2003 SCHOLARSHIP EXAMINATION
PRACTICAL SECTION

The University of Waikato
Te Whare Wānanga o Waikato

DEPARTMENT	Computer Science
PAPER TITLE	7th Form School Scholarship
TIME ALLOWED	Six hours with break for lunch at the discretion of the supervisor.
NUMBER OF QUESTIONS IN PAPER	Three
NUMBER OF QUESTIONS TO BE ANSWERED	Candidates are to answer ALL THREE questions. All questions are important. Answer as much of each question as you can. Plan your time to allow a good attempt at each question, but be aware that Question three is the most difficult and will take considerably longer than the others.
GENERAL INSTRUCTIONS	Please hand in listings, notes and answers to written questions, and a floppy disk with your program / computer work. Please make sure that programs are stored as plain text files. You cannot assume that the examiner has available any special software that might be required to read your files. Candidates may use any texts or manuals for reference during the exam.
CALCULATORS PERMITTED	Yes

TURN OVER

1. Borrowing from the Bank (Spreadsheet Use)

In this question you are asked to use a spreadsheet to do calculations and to display the results. We expect that the spreadsheet will be used for all calculations - you will be marked down for performing calculations by hand and directly entering the results. Your work will be graded on three criteria.

- *(a) The accuracy of your results.*
- *(b) The skill you show in making use of the capabilities of the spreadsheet.*
- *(c) The presentation of your results. We have deliberately not provided any instructions concerning layout or formatting*

Robin wants to borrow money to buy a car. BustBank offers loans, with a choice of repayment systems and interest rates. To explain their rules we will use, as an example, a loan of $24,000 to be repaid over 24 months, with an interest rate of 6% per annum.

Each of the repayment schemes follows the same basic pattern. Payments must be made monthly. In our example there will be 24 payments to make. Each payment involves paying back part of the loan, and paying interest on the amount not yet repaid.

System 1: The 'standard loan'. Monthly payments on the loan itself are all equal. Loan and interest payments are made every month. For the example: the first payment would be 1/24th of the $24,000 ($1,000) and 6% interest for one month on $24,000 ($120), adding to $1,120. The second payment still includes $1,000 of the loan, but interest is only charged on $23,000 because $1,000 of the loan has been repaid, adding to $1,115.

System 2: The 'interest free start' loan. In this system the borrower does not have to pay any interest for the first few months. Regular loan repayments must still be made. With a two month 'interest free' period the first two repayments on our example loan would each be just $1,000. The third payment would be $1,000 plus one month's interest on the $22,000 left of the loan at that time; adding to $1,110.

System 3: The 'interest only start' loan. In this system the borrower does not have to make any loan repayments for the first few months. They do still have to pay interest every month. With a four month 'interest only' period the first four payments would each be $120 (one month's interest on $24,000). The fifth and following payments would include a loan repayment of 1/20th of the loan (because the whole loan must now be repaid in just 20 months) as well as interest. Payment five is $1,320.

(a) Build a spreadsheet to allow Robin to compare these systems. It should show a table of payments being made each month, with totals at the bottom. You may assume that all loan options Robin considers will involve exactly 24 monthly payments. However, from time to time, the bank alters its interest rate and the number of months 'interest free' or 'interest only', and Robin would like to be able to quickly try new values in the spreadsheet. Try your spreadsheet with the following values.

Robin wants to borrow $36,000. The banks offerings are:

System 1 (standard) with an interest rate of 12%.

System 2 (interest free start) with four months interest free and 15% interest.

System 3 (interest only start) with four months interest only and 11.5% interest.

Which option is the cheapest?

(b) Produce a graph showing the total amount paid against time, for each system.

CONTINUED

2. Numbers and Statistics (Careful and Accurate Programming)

Your programming work in this question will be assessed on two criteria:

(a) Completeness and accuracy of the program.

(b) Good presentation. That is, it should make good use of programming language facilities, be well organised, neatly laid out, and lightly commented.

