SQA Advanced Unit Specification

General information for centres

Unit title: Mathematics for Computing 2

Unit code: HR6T 48

Unit purpose: This Unit is about preparing candidates to model real problem situations mathematically and apply advanced mathematical techniques (at pre-calculus level) that can be implemented by computer. The Unit introduces the topics of matrices, series, probability and recursion; and techniques that are applicable to a wide range of problems. Although the emphasis is on application, the mathematical principles are thoroughly and consistently developed. The Unit is primarily designed for candidates who intend to specialise in programming or candidates who intend to proceed to the third year of a Computing degree course.

On completion of the Unit the candidate should be able to:

1. Demonstrate an understanding of matrices and apply matrix methods to problem situations
2. Demonstrate an understanding of series, probability and recursion, and their application to practical problems

Credit value: 1 SQA Credit at SCQF level 8: (8 SCQF credit points at SCQF level 8)

SCQF (the Scottish Credit and Qualifications Framework) brings Scottish qualifications into a single framework of 12 levels ranging from SQA National 1 to doctorates. The SCQF includes degrees; SQA Advanced Certificates/Diplomas; SQA National Qualifications; and SVQs. Each SQA Unit is allocated a number of SCQF credit points at a specific level. 1 SCQF point = 10 hours of learning. SQA Advanced candidates are normally expected to input a further number of hours, matched to the credit value of the Unit, of non-contact time or candidate-led effort to consolidate and reinforce learning.

Recommended prior knowledge and skills: Access to this Unit will be at the discretion of the Centre, however it is strongly recommended that candidates should have achieved the core skill of Numeracy at SCQF level 5 level. Knowledge of Mathematics at National 5 is also desirable. This may be demonstrated by the achievement of appropriate National Units or Courses or by the SQA Advanced Unit HP1H 47: *Mathematics for Computing 1.*

Core skills: This Unit gives automatic certification of the following core skill component:

- *Numeracy: Using Number* at SCQF level 6

Context for delivery: This Unit is included in the framework of a number of SQA Advanced Certificate and SQA Advanced Diploma group awards. It is recommended that it should be taught and assessed within the context of the particular group award to which it contributes.

Assessment: The Unit may be assessed by a single instrument of assessment which would require candidates to apply a range of mathematical techniques to a single problem situation. The assessment may be extended to integrate the assessment requirement of other Units within the framework, eg writing a program to implement the mathematical solution.

An alternative is two smaller assessments, (one for each Outcome), covering a particular aspect of the Outcome to which it applies. In all cases, the assessment instruments must have a reference to or be in the context of computing and presented as a problem situation.

Since the core skill of component *Numeracy: Using Number* at SCQF level 6 is embedded in this Unit, it is strongly recommended that you follow the assessment guidelines given. If you wish to use a different assessment model you should seek prior verification of the assessment instrument(s) you intend to use to ensure that the core skill component is still covered. Please note, candidates must achieve all of the minimum evidence specified for each Outcome, combination of Outcomes, or for the Unit as a whole in order to pass the Unit and achieve the core skill component.

The single combined assessment will generally not be presented in a supervised situation. The candidate will be given the assignment up to two weeks prior to the submission date.

If it is the preferred assessment option, the two separate time-controlled open-book assessments should be presented immediately after the teaching for the corresponding Outcome has been completed. The time duration is at the discretion of the Centre, but a two hours assessment for each Outcome is recommended.

Unit specification: statement of standards

Unit title: Mathematics for Computing 2

Unit code: HR6T 48

The sections of the Unit stating the Outcomes, knowledge and/or skills, and evidence requirements are mandatory.

Where evidence for Outcomes is assessed on a sample basis, the whole of the content listed in the knowledge and/or skills section must be taught and available for assessment. Candidates should not know in advance the items on which they will be assessed and different items should be sampled on each assessment occasion.

Outcome 1

Demonstrate an understanding of matrices and apply matrix methods to problem situations

Knowledge and/or skills

- Matrices as a method of representing ordered data and the relationship with program variable arrays
- The ability to add, subtract and perform scalar multiplication on matrices and multiply two matrices
- The ability to find the inverse of a matrix by elementary row operations
- The transpose of a matrix and index notation
- The ability to apply matrix techniques to a range of applications including solving simultaneous linear equations, application to vector transformation and rotation, maps and graphs
- The ability to relate matrix applications to computing

Evidence requirements

Evidence for the knowledge and/or skills in this Outcome will be provided on a sample basis.

