1) then you must display and log appropriate messages but you do *not* send helicopters and lifeboats, that's the job of another rescue management centre.

[1] Online Etymology Dictionary http://www.etymonline.com/index.php?search=Mayday

5 Rescue Assets - Helicopters and Lifeboats

There are various factors that determine which assets are the best to deploy to help a ship or mariner that is in danger.

When allocating lifeboats and helicopters you should allocate those that can get to the ship in danger the quickest, subject to them being able to return safely to their base afterwards.

- Helicopters

  Helicopters can travel quite quickly, but have restricted fuel supplies and can thus only be airborne for limited periods.

  Helicopters will have a "maximum active deployment time" as mentioned below in section 7.2. You **must** assume that the time spent hovering helping a ship in peril **is** counted within that time. A hovering helicopter uses at least as much fuel as one that is en route.

  The UK's helicopter search and rescue service is currently supplied by the combined efforts of the Royal Air Force, the Royal Navy, the Maritime and Coastguard Agency and Bristow Helicopters Ltd. The UK government has announced that a new contract has been issued for Bristow Helicopters Ltd to take over all SAR operations progressively from 2015. A summary of the current and future positions is available at the following URLs.

  `http://en.wikipedia.org/wiki/RAF_Search_and_Rescue_Force`
  `and`
  `https://www.gov.uk/government/speeches/search-and-rescue-helicopters--5`

  Ireland's Coast Guard service also help cover this region and their "search and rescue" service and details of their facilities can be found at the following URLs.

  `http://www.transport.ie/marine/IRCG/SearchRescue/index.asp?lang=ENG&loc=2101`
  `and`
  `http://www.transport.ie/viewitem.asp?id=8622&lang=ENG&loc=2103`

- Lifeboats

  Lifeboats travel relatively slowly but carry quite large supplies of fuel. Inshore lifeboats will be able to remain at sea for reasonably long periods but may have limitations such as only being allowed to operate relatively close to their base. Having said that, inshore lifeboats can be quite fast. Offshore lifeboats will be able to safely remain at sea for extended time periods and will be permitted to operate a long way from their base.

  Lifeboats will have a "maximum active deployment time" as mentioned below in section 7.2. However, you can assume that the time spent stationary helping a ship in peril is **NOT** counted in that time. While floating stationary a lifeboat uses almost no fuel.

  The RNLI (Royal National Lifeboat Institution [2]) operate lifeboats in both the UK and Ireland. Details of the RNLI's lifeboat fleet and lifeboat stations can be located at the following URLs.

  `http://rnli.org/aboutus/lifeboatsandstations/lifeboats/Pages/The-fleet.aspx`
  `and`
  `http://rnli.org/aboutus/lifeboatsandstations/stations/Pages/Stations-a-z.aspx`

[2] http://rnli.org/Pages/Default.aspx

6 Locations and Distances

6.1 Units used in our Data and for Measurements and Calculations

In accordance with one common maritime choice, the "units" used in our data are degrees of latitude and longitude for locations and degrees from true north (clockwise) for directions, nautical miles for distances and knots (nautical miles per hour) for speed.

Ship, Helicopter and other Positions Positions will be provided as a latitude and longitude in degrees and expressed as real numbers.

Latitude: will be given as positive values in the Northern hemisphere and negative values if a location is south of the equator. Thus, latitude goes from 0.000 at the equator to 90.000 at the North Pole and to -90.000 at the South Pole.

Longitude: will be given as positive values if the location is to the East of the Greenwich meridian and negative values if the location is to the West of the Greenwich meridian. Thus, longitude will go from 0.000 at Greenwich to negative values as you travel due West and -180.000 is the far side of the world. As you head due East from Greenwich, values increase positively and 180.000 is also the far side of the world, and indeed exactly the same place as -180.000.

Example Locations: Here are two example locations.

1. A latitude of 52.500 with a longitude of -5.500 is in the middle of St. Georges's Channel (i.e., between Wales and Ireland)
2. A latitude of 51.767 with a longitude of 1.283 is the approximate location of the Gunfleet light about 10 miles off Frinton-on-sea on the UK East coast just North of the entrance to the Thames estuary in the southern North Sea.

Speed Speeds will be given in knots as a real number. This will be given as "speed over the ground".

Direction of travel The direction of travel will be given in "degrees true" and will be the "course over the ground". The direction tells you where a ship or helicopter is going. Thus, a direction of 45.000 means a ship or helicopter is heading towards the North East, a direction of 180.000 means due South and 270.000 means due West.

