Section A: Multiple Choice Questions

A1 C  Not Broadcast, not necessarily synchronous or radio
A2 B  Definition
A3 C  Standard cell references and operator priority
A4 C  Definition
A5 D  Relative/Mixed references: formula changes to =$B3+B$3 on copying
A6 C  Invalid cell reference
A7 A  Doh! Isn’t that what a firewall is for? ;-)
A8 D  See, sometimes “None of the above” is the right answer!
A9 A  Absolute references $B$1, unaffected by copying to a new cell
A10 D  Definition. There are only a few wildcard characters, easy to learn.
A11 C  Definition.
A12 D  Understanding of how RFID tags work
A13 B  Definition. ASCII codes were often stored and transmitted as an 8-bit byte, but the extra bit was used for error detection purposes (parity check)
A14 A  Understanding of the concept redundancy in databases
A15 C  Understanding of how DHCP works
A16 C  Understanding of some AI concepts
A17 A  Understanding of the concept “data types” in databases
A18 C  Understanding of a typical SQL statement
A19 C  Equivalent to  \( (4 \times (9 \wedge 1)) / 2 \)
A20 A  Definition.
A21 B  C generates a list of indexes only, A and D are invalid VBA
A22 C  Loop semantics. On the last iteration \( x \) becomes 5 and \( y \) is the sum of all integers up to an including the final value of \( x \)
A23 D  Definition (de Morgan’s Law). Try substituting “rich” for \( \text{blnA} \) and “happy” for \( \text{blnB} \). If you’re either rich or unhappy and I’m not, then I must be not rich but (= and) happy.
A24 C  Definition and lecture examples. Remember BadFib?
Section B: Short Answer Questions

B1. (2 marks) Describe the two most important advantages of using Distributed/Grid Computing.

- It allows us to solve massive computational problems by using large numbers of heterogeneous computers (and other related resources), located in different places belonging to different users, over a network(s).
- It could be used to provide increased reliability and performance.

B2. (2 marks) What is the main function of DNS?

DNS (short for Domain Name System) translates Hostnames (human readable computer names on the Internet) into IP addresses (which are often hard to remember) and vice versa.

B3. (2 marks) Briefly outline two applications of Public Key systems.

- Securing a message (using encryption/decryption)
- Message authentication (using digital signatures)

B4. (1 mark) Provide one major difference between synchronous and asynchronous communication.

Synchronous communication requires that both the sender and receiver are active at the same time. This is not required in asynchronous communication, where the sending and the receiving could occur at different times.

B5. (3 marks) What is the purpose of conditional formatting in a MS Excel spreadsheet? Give an example in your answer.

Conditional Formatting in MS Excel allows a user to apply formatting to a cell(s) depending on the value of the cell or the value of a formula.

For example, if a value of the cell is less than 50, we want to change the background colour of that cell to red colour. We could achieve this goal using the conditional formatting feature in MS Excel.

B6. (1 mark) Provide one major difference between ASCII and Unicode encodings.

Unicode uses 16-bit representation and supports all languages. ASCII uses 8-bit representation (or 7-bit for original ASCII) and does not support all the languages.
B7. (2 marks) Provide two major differences between sequential computing and parallel computing.

- In sequential computing, only one instruction is executed at a time. In parallel computing, multiple instructions are executed at a time.
- In general, it is easy to design software for a sequential hardware, but relatively difficult to design software for a parallel hardware.

B8. (2 marks) Briefly outline two different approaches we can use to solve single variable equations in MS Excel.

- Graphical approach, by plotting a graph and looking for a root value
- Using Goal Seek or Solver

B9. (2 marks) In MS Excel, what does the tool named “Descriptive Statistics” offer?

The tool named “Descriptive Statistics” in MS Excel allows a user to quickly calculate many commonly used statistical functions for a given data set and produces a useful report.

B10. (5 marks) You have been asked to develop a spreadsheet to store and analyse marks for students in a given course. You need to store the following marks for each student in a course: Assignment_1, Assignment_2 and Exam. Assignment_1 and Assignment_2 marks are out of 25 each, and Exam marks are out of 50. Assume that there are 30 students in the course.