If a single assessment task is adopted to cover the entire Unit then at least eight of the 12 total Unit knowledge/skill items must be covered by the assessment with a minimum of four from this Outcome which must always include items 5 and 6 from each Outcome.

For a single holistic Unit assessment, the candidate will be presented with a single problem two weeks prior to the submission date, or earlier if assessment is combined with other Unit assessments. The assessment instrument should be varied regularly. The assessment will refer to a practical situation relevant to an aspect of computing. The candidate will submit the response as a short report. Guidance on the format of the report and a minimum list of contents must be provided along with the assessment to ensure that the evidence requirements are met.

Alternatively, if this Outcome is tested with one dedicated assessment, a minimum of four of the knowledge/skills items from this Outcome must be tested which must always include items 5 and 6 from this Outcome.

The assessment will be a single problem stated within a practical context and presented in a supervised situation.

The candidate should have no pre-knowledge of the instrument of assessment and the assessment instrument should be varied regularly.

Where an item is sampled, a candidate's response can be judged to be satisfactory where the evidence provided is sufficient to meet the requirements for each item by showing that the candidate is able to:

- Explain the advantages of using matrix notation and how a matrix could be stored and manipulated in the context of programming - the candidate should provide a commentary on a particular application
- Demonstrate the ability to perform matrix addition, matrix subtraction, scalar multiply and matrix multiplication, each on a minimum of one occasion with matrices no smaller than 3x3
- Find the inverse of a 3x3 matrix by elementary row operations; the matrix should have no more than two elements that are zero. Matrices may be selected such that the elements of the inverse matrix are integer - one correct solution is required
- Describe index notation and explain how the transpose of a matrix is obtained; describe the meaning of the scalar product and how it is calculated – the candidate should provide a commentary on a particular application where these operations are used
- Apply matrix techniques to mathematically represent a problem situation and use matrix techniques to develop a solution to the problem – the problem must be non-trivial and relate to the candidate's subject or interest area
- Describe how a matrix model could be implemented by computer or how it is applicable to computing.

Whatever the form of assessment, it is mandatory that it should include data presented in a minimum of two graphical forms and that the candidate is required to effectively present results in a minimum of two graphical forms. The level of difficulty should be set to require candidates to identify the underlying features in graphs with no scale on axes and communicate information in tables, graphs, charts, diagrams or qualitative form.

The candidate should attain a minimum of 60% of the available marks to reach the standard required for a pass. This must include satisfactory evidence for items 5 and 6 in the knowledge and/or skills section of each Outcome

Assessment guidelines

If the preferred assessment method is two individual assessments, one of these must test this Outcome. It is recommended that the time duration for the assessment should be restricted to two hours and that notes, textbooks and **scientific** calculators be made available to the candidate.

Candidates are expected to describe the solution to a single problem and in some cases how a computer could implement it. However, the actual implementation is not a requirement and should only be requested as part of a combined assessment integrated with other Units.

Outcome 2

Demonstrate an understanding of series, probability and recursion, and their application to practical problems

Knowledge and/or skills

- The ability to give a functional expression for a series
- The ability to express a series recursively
- The ability to find the sum of a series
- The ability to use simple probability
- Application of series, probability and recursive techniques to develop a solution to a range of problems
- How series, probability and recursion relate to computing

Evidence requirements

Evidence for the knowledge and/or skills in this Outcome will be provided on a sample basis.

If a single holistic Unit assessment task is adopted to cover the entire Unit then at least eight of the 12 total Unit knowledge/skill items must be covered by the assessment with a minimum of four from this Outcome which must always include items 5 and 6 from each Outcome.

For a single holistic Unit assessment, the candidate will be presented with a single problem two weeks prior to the submission date or earlier if assessment is combined with other Unit assessments. The assessment instrument should be varied regularly. The assessment will refer to a practical situation relevant to an aspect of computing. The candidate will submit the response as a short report. Guidance on the format of the report and a minimum list of contents must be provided along with the assessment to ensure that the evidence requirements are met.

Alternatively, if this Outcome is tested with one dedicated assessment, a minimum of four of the knowledge/skills items from this Outcome must be tested which must always include items 5 and 6 from this Outcome.

The assessment will be a single problem stated within a practical context and presented in a supervised situation.

The candidate should have no pre-knowledge of the instrument of assessment and the assessment instrument should be varied regularly.