Our Shipping Area We are only "interested" in what ships are doing if they are within "our" shipping area. In other words if they are on, or will enter, the sea area covered by the chart "Carnsore Point to Braich-y-Pwll".

- If ships go further South than a latitude of 51.667, or go further North than 52.833, you can assume they have sailed away and we no longer need to answer their Mayday calls.
- If ships go further East than a longitude of -3.833, or go further West than -6.667, you can also assume they have sailed away and we no longer need to answer their Mayday calls.
- But a twist; ships which are South of 51.667 but are sailing North might enter our area and so on. Thus, you may need to consider which direction ships are going, as well as where they are, to be sure you can decide to ignore them.

6.2 Estimating a Ship's Position

Clearly, you will need to calculate the current location of ships based on their last reported location and course and the amount of time that has passed since that report was received.

You are required to write a function that calculates the estimated location of a single ship. That function should take a location, a course and speed and a time duration (since last report) as parameters and it should produce a new estimated location as its return value.

Here's the "mathematics" you need to be able to calculate a new position.

In what follows:

CG represents the course over the ground of a ship from True North in degrees

CGR represents the course over the ground of a ship from True North converted into radians

K represents the speed of a ship in knots

D represents a duration of time in minutes

LTS represents a ship's last reported latitude in degrees

LTSR represents a ship's last reported latitude converted into radians

LGS represents a ship's last reported longitude in degrees

LTF represents a ship's estimated latitude in degrees after the time duration

LGF represents a ship's estimated longitude in degrees after the time duration

To convert an angle in degrees to an angle in radians you have to multiply by PI and divide by 180.0 The value for PI can be accessed as a constant called M_PI which is defined in the math.h standard header file.

That is,

(angle in radians) = (angle in degrees) * M_PI / 180.0;

The references to sin and cos in the equations given below are references to the mathematical functions sin and cos.

Pre-compiled versions of these exist in the C standard maths library and their signatures are again defined in the math.h standard header file. As well as including the math.h header file in your programs, you will also need to tell the compiler to link in the math library. This is normally done by having an extra -lm parameter on the compiler command line.

LTF = LTS + (K * cos(CGR) * D) / 3600.0;

LGF = LGS + (K * sin(CGR) * D / cos(LTSR)) / 3600.0;

The accuracy of above mathematics depends on a few simplifying assumptions such as the earth being round and the ship movement not being "too fast", nor too near the North or South poles.

6.3 Calculating Distances Between Locations

We have provided a compiled version of one function that calculates the distance between any two locations, that is detailed in a section 8 of this worksheet.

7 The Data Files Provided

We provide a set of data files to describe the last known location of the ships and the directions in which they were moving, rescue assets that are available and Mayday calls that occur.

7.1 Ship Locations

The Ship Location files contain a set of lines of information representing the state of our shipping area at a certain point in time.

Each data file contains a fixed format introduction followed by a set of entries, one per ship. Your program **MUST NOT** assume any fixed specific number of ships but should process the file until end-of-file is encountered.

The first line of the file contains the date and the time using a 24 hour clock. The line contains six integers, separated by spaces, which are day month year hour minute second.

The remainder of each file contains lines of information, each line representing a single ship and its location and speed. You may assume that the ship AIS ID values are all unique.

The values supplied on each line are separated by spaces and the values and their types are in the following order:

an "AIS ID" for the ship as an alphanumeric string without any embedded gaps;

the latitude of the ship in degrees as described above;

the longitude of the ship in degrees;

the course over the ground of the ship in degrees;

the speed over the ground of the ship in knots.

Here is the contents of a (small) example file for the 13th November 2011 at 13:04:00.

13 11 2011 13 04 00

GW1927 52.408 -4.117 1.000 0.000

GS452 51.750 -4.300 5.000 10.000

EI597 52.100 -6.000 90.000 12.000

EI600 52.000 -5.900 10.000 15.000

You will notice that one boat, whose AIS ID is given as GW1927, is currently moored in Aberystwyth harbour and thus is not actually moving!

7.2 Rescue Asset Locations

The Rescue Asset Location files contain a set of lines of information representing the rescue assets relevant to our shipping area.

Each data file contains a set of entries, one per rescue asset. Your program **MUST NOT** assume any fixed specific number of rescue assets but should process the file until end-of-file is encountered.