You need to do the following analysis:

- Calculate Total marks using the following formula:
  Total = Assignment_1 + Assignment_2 + Exam
- Find the top 10 students, based on Total marks
- Find students whose Exam marks are greater than the average Exam marks for the entire class.

Draw a diagram illustrating the design of your spreadsheet application. Indicate any formulas or features of MS Excel that you would use.

Answer:

- In the diagram below, enter the formula “=A2 + B2 + C2” into the cell D2, copy the cell D2 and paste into cells D3 to say D31 to calculate Total marks for all the students
- Find top 10 students using a “Filter” available in MS Excel, or sort the data and select the first 10.
- Find an average Exam marks using an average function available in MS Excel, and store it into a cell (say E2). Now use the “conditional formatting” feature to highlight students whose marks are greater than the average Exam marks (available in the cell E2).

(continued over...)

3
B11. (2 marks) Explain the difference between the concepts of “Expert Systems” and “Machine Learning Systems”, in Artificial Intelligence.

In a process of building an intelligent system, “Expert Systems” try to capture and use knowledge from an expert(s), while “Machine Learning Systems” try to discover knowledge from observations of the world.

B12. (2 marks) Briefly explain the role of a “digital certificate” in Public Key Infrastructure (PKI).

Digital certificates are issued by a certification authority to associate a public key (and other relevant information) with the corresponding owner of the certificate. A digital certificate typically includes information like the name of the owner, owner’s public key, serial number, expiration date, signature of trusted third Party, etc.
**B13. (4 marks)** Assume the active worksheet contains a square matrix of numbers in the upper left corner similar to that shown below. Write statements that find the largest value in any cell on the major diagonal (these cells are shown shaded: A1, B2, C3 etc), and assigns it to the variable \( \text{diagmax} \).

The first empty cell on the diagonal (in this case F6) terminates the process. Declare any other variables you need.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>29</td>
<td>49</td>
<td>49</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>79</td>
<td>36</td>
<td>67</td>
<td>77</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>95</td>
<td>59</td>
<td>28</td>
<td>89</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>83</td>
<td>68</td>
<td>96</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>31</td>
<td>20</td>
<td>10</td>
<td>16</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Answer:**

```vba
Dim pos As Integer
Dim Max As Double

diagmax = ActiveSheet.Cells(1, 1)
pos = 2
While ActiveSheet.Cells(pos, pos).Value <> ""
    If ActiveSheet.Cells(pos, pos).Value > diagmax Then
        diagmax = ActiveSheet.Cells(pos, pos).Value
    End If
End If
pos = pos + 1
Wend
```

*Note that finding the largest (or smallest etc) starts with the first possible value, which is updated selectively as you go through the other elements.*

**B14. (2 marks)** Explain how to place a spin button on a sheet and allow it to control a cell on the sheet (say D5). What restrictions are there on the value displayed?

Select the spin button icon from the control toolbox, click and drag onto the sheet, set the `LinkedCell` value in the Properties window to the cell to be controlled and adjust the `Min` and `Max` properties.

The controlled value must be an integer.
**B15.** *(3 marks)* Consider the following function, which is intended to determine whether a string is palindromic (exactly the same when written backward as forward). It has one syntax error (that Debug/Compile would find), one semantic error that prevents the function from returning anything and one logical error. Identify each error and describe it in a few words only.

```
' returns True if str is palindromic, else False
Function IsPal(str As String) As Boolean
  Dim pos As Integer  ' next position in str
  Dim c1 As String         ' single characters
  Dim slen As Integer  ' length of str

  ' Alg: match character with the one at opposite end
  slen = Len(str)
  For pos = 1 To slen \
    c1 = Mid(str, pos, 1)  ' character at pos
    c2 = Mid(str, slen-pos, 1)
    ' character at corresponding position from end
    If c1 <> c2 Then
      IsPal = False
      Exit Function
    End If
  Next pos
End Function
```

Error 1 (syntax): the variable \( c_2 \) is not declared

Error 2 (semantic): the function has no return value if the characters always match

Error 3 (logical): the calculation of the matching character position is wrong: \( \text{slen-pos} \) should be \( \text{slen-pos+1} \)
B16. (2 marks) In the space below write statements that extract the character exactly in the middle of a string `str`, provided the string has an odd number of characters, and assigns it to `strMiddle`. If `str` has an even number of characters assign the empty string instead.