Your task is to write a program to read in a series of numbers and compute some statistics. The way in which you should handle input, and the statistics you should calculate, are as illustrated in the sample run following (data input by the user is underlined)

```
Please enter the number of values to be processed:   10
Enter value  1:  2.3
Enter value  2:  6
Enter value  3:  5
Enter value  4:  8
Enter value  5:  13.4
Enter value  6:  19
Enter value  7:  3
Enter value  8:  1.7
Enter value  9:  1.4
Enter value 10:  16
You entered 10 values.
Your values add to 75.8
The average of your values is 7.58
The largest value is 19
The smallest value is 1.4
The variance of your values is 40.68
```

Note: To calculate variance: sum the squares of the differences of each value from the average and divide the sum by one less than the number of values. For the example:

$$(2.3 - 7.58)^2 + (6 - 7.58)^2 + (5 - 7.58)^2 + (8 - 7.58)^2 +$$
$$(13.4 - 7.58)^2 + (19 - 7.58)^2 + (3 - 7.58)^2 +$$
$$(1.7 - 7.58)^2 + (1.4 - 7.58)^2 + (16 - 7.58)^2$$

all divided by 9

3. Helping the crossword puzzle solver (Problem Solving and Programming)

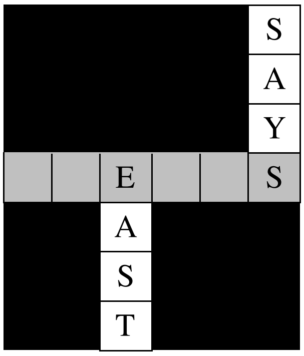
Your programming work in this question will be assessed on two criteria:

(a) Your approach to the problem. We will be looking at your work for evidence that you found good ways of storing the necessary data, and devised algorithms for finding and displaying the requested results. Please hand in any notes and diagrams, which describe what you are attempting to program.

(b) The extent to which your program works and correctly solves the problem.

Crosswords puzzles of all kinds are popular. The finest kind of crossword, in the view of fans, is the ‘cryptic’, where clues are often indirect and obscure. Several types of clue are used in cryptic crosswords. The one that concerns us in this question is the ‘anagram’ – the clue contains a word or words whose letter must be rearranged to find the answer. For example the letters of the word ‘sadder’ can be rearranged to form ‘dreads’.

Your task is to write a program to help a cryptic puzzler to solve anagrams. The first inputs for your program are the letters to be rearranged (S, A, D, D, E, R in our example above). There is another input though. Usually the puzzler knows the positions of some of the letters. Our example might have occurred after some other words had been entered into the puzzle, thus:



So, there must be a second part to the input – the known positions and letters. Given the inputs, your program could produce a list of all possible permutations for the solver to look through.

```
A D E D R S      A D E R D S
A R E D D S      D A E D R S
D A E R D S      D D E A R S
D D E R A S      D R E A D S
D R E D A S      R A E D D S
R D E A D S      R D E D A S
```

Of course the number of possibilities grows very rapidly with the number of letters. One issue to consider is how to avoid having your program get trapped into a loop displaying an absurd amount of output.

Your challenge is to find ways of reducing the number of possibilities displayed and/or find ways of displaying the answers that make it as easy as possible for the solver. You could allow the solver to ‘hypothesise” – for example that the first letter was ‘D’ and show possibilities. You could include some rules – for example only certain pairs of consonants can occur at the start of words. You could take advantage of knowing the words in the answer (puzzles usually state the way an answer is broken into words – a seven letter answer might be specified as 4, 3 meaning a four letter word followed by a three letter one).

(Question 3 Continued Next Page)

Some more examples anagrams:

THE EYES ----- THEY SEE
FIR CONES ----- CONIFERS
ASTRONOMER ----- MOON STARER

Write a program to display permutations of letters. Save a copy of that program to hand in.

Choose some ideas for enhancing the program and implement as much as you can.

Write a brief summary of your ideas and your achievement in implementing them.

NOTE: Use of a computer dictionary (some form of look-up table) would be considered inappropriate. Even if you have one available, you may not use it.

Please hand in printed listings of your programs, and a version of the program on a floppy disk. The version on floppy disk should be a plain text version. You should also hand in any notes, diagrams or explanations you have written. These are particularly important if you do not get your program running correctly as they give us information about your method of approaching the problem. If your program generates output to a file, or to a printer, you should also include printed version of the output.