The file contains lines of information, each line representing a single rescue asset, its callsign, type, location, operational speed, maximum deployment duration and the length of time that it takes to service, refuel and re-crew the asset when it returns to base after it has assisted with a rescue.

You may assume that the asset callsigns are all unique.

The values supplied on each line are separated by spaces and the values and their types are in the following order:

a "callsign" for the rescue asset as an alphanumeric string without any embedded gaps;

the asset's type as a single letter (H for helicopter, L for lifeboat);

the placename of the rescue asset's base as an alphanumeric string without any embedded gaps;

the latitude of the asset's base in degrees as described above;

the longitude of the asset's base in degrees;

the operational speed over the ground of the asset in knots;

the maximum active deployment time of the asset in minutes;

the time it takes to service/refuel/re-crew the asset after it returns to base in minutes;

Here is the contents of a (small) example rescue asset file.

```
Fishguard_ILB L Fishguard 52.006 -4.981 25.0 180 30
Cardigan_Lifeboat L Cardigan 52.092 -4.684 35.0 150 100
Cardigan_ILB L Cardigan 52.092 -4.684 25.0 180 30
New_Quay_Lifeboat L New_Quay 52.208 -4.355 16.0 525 120
New_Quay_ILB L New_Quay 52.208 -4.355 25.0 180 30
Rosslare_Lifeboat L Rosslare 52.255 -6.336 25.0 600 120
Rescue_116 H Dublin_airport 53.421 -6.270 120.0 225 60
Rescue_122 H Valley_airport 53.248 -4.535 112.0 300 60
```

You will notice that some RNLI locations have more than one lifeboat.

7.3 Mayday Calls

The Mayday Call files contain a set of lines of information representing Mayday calls from ships in our area.

Your program **MUST NOT** assume any fixed specific number of Mayday calls but should process the file until end-of-file is encountered.

Each line of the file contains the date and the time using a 24 hour clock given as six integers, separated by spaces, which are day month year hour minute second followed by the "AIS ID" (as described in section 7.1) of the ship that issued the Mayday call. The line then has two further integers being the number of minutes it will need assistance from the lifeboat and the helicopter respectively after they arrive to help.

Here is the contents of a small Mayday file for the 13th November 2011.

13 11 2011 13 04 00 GS452 45 20

13 11 2011 15 14 23 EI597 60 30

13 11 2011 15 34 35 EI600 20 15

7.4 Alternative Data Files

The data we provide must not be "hard coded" into your program in any way.

It should be possible for us to provide new data files and then use your program to process the data purely by us providing the names of the new files.

Indeed, we may choose to deliberately have some extra data files, of exactly the same format, which we choose to use when we evaluate your program, which quite deliberately are **NOT** available to you during development!

While any files we would use would be of the same format to those described above, we would have different numbers of ships etc.

8 Pre-compiled Program Code that we Supply

We are providing you with a pre-compiled version of a function that you are **required** to use.

1. The function calculates the shortest distance, in nautical miles, between two places on the earth's surface. It thus calculates the so-called Great Circle distance. The code is based on the calculation as described on the web page

   `http://en.wikipedia.org/wiki/Great-circle_distance`

   The detailed definition of the interface to our function is provided as a well commented C header file which is downloadable from the Blackboard course web site.

   However, in simple terms, it expects you to supply two locations and it then calculates the distance between them.

   Each "location" must be supplied as a struct with two fields. The fields are real numbers (as type double) and represent the latitude and longitude of a location. The function's return value is a double and is the distance between the two locations in nautical miles.

9 Your Missions

9.1 Your Main Mission

For this section we assume that ships stop at the time they issue a Mayday call (imagine their engine fails).

You are required to have the following features in your program.

1. The program must prompt for the name of a file which contains details of all the rescue assets that are available to assist in our sea area.

   Your program must open the file, read in all the data and store it in a suitable set of data structures and variables etc.

2. The program must prompt for the name of a file which contains the shipping data for our sea area.

   Your program must open the file, read in all the data and store it in a suitable set of data structures and variables etc.

3. The program must prompt for the name of a file which contains details of all the Mayday calls.

   Your program must open the file, read in (only) the first Mayday call.

4. Your program should now select and allocate a helicopter and a lifeboat to assist the mariners and their ship and display the times at which they arrive. You should make selections on the basis of the rescue assets that can arrive first but subject to them being able to return safely to their base after helping for the time periods indicated in the Mayday call.