Where an item is sampled, a candidate's response can be judged to be satisfactory where the evidence provided is sufficient to meet the requirements for each item by showing that the candidate is able to:

- Give a formula for the nth term of a non-trivial series – two correct responses are required
- Express a non-trivial series recursively – two correct responses are required

- Use summation notation and give the sum of a geometric or arithmetic series – one correct response is required
- Find the probability for a single event and series of independent events – one correct response for each of the two types is required
- Apply series, recursion and probability techniques to mathematically represent a problem situation and develop a solution to a range of problems. In assessing this item, only a single non-trivial problem situation related to the candidate's subject or interest should be presented. The response is deemed acceptable if there is evidence that the methods of either series or recursion or probability have been applied correctly and have contributed significantly to the development of a solution)
- Describe problems involving series, recursion and probability model that could be implemented by computer or how they are applicable to computing. (The assessment should only consider one particular problem).

The candidate should attain a minimum of 60% of the available marks to reach the standard required for a pass. This must include satisfactory evidence for items 5 and 6 in the knowledge and/or skills section of each Outcome

Assessment guidelines

If the preferred assessment method is two individual assessments, one of these must test this Outcome. It is recommended that the time duration for the assessment should be restricted to two hours and that notes, textbooks and **scientific** calculators be made available to the candidate.

Candidates are expected to describe the solution to a single problem and in some cases how a computer could implement it. However the actual implementation is not a requirement and should only be requested as part of a combined assessment integrated with other Units.

Administrative Information

Unit code: HR6T 48

Unit title: Mathematics for Computing 2

Superclass category: RB

Date of publication: August 2017

Version: 01

Source: SQA

SQA acknowledges the valuable contribution that Scotland’s colleges have made to the development of SQA Advanced Qualifications.

FURTHER INFORMATION: Call SQA’s Customer Contact Centre on 44 (0) 141 500 5030 or 0345 279 1000. Alternatively, complete our Centre Feedback Form.

Unit specification: support notes

Unit title: Mathematics for Computing 2

This part of the Unit specification is offered as guidance. The support notes are not mandatory.

While the exact time allocated to this Unit is at the discretion of the Centre, the notional design length is 40 hours.

Guidance on the content and context for this Unit

Candidates should initially be taught the methods and techniques in a general or abstract way, but it is essential that these techniques be applied to a large range of real problems relevant to their subject area prior to an assessment. Sample applications from which the Centre can extract or develop assignments and assessment material are listed below. These may be in association with other Units and thus there is the possibility of combining assessments.

Candidates must be introduced to the processes of research and apply these. Candidates must be able to undertake analysis, synthesis and evaluation. They are encouraged to develop innovative solutions.

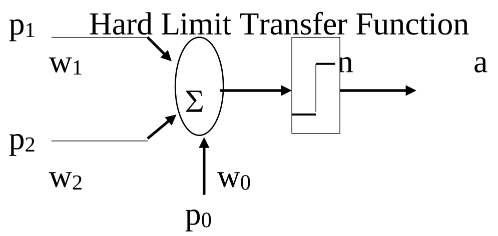
In the examples given, the level of difficulty is variable and candidates can be given assistance. However, the Centre should ensure any assessment created satisfies the minimum requirement of representing a problem in mathematical form and then use an appropriate calculation to develop a solution.

Note that the following examples are not exemplar assessments, but ideas on how the mathematical practices and techniques developed in this Unit could be related to real applications.

APPLICATION EXAMPLES

1. Artificial Neural Networks

A brain neuron can be modelled as shown below.



The n is $\mathbf{w}^T \mathbf{p}$ where **w** and **p** are column vectors and $p_0 = 1$. The hard limit transfer function gives an output $a = 1$ if $n \geq 0$, $a=0$ otherwise. The candidate would be expected to find values for the weights that will linearly separate a set of eight numbers in 2-d space into two distinct classes.

The problem may be extended by using a linear transfer function and introducing LMS training ($\mathbf{w}^{I+1} = \mathbf{w}^I + \mathbf{p}$ * error*Learning Rate). The candidate may then write a program to apply the training points and the expected value over many epochs for a chosen learning rate (0 – 1) to obtain the weight of the separation line.

2. Graphics

The candidate should be shown by trigonometry that a vector may be rotated in 2-d by an angle θ by multiplying by the 2x2 matrix

$$\begin{bmatrix} \text{Cos}\,\theta & -\text{Sin}\,\theta \\ \text{Sin}\,\theta & \text{Cos}\,\theta \end{bmatrix}$$

A method of rotating 2-d figures on the screen should be described and programs written to move objects on the screen in small angular steps to give the impression of continuous smooth movement. Candidate could be asked to find the inverse transformation (both by replacing θ with -θ and finding the inverse of the matrix).

