PRIFYSGOL CYMRU; UNIVERSITY OF WALES

M.Sc. AND DIPLOMA EXAMINATIONS

JANUARY 2003

SWANSEA

Computer Science

CS M31 Programming

Attempt 2 questions out of 3

Time allowed: 2 hours

Students are permitted to use the dictionaries provided by the University

Students are NOT permitted to use calculators

PRIFYSGOL CYMRU; UNIVERSITY OF WALES

M.Sc. & DIPLOMA EXAMINATIONS JANUARY 2003

SWANSEA

*Students may refer to any notes or books, but are **not** allowed to re-use any existing files stored on floppy disks or on the network.*

Computer Science

Module: **CS_M31**

PROGRAMMING

The examination is in two parts.

From ***10.00am-12 noon***, you should ***design*** your solution to the following problem. **Students are not permitted to leave this part of the examination early.**
Your design should be handed in at the end of this time.

It will be returned to you at the start of the second session ***(1.00pm-5.00pm)*** when you will have access to a PC and given a unique login account for the examination. You will also be supplied with a formatted PC disk on which there will be a number of text files with various file names. You should label this disk with your candidate number, your login name, and, optionally, your full name.

You will be expected to implement the solution that you designed in the first session, using the data on the provided disk, and to save your final solution on the same disk for submission. You must save all of your code and your results files on to the disk provided. The version of your program that you wish to be considered as your submission should be saved in to a file called "EXAM.PAS" or "EXAM.DPR". All other .PAS files on the disk will be ignored (unless they are units utilised from within EXAM.PAS). This disk should be handed in at the end of the exam, along with your design document, plus any other comments you would like to make indicating problems, progress, changes in the design etc.

The question comprises multiple parts. You should aim to complete all parts in the time allotted. Your solution will be marked on style, correctness and completeness.

You should save your work regularly during the exam. Work lost during the exam cannot and will not be taken in to account during marking. The exam question is not designed to be demanding, it is intended to give you an opportunity to demonstrate your understanding of what constitutes good style in programming. Do not waste time implementing code that is not required by the question. This includes the coding of elegant user interfaces. As a rule of thumb: Unless it is explicitly required by the question, leave it out.

Time allowed: **2 hours** and **4 hours**

Background:

You have been contracted to produce software that will check the mileage allowance claim forms of a team of travelling salesmen. On a monthly basis, the salesmen submit a claim-form detailing their routes and the number of miles they are claiming for. The format of the submitted information is detailed below. The software must use a look up table (read in from the disk) to check that the mileage being claimed is not more than 120% of the mileage calculated by using the look up table.

The software must produce as output, two lists:

- The first list will be all those members of the sales team who have over-claimed by more than 20%.

- The second output list will be a sorted list of all sales people, containing a tabulation of the following information:

  Name of salesperson
  Total Distance Claimed
  Total Distance Calculated
  Difference between distances

The list should be sorted in descending order of distance travelled (i.e. greatest distance travelled to smallest distance travelled).

The full process is described as follows:

1) Read distance table from file (**Places.txt** and **Distances.txt**)
2) Read first salesperson's details from file (**Travel.txt**)
3) Calculate the "correct" distances using the look up table
4) Compare the "correct" and salesperson-provided distances
5) Produce the required output

The distance table is provided as two text files in the following formats:

Places.txt contains a list of place names (one per line, in ASCII format) as illustrated here:

Aberystwyth
Bangor
Bridgend
Cardiff
.
.
.

Rhyll
Swansea
Tenby
Wrexham

The second supplied file (**Distances.txt**) is also a plain ASCII file, containing 3 space-delimited numbers per line in the following format:

Starting Point (Integer)*[space]*Ending Point(Integer)*[space]*Journey Distance (Real)

e.g.

1 2 32.6
1 3 97.4
1 4 87.3
1 5 104.9

etc…

The meaning of the first line of **Distances.txt** is that the calculated distance between Aberystwyth (1) and Bangor (2) is 32.6 miles. The second line shows that the distance between Aberystwyth (1) and Bridgend (3) is 97.4 miles. This file contains one line for every possible pairing of locations listed in **Places.txt**. However - the distances are assumed to be symmetrical so you will find a distance between 1 and 5 - but not a distance for 5 to 1, since they are assumed to be the same. Also, all distances from 1 to 1, 2 to 2 are assumed to be zero, and are not included in the **Places.txt** file.

The data received from the sales staff on their mileage claim forms is pre-processed for you, into a single file (of ASCII text) called **Travel.txt** illustrated below.

The file contains a repeating group of 3 lines. Each group of three lines represents the following:

Sales person number (a unique ID)
Sales person name (corresponding to the ID number)
Total distance claimed for*[Space]*List of Locations (at least two) separated by *[Space]*s

e.g.

1
James Brown
269.7 1 2 3
1
James Brown
512.6 1 7 10 9 1
2
Graham Hill
312.2 5 7 2

This three-line pattern is repeated throughout the file. The third line of a group of three will always contain at least **one** real number, followed by ***at least two*** integers. However - the number of journeys made in one day may be more than one. Hence on the first sample of three lines, we see that Salesman 1 (James Brown) drove from Aberystwyth (1) to

Bangor (2), and from Bangor (2) to Bridgend (3). Mr Brown claims that the total distance was 269.7 miles. He also claims he made a circular journey of 512.6 miles via places 7, 10, and 9.

Questions – Attempt all questions

1) From the floppy disk provided, read in the distance table and store the information in an appropriate data structure.

[10%]

2) From the floppy disk provided, read in the place names and store them in an appropriate data structure.

[10%]

3) From the floppy disk provided, read the information contained in the file **Travel.txt** and store it in an appropriate data structure.

[10%]

4) Create a data structure in which to store the claimed and calculated distances travelled by each salesperson. Process the information read in from the **Travel.txt** data structure to sum the claimed distances travelled and calculate the expected distances travelled for each salesperson. Store the information in the data structure created in answer to part **3**.

[30%]

5) Pass through the data - looking for any salesperson whose claimed distance is more than 120% of the calculated distance. Display the names of these sales people on screen in a tidy format; include the actual distance, claimed distance and percentage difference between the two.

[20%]

6) Write the entire list of sales people to file (**Output.txt**) in ascending order of claimed distance travelled, in the format specified earlier in this specification document.

[20%]