5. As well as the output mentioned above, a log file should be produced to record any significant events that occur. The log entries should show times, the ships and rescue assets involved and the nature of the event.

9.2 Your Extended Mission

In this section you must deal with a slightly more complex scenario where there are multiple Mayday calls which you have to service.

You now need to allow for the time spent on a call and the time for rescue assets to return to base and refuel before becoming available to help with another Mayday.

You are now required to produce a **second** program, (obviously developed from your first program) which has the same features as the first program but is now capable of dealing with all of the Mayday calls included in each Mayday file.

You now have to cope with allocating the best helicopter and lifeboat given that some may still be busy dealing with a previous call, returning to base or refuelling.

9.3 Implementation Requirements

You are expected to make good use of the facilities of the C programming language (1989), C99 or C11 that are supported by our compilers.

Your program **must** make use of **arrays** and/or linked data structures at appropriate places.

Your program **must** make good use of **functions** to modularise your code in a sensible way.

9.4 Comments and layout

We do of course expect you to sensibly comment your programs, making sure that all comments add real value and do not just, in essence, duplicate code. The program should have good layout and must use meaningful names for variables, structures and other identifiers.

9.5 Choices you are free to make

You need to read in the data files and store the information within suitable data structures, which you have designed, within your program.

Your program should use multiple functions, located in multiple files as you choose.

As part of our assessment, we will evaluate the quality of the choices you have made.

9.6 Testing and Data Processing Requirements

As well as any testing you conduct to debug and validate your program, you **MUST** also analyse each of our data files.

We will produce a second **"further information" document** clearly stating exactly how many datafiles we have produced, their names and how you access them.

10 Assessment Criteria

Bearing in mind that this project concerns software development, the most appropriate "assessment criteria" are those in Appendix AA of the student handbook, namely those for "Assessment Criteria for Development".

This assignment is worth 50% of the total marks available for this module.

NOTE: Solutions which do not tackle the extended mission could potentially be awarded a mark of up to 75% if they cannot be criticized in any manner and are accompanied by all other materials as indicated below.

Solutions must also tackle the extended mission for marks of 76% or above.

11 The Material You Must Submit

The submission must be made electronically using Blackboard. There will be appropriate links in the Assignments part of the course materials for module CS23710.

You must submit (via Blackboard) four things.

1. NetBeans Project Export - "Main Mission".

   Your NetBeans Project Export **MUST** be made using NetBeans as installed on our Linux Mint computers in room C56.

   You MUST make sure that your export includes all of the source code of your program addressing the "main mission". **Note:** we will be judging the quality of the layout of the program you have produced.

2. A Design, Compile / Run Document - “Main Mission”

   You MUST submit a document (in PDF format). This document must include:

   - A short section of between two and four sides of A4 in length describing the design decisions you made. Include diagrams if you wish to help explain your design.
   - If you have used any minor pieces of code taken from elsewhere, or drawn understanding from published books or web pages, you should include appropriate references and a bibliography in your document. Please make sure you fully understand the department’s stance on plagiarism as described in the current student handbook.
   - The document must also include a section showing the output produced when attempting to compile your program for the “main mission” and any errors or warnings that are produced. This can be a collection of “screen shots’
   - The document must also include a section showing the output produced when running your program for the “main mission” and it processing our supplied data files. This can be a collection of “screen shots”.
   - The document must also include a subset of the contents of your log files.

3. NetBeans Project Export - “Extended Mission”.

   A second NetBeans Project Export but this time related to your solution of the “extended mission”.

4. A Design, Compile / Run Document - “Extended Mission”

   You MUST submit a second document (in PDF format) in exactly the same format as above, but this time containing material related to your solution of the “extended mission”.

NOTE: This is an “individual” assignment and must be completed as a one person effort by the student submitting the work.

By making your submission on Blackboard it will be deemed that you are declaring the submission to be your own work.

12 Submission Date

Your solution to this assignment must be made on Aberlearn Blackboard before 4:00pm on Thursday 12th December 2013.

It is not acceptable to miss any lecture for the purposes of completing assignment work, in terms of the department’s monitoring of student attendance and any actions that might be taken in respect of poor attendance.

13 Feedback

To enable you to generate “self feedback” we plan to release our own “outline/sample solution” (which we note will not be perfect!) during December, hopefully before Xmas day. You will then be able to use this to see how we tackled the problem, especially in any areas where you had difficulties.

It is our intention to provide you with individual feedback on your work by approximately mid January.