The technique can be taken further by defining the matrix for 3-d rotation (angles θ and φ are specified) and plotting the 2-d projection as a wire outline shape is moved. A screen saver based on this principle could be produced.

3. Data Modelling

The modelling process involves representing a continuous real system as a discrete grid (usually 3-d) with a parameter value (or values) associated with each point. These are matrices or tensors. The behaviour of the real system is estimated by iterating over finite time intervals. With every step, each point is changed by including the estimated effect of its nearest neighbours over that time interval. The model will show how an initial set of conditions develop over time.

The candidate could produce a program to model a practical system and tune the parameters such that the response matches real data.

Example: Heat flow

A steel rod of length 400 cm and at temperature of 20 degrees Centigrade is plunged to half its depth in boiling water. Model how the temperature changes with time.

The method is to label points at 1 cm intervals, with temperature T^I (t) (where I identifies the point and t is the time) to form a finite series. The increase in T over a single iteration for position I is a constant times $T^{i+1} - 2\,T^i + T^{i-1}$. A graph can be produced showing the heat flow.

These methods can be extended to 3-d and could include flow and equilibrium problems

4. Time Series Analysis

The candidate could input up to 100 time series data from a real source into a computer (his/her own program or an application program). S/he could display a moving average and perform a trend analysis to obtain a long range forecast. Statistical analysis functions within the development software should not be used. It is important to evaluate the effectiveness of the prediction. The practical application could be the global ocean temperature over the last 100 years with the data supplied. To evaluate the effectiveness of the prediction, the candidate should compare the prediction based on a subset of the data with what actually occurred.

5. Correlation

A computer may be used to store pairs of co-ordinate points. Large data sets could be supplied as a file (at least 100 data with a diverse range such as, for example, the apparent brightness and red-shift of stars or galaxies, the weight and height of a representative group of people, etc).

The objective is to develop an algorithm to find the best fit line (minimising the sum of the perpendicular distances squared, regression) and display the line details and the correlation coefficient.

The candidate should compare the program to a standard package such as MINITAB or the EXCEL which have these functions built-in, and evaluate the results.

6. Route Planner

The candidate may be presented with a geographical map (perhaps of the local area or town). A distance digraph should be generated. Implementing an appropriate algorithm, the candidate should produce a program to determine the best route between two places. The effectiveness of the program should be evaluated and methods of incorporating factors such as busy junctions or poor roads into the calculation should be considered (but not necessarily implemented).

7. Encryption

One method of encryption is to represent the letters by the numbers 1 to 26 (0 for a space). The string

TRY TO DECODE THIS

Is equivalent to

20 18 25 0 20 15 0 4 5 3 15 4 5 0 20 8 9 19

This can be written as a 3 x 8 matrix (padding with zeros):

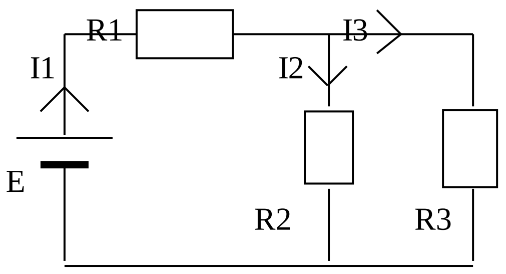
$$\mathbf{S} = \begin{bmatrix} 20 & 18 & 25 & 0 & 20 & 15 & 0 & 4 \\ 5 & 3 & 15 & 4 & 5 & 0 & 20 & 8 \\ 9 & 19 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

This can be encoded by multiplying by a 3 x 3 matrix **A** which has the property of integer entries only and $\mathbf{A}^{-1}$ has only integer entries. The candidate could find such a matrix and write a program to code and decode this message.

Evaluate the effectiveness of this message – how could a person determine the matrix structure (it does not have to be 3 x 8) and the decryption matrix?

Investigate the DES encryption algorithm and how it is implemented.

8. Electrical Networks



The circuit above has the following equations:

$$I1 - I2 - I3 = 0$$
$$E - I1.R1 - I2.R2 = 0$$
$$I2.R2 - I3.R3 = 0$$

For R1 = 10, R2 = 50, R3 = 100, the candidate should determine the current flows through the circuit.

9. Balancing Chemical Equations

The candidate should determine what proportion of each substance balances the equation

$$C_2H_6 + O_2 \rightarrow CO_2 + H_2O$$

Method

Write

$$x_1\, C_2H_6 + x_2\, O_2 \rightarrow x_3\, CO_2 + x_4\, H_2O$$

Obtain linear equations balancing the number of carbon, hydrogen and oxygen atoms. Solve by elementary row operations.

