CS1110 Introduction to Systematic Programming
Week 2 -- Practical Class 2

Programming Exercises
Problem 1
Convert the algorithm that you developed for question 1 (including part (b)) on Problem Sheet 1 into a complete **interactive** Ada program. Don't forget to include suitable `Put` statements in your program to output appropriate user instructions and prompts. If you did not complete the problem yourself, convert the algorithm given overleaf.

Type in this program in Emacs, and save it in a file called `telephone.adb` (say)

Then compile the program. If it fails to compile, read the compiler error messages and then use Emacs to correct any errors in the program. Then compile it again (don't forget to save the file first!).

When it compiles successfully, run it and test it for various short lists of telephone call costs such as:

```
27 103 12 6 52 0
```

If the program gives incorrect results, try to figure out why. Then use Emacs to correct the program and re-compile it and run it again. Make sure the program works correctly for an empty list of telephone charges.

Problem 2
You may not have time to complete this exercise in class. If not, try to complete the exercise in your own time.

The greatest common divisor (GCD) of two integers is the largest positive integer which divides them both exactly. For example, the GCD of 45 and 75 is 15.

The GCD has many applications in mathematics, for example in order to reduce a fraction such as 45/75 to its 'lowest terms' one needs to divide numerator and denominator through by their GCD, producing the result 3/5.

The GCD could be determined by factorising the integers completely into prime numbers and then 'cancelling out' common prime factors but this is very inefficient. There is a better algorithm dating back to the Greek mathematician Euclid. This is given below:

```
-- Algorithm to calculate the Greatest Common Divisor (GCD) of
-- two non-negative integers.  This is known as the Euclidean
-- Algorithm and dates back more than 2000 years.

Get(Item => First)
Get(Item => Second)
WHILE Second ≠ 0 LOOP
   Set Remainder to the remainder when First is divided by Second
   First := Second
   Second := Remainder
END LOOP
Put(Item => First)      -- this is the required GCD
```

Convert the above algorithm into a complete interactive Ada program, adding various `Put` steps to output user instructions and explanatory text where appropriate. Enter the program into the computer using Emacs, then compile, run and test it for various inputs.

What happens with your program if

the value of `First` is less than `Second`?
either `First` or `Second` or both are zero?
the user unexpectedly inputs one or two negative integer values?

Model Solution for Question 1 on Problem Sheet 1

```
TotalCost := 0
NumberOfCalls := 0
NumberOfDearCalls := 0

Get(Item => Cost)
DearestCallSoFar := Cost

WHILE Cost /= 0 LOOP
   TotalCost := TotalCost + Cost
   NumberOfCalls := NumberOfCalls + 1

   IF Cost > 50 THEN
      NumberOfDearCalls := NumberOfDearCalls + 1
   END IF

   IF Cost > DearestCallSoFar THEN
      DearestCallSoFar := Cost
   END IF

   Get(Item => Cost)
END LOOP

Put(Item => "Number of calls made was ")
Put(Item => NumberOfCalls)

Put(Item => "Total cost of calls made was ")
Put(Item => TotalCost)

Put(Item => "The number costing more than 50p was ")
Put(Item => NumberOfDearCalls)

Put(Item => "Cost of the most expensive call was ")
Put(Item => DearestCallSoFar)
```

Model Solutions On-Line

Model solutions for the problem sheets are made available on-line on the CS1110 module web site on the CS Intranet. Solutions are released as the module progresses usually a day or so after the relevant tutorial period in which the problem is discussed. You will find them under the **Tutorial Material** link.

Documents are usually available in two forms: as the MS Word files (extension `.doc`) and as PostScript files (extension `.ps`). If viewed by a web browser in Windows then normally the Word or Postscript file will be displayed automatically. However,when using Netscape on Unix you cannot view MS Word documents (unless you install Star Office software -- not discussed in this module), but you can view PostScript documents. Currently Netscape will automatically open the document when you click on the link using CDE Image Viewer. However GhostView is a better viewer and we recommend that you use it instead. Like Gnat GhostView is Gnu software from the Free Software Foundation.

To make Netscape open PostScript files with GhostView, select **Preferences** from the Netscape **Edit** menu. Click on the triangle next to Navigator so that the options "Languages", "Applications" etc. become visible, then click on **Applications**. In the Scroll-Box that appears scroll down until **PostScript Document** appears (there is quite a long list to scroll through). Click on **PostScript Document** to select it, then click **Edit**. Then in the Helper Application Window that appears, click against **Application** and replace `/usr/dt/bin/sdtimage %s` by

```
/usr/local/bin/ghostview %s
```

then click **OK** (twice). In all future sessions PostScript files will now be opened inside Netscape by GhostView rather than the CDE image viewer. Use the **Page** menu in GhostView to move through the document from page to page. Use the **MagStep** menu to change the size of the displayed text.

Note you can set up other helper applications to be used to view other types of document in the same way. For example to have Netscape open PDF documents with Adobe Acrobat Reader, set the helper application for PDF Documents to

```
/usr/local/bin/acroread %s
```

Downloading Files via Netscape
You can also download a copy of any of the files on the CS1110 web site to your Unix user area by right-pressing with the mouse on the file link `Solutions01.ps` (say), and then selecting **Save Link As ..** from the pop-up menu that appears and then type ~/`Solutions01.ps` in the Selection text box.

As ~ is short for your home directory, the file will be stored in your Unix home directory under the name `Solutions01.ps`. Note that as you type `~/` in the text-box, it will turn magically into `/eas/d206/your-unix-username/`.

You can then view the file by typing (in an xterm window):

```
ghostview Solutions01.ps &
```

or print to the laser-printer `cs265ps` (situated in Room MB265) by typing instead

```
print cs265ps Solutions01.ps
```

Note this costs 3p per sheet and only works if you have paid money in into your account by going to Room 477. See the CS Support Document on printing for more info..

Alternatively PostScript files can be printed from within GhostView selecting **Print...** from the Ghostview **File** menu. Type the name of the laser-printer you wish to use in the pop-up box that appears. Note you can print a selection of pages by marking them using **Mark** from the **Page** menu and then selecting **Print Marked Pages** from the Ghostview **File** menu.