10. Transmission Error Detection and Correction: Hamming Codes

There exist a class of problems where the matrix entries are binary digits (0, 1). For such systems, arithmetic operations are **modulo 2** (i.e. 1 + 1 = 0 not 10).

The Hamming (7,4) code transforms 4 bits of data into 7. If the data bits are $\mathbf{B}^T$ = (b1, b2, b3, b4) and the check codes are $\mathbf{C}^T$ = (c1, c2, c30, the message sent is:

$$\mathbf{M}^T = (c1, c2, b1, c3, b2, b3, b4)$$

The check codes are obtained with the matrix:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{C} = \mathbf{H} \text{ x } \mathbf{B}$$

If

$$\mathbf{E} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

The transmitted message has the following property:

$$\mathbf{E} \text{ x } \mathbf{M} = \mathbf{0}$$

By multiplying the received message by E, the candidate should demonstrate that a single error can be detected and corrected.

For more complex error checking, the candidate should consider at how a CRC is generated and implemented for an X25 Ethernet packet.

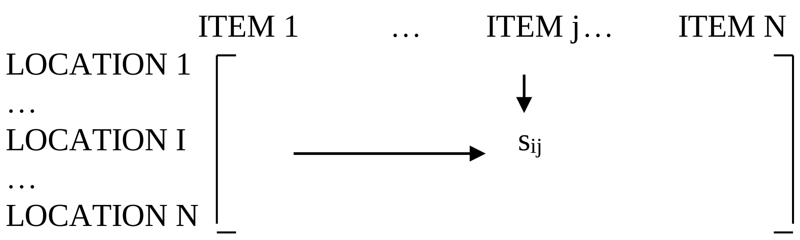
11. Stock Cost Calculations

A company has N separate offices each with M different items of stock, where M and N are any integers.

The candidate should develop a matrix to calculate the stock cost and stock price and implement this using a computer. Test with generated data sets.

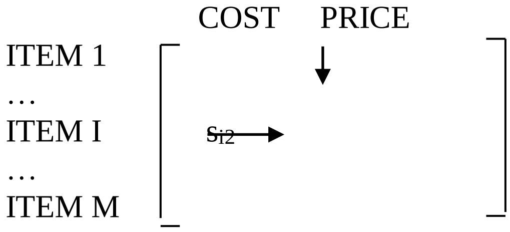
Outline Solution

The stock matrix **S** has elements s_{ij} where the matrix has the form:



Where s_{ij} is the stock level of item j at location i.

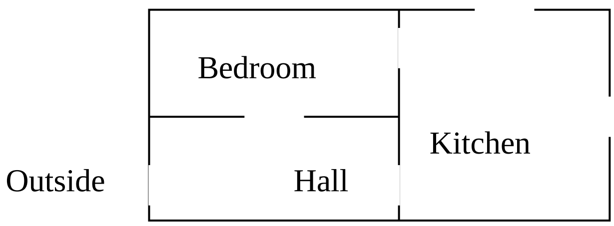
A cost matrix **C** with elements c_{i1} and c_{i2} is similarly defined:



These may be implemented as arrays. Multiply **S** times **C**. The candidate should consider what is the resultant multiplied by to give a final cost and price total for the organisation? Does the matrix multiplication order matter?

12. Plans and Maps

The ground plan of a house is shown below:

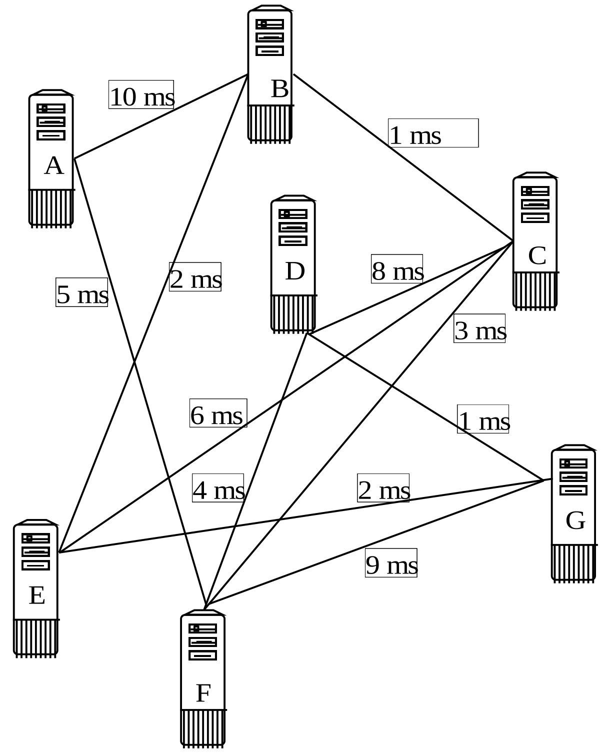


The candidate should draw an equivalent graph and decide if it is possible to walk through the house passing through each door only once?

As an extension to this type of problem, the candidate may be asked to draw a graph showing the interconnection between the counties of Scotland. Can a tourist organise a route through these counties to visit each once and arrive back at the starting point?

13. Computer Networks

Seven servers are connected as shown in the diagram. The connections are marked with the average time taken to transfer a packet of data.



The candidate should produce incidence, adjacency and cost matrices.

Is the graph planar; Eulerian; Hamiltonian?

They may be asked to specify a general algorithm for establishing the best route between any two points (not by inspection).

The algorithm may be developed further by considering that the time between nodes is not static but dependent on the traffic flow.

14. Matrix Power Series

Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

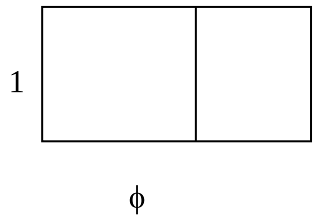
Let $\mathbf{v} = (1, 2)^T$. Consider the series f(n) where $f(n) = \mathbf{A}^n \times \mathbf{v}$.

The candidate should generate the first 6 terms of the series and define the series recursively.

Solution

$\mathbf{F}(n) = \mathbf{F}(n-1) + \mathbf{F}(n-2)$: $\mathbf{F}(1) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$; $\mathbf{F}(2) = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

This is a matrix form of the Fibonacci sequence which recurs in biological systems. The candidate could then explore the relationship between f(n) / f(n-1) as n tends to infinity and the **golden ratio** ϕ:



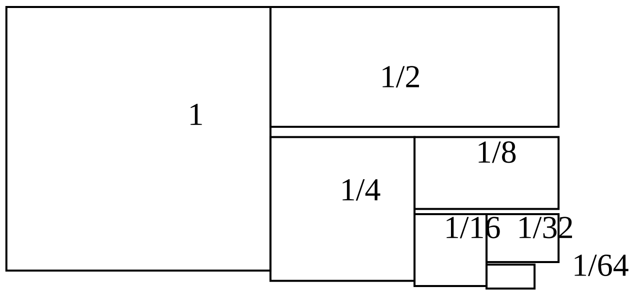
If the rectangle is divided onto a square and a rectangle as shown, the sides of the smaller rectangle are in the same ratio as the original rectangle, and so on as the procedure is repeated.

The candidate should develop a series whose sum equal to the area, ϕ, of the rectangle.

One other aspect of interest is to represent the following geometric series graphically and thus deduce the sum as the number of terms tends to infinity.

$1 + ½ + 1/4 + 1/8 + \ldots$

Solution



This shows the first 7 terms of the series. It is clear that the sum of the series tends to 2 as the number of terms increase.

15. Series Approximation of Complex Functions

Sin x is the perpendicular height of a Unit vector after a rotation of x radians anti-clockwise from the horizontal. The actual value can be measured but it can also be derived mathematically by adding the terms of the following infinite series:

$x, -x^3/3!, x^5/5!, -x^7/7!, x^9/9!, \ldots$

The candidate should state the n^{th} term of the series.

They may write a program to iteratively calculate the sine of an angle (use a recursive function call to calculate the factorial). How is it decided how many terms to take?

16. Fourier Transforms

Data transmission is based on the fact that sinusoidal waves if different frequencies can be superimposed and sent together down a transmission line. The total wave will have a complex shape, but at the other end, a signal at a particular frequency can be extracted. If the amplitude is varied, this variation can be detected (although there are complications with noise and attenuation) at the other end in spite of the other signals on the same channel. This is called amplitude modulation.

A specific frequency can be extracted by hardware using a resonance circuit comprised of a capacitor and inductor – a radio receiver, but the signal can be extracted mathematically by performing a Fourier transform.

To illustrate that any signal shape can built up from waves, the candidate should write a program (or use a spreadsheet) to produce a plot of A_1 cos x where A_1 is any number and x ranges from 0 to 4π (720^o). The infinite series below should be generated:

1 cos x, 1/3 cos 3x, 1/5 cos 5x, 1/7 cos 7x ..

The candidate should be asked to add the terms together at each point x and display the result (allowing the number of terms to be selectable).

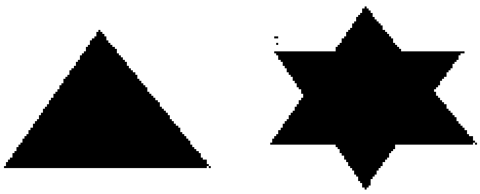
If a computer is sending the digital data 0101 down a coaxial network cable at 2 Mbaud, the consequences if the line will only carry signals up to a frequency of 10 MHz should be determined?

17. Fractals

A fractal has the property of "self-similarity' – small parts of the shape are very similar to the whole shape itself.

Fractals are generated by functions, but they can be created geometrically, one example being as follows: take an equilateral triangle and add smaller copies to each side. Keep repeating for n steps:

n = 1 n=2 n=…..



With more and more steps, the pattern resembles that of a snowflake. The candidate should develop two series, one with the increased area and the other with the increased perimeter per step.

Using a computer or otherwise, the total perimeter and area as n tends to infinity should be found.

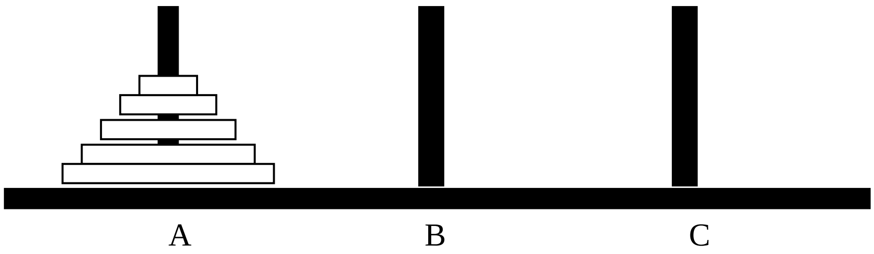
The candidate may next compare this with a similar problem where a square is divided into 9ths and the centre square filled. In the next step, each smaller square is divided into 9ths with the centre one filled. What is the total filled area as the number of steps tends to infinity?

The candidate may proceed to look at how fractal images are generated and their applications in game graphics.

18. Tower of Hanoi Game

The Tower of Hanoi game consists of a set of disks of different size that are stacked in three columns but no disk can sit on one of smaller diameter.

The purpose of the game is illustrated by the following 5-disk diagram where column A has to be moved to column C by a number of moves. The candidate should find the minimum number of moves for a 5-disk game.



The candidate should then express the number of moves for an n-disk game recursively without trying different numbers of disks. The closed form solution should also be given.

Solution

To move a column of 5 disks is made up of three distinct stages: Move the top 4 disks from A to B; Move the largest disk from A to C; move the four disks in B to C

Generalising: HANOI(n) = 2 x HANOI(n-1) +1
HANOI(1) = 1

The closed form solution is: HANOI(n) = $2^n - 1$

19. Euclid's Algorithm

Euclid's Algorithm states that:

$$\gcd(M, N) = \begin{cases} \gcd(M \bmod N, N) & \text{if } M > N \\ \gcd(M. N \bmod N) & \text{if } N>=M \end{cases}$$

$$\gcd(N, 0) = N$$

If gcd is the 'greatest common divisor' the candidate should verify with some examples that the algorithm works.

A program should be written to find the greatest common divisor of any two numbers and test with suitable data.

As an additional activity, the candidate should look at Euler's number e: Considering the series 1, 1/1!, 1/2!, 1/3!, .. , 1/n!, .. what is the sum of these numbers as n tends to infinity?

20. Programming Algorithms

The candidate should write two programs to calculate the factorial of a number. One program should adopt an iterative approach; the other using recursion.

SPEED, COMPLEXITY OF PROGRAM, STACK USAGE and other relevant issues should be compared. Classes of problems where recursion is the better implementation choice and others where iteration is the better approach should be defined.

Finally, he/she could consider the application of recursion to sorting of lists.

21. Lighthouse

An observer on a ship notes the times in seconds at which flashes from a lighthouse were visible. The first 7 measured points are indicated below:

3, 5, 15, 17, 27, 29, 39

The candidate should define the recursive step and the stopping condition(s).

Solution

LIGHT(n) = LIGHT(n-2) + 12
LIGHT(1) = 3; LIGHT(2) = 5

22. Games of Chance

£1,000 is hidden behind one of three doors. A contestant pays £400 in advance to guess where the money is. If he guesses correctly, he gets the money. Immediately after a door is selected, the presenter (who knows in advance where the money is) opens another door which does not have the money behind it and offers the contestant the possibility of changing their choice for £200. A contestant rejects this option reasoning they already have a 50% chance of winning, so why pay more? Using the laws of probability, the candidate should demonstrate that this reasoning is faulty.

Solution

There are only three possibilities after the door is opened; if the contestant changes, two are favourable. The probability of success is therefore 2/3. Accepting this choice, the contestant wins an average of £40 on each game.

23. Bacteria Culture

Starting with 1 cell, there is equal probabilities of the cell dying, doing nothing, replicating or making 2 copies in the next time interval. The candidate should produce a program to calculate how the population develops with time.

He/she should be asked to calculate the probability of the cell culture becoming extinct.

24. Binomial Theorem

An entrepreneur purchases 10 computer base Units with motherboards that failed quality checks for a total of £1,000. Specifically, there is a 0.3 probability of failure in the first year. He sells them on with a guarantee that cost be refunded if the machine fails within one year.

The candidate should consider the price he should sell them for to ensure a 75% chance of making £2,000 profit? What is the probability of making a loss at this price?

(Find the binomial coefficients using Pascal's triangle and use a spreadsheet to investigate the possibilities)

Some Centres may choose specialised applications in the fields of signal processing, multiprocessor systems, business/economics, game theory etc, or look at specific complex problems such as finite state machines, Markov chain, linear programming, properties of the magic square, permutations and scheduling or any subject already familiar to the candidate.

Guidance on the delivery and assessment of this Unit

When teaching the first Outcome, it would be beneficial to use pseudo-code to show how matrices can be multiplied within nested loops. It should be emphasised that the multiplication of matrices is non-commutative (order matters). Candidates should understand the significance of the identity matrix.

In some cases the assessment deliverable will be a computer program. The program need not be complex, but will have to accept data input data and implement the appropriate mathematical method(s). Candidates may be supplied with program templates.

The programs should be documented to demonstrate the algorithm used. It is desirable that candidates be encouraged to evaluate the effectiveness of the algorithm. The assignments may be smaller versions of standard applications and thus show candidates how such programs are constructed.

Candidates are encouraged to use programs such as MathCad for presentation and calculation. A variety of specialised application programs for modelling and calculation should be available to candidates, particularly if the underlying grid structure for data storage and editing is accessible. The objective is to show candidates how such applications are constructed.

Another category of assignment is the application of a technique to solve a problem in the field of computing. In this case the deliverable is a short report where the application of the mathematical technique(s) is included and the results evaluated. Achievement is evidenced by the criteria being achieved for each deliverable. The level of difficulty of the mathematical method that is acceptable is described in each Outcome.

The Unit may be assessed with two individual assessments, one for each Outcome, or one single assessment where the knowledge/skills items are sampled. One essential aspect to the design of the assessment(s) is that the Numeracy core skill component “Using Number” be assessed at SCQF level 6.

Open learning

If this Unit is delivered by open or distance learning methods, additional planning and resources may be required for candidate support, assessment and quality assurance.
A combination of new and traditional authentication tools may have to be devised for assessment and re-assessment purposes. For further information and advice, please see *Assessment and Quality Assurance for Open and Distance Learning* (SQA, February 2001 — publication code A1030).

Equality and inclusion

This unit specification has been designed to ensure that there are no unnecessary barriers to learning or assessment. The individual needs of learners should be taken into account when planning learning experiences, selecting assessment methods or considering alternative evidence.

Further advice can be found on our website www.sqa.org.uk/assessmentarrangements.

General information for candidates

Unit title: Mathematics for Computing 2

This Unit teaches two important mathematical techniques that can be implemented by computer to provide you with powerful tools for computer analysis in many different situations. All the teaching relates to practical problems that are either in the context of computing or will benefit from computer analysis.

The first Outcome introduces matrices and shows you how problem situations can be represented by matrices. You will then use elementary row operations to find a solution to the problem.

The second Outcome deals with probability, series and recursion and their application to practical problems.

The material is set at pre-calculus level and you need to have achieved a mathematics qualification at National 5 before starting the Unit. Alternatively, you should have completed the SQA Advanced Unit HP1H 47: *Mathematics for Computing* 1.

There will be one project-type assessment at the end of the Unit or two short assessments, one for each Outcome, each of approximately two hours duration. The assessments will relate to a practical problem.

The mathematics and problem solving skills acquired with the completion of this Unit would be very beneficial if you wanted to go on to complete a degree in computing.