Examples:  
- `str = "car"`  `strMiddle = "a"
- `str = "cart"`  `strMiddle = ""`

Answer:

```vba
If Len(str) Mod 2 = 0 Then ' even, no middle char
    strMiddle = ""
Else
    strMiddle = Mid(str, Len(str) \ 2 + 1, 1)
End If
```

A slightly trickier alternative solution is the single statement

```vba
strMiddle = Mid(str, Len(str) \ 2 + 1, Len(str) Mod 2)
```

which relies on the fact that you can specify a zero length to `Mid`, in which case it doesn’t matter where the empty string is extracted.
Section C: Programming Questions

While you only need to complete 3 programming questions in the exam, you should obviously attempt all of them in this sample paper for practice.

C1. The Harvard Step Fitness Test estimates the fitness of an athlete by the following procedure: the athlete steps up and down from a standard gym bench once every two seconds for 5 minutes. After this the athlete's pulse (in integral beats per minutes) is taken after 1 minute, 2 minutes and 3 minutes. Call these values \( p_1, p_2, \) and \( p_3 \). The fitness score \( s \) is calculated using this formula

\[
s = \frac{30000}{p_1 + p_2 + p_3}
\]

A 16-year old female athlete is considered to have an Excellent fitness if \( s \) is 86 or more, and Poor fitness if \( s \) is less than 50.

Write a VBA Function that accepts the three pulse rates and returns one of the strings "Poor", "OK", or "Excellent", according to the calculated score. You may use the header as shown below.

```vba
Function Harvard16Female(p1 As Integer, p2 As Integer, p3 As Integer) As String
    Dim s As Double
    s = 30000 / (p1 + p2 + p3)
    If s >= 86 Then
        Harvard16Female = "Excellent"
    ElseIf s >= 50 Then
        Harvard16Female = "OK"
    Else
        Harvard16Female = "Poor"
    End If
End Function
```

Remarks: No tricky stuff here. Just calculate and classify using a couple of conditions. Marks would be deducted for
- not declaring \( s \)
- using the wrong division operator
- using incorrect or redundant Boolean expressions
- not returning the string correctly, say by assigning to a local variable or just using Debug.Print or something

Note: In practice, the literals would be defined constants, but the exercise has been simplified for exam purposes.
C2. Write a VBA subprogram that writes a table of squares and cubes in rows 2 to 21 of the active sheet, the first few rows of which are show below:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>n</td>
<td>n^2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>36</td>
</tr>
</tbody>
</table>

Const MAX = 20  
Const START_ROW = 2

Sub DrawSqrCubeTable()
    Dim row As Integer
    Dim n As Integer  ' Long is probably better

    row = ROW_START
    For n = 1 To MAX
        ActiveSheet.Cells(row,1) = n
        ActiveSheet.Cells(row,2) = n^2
        ActiveSheet.Cells(row,3) = n^3
        row = row + 1
    Next n
End Sub

Remarks: Similar to the factorial lab exercise. Separate counting the rows from calculating data values, even when they’re closely related.

Marks would be deducted for:
- poor procedure structure
- incorrect loop
- wrong cell references
- no constants
C3. Many functions can be approximated by their Taylor Series expansion. The expansion of \( \sin x \) is given by the formula

\[
\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots
\]

For example, this is the approximation using the first four terms of the formula:

Write a VBA function that approximates \( \sin x \) using a set number of initial terms of the series as specified by a parameter. *Hint*: consider how each term is derived from the previous one.

**Discussion:** each term changes sign and the numerator is multiplied by \( x^2 \). The denominator is multiplied by two adjacent integers, depending on which term we’re up to. Starting from the \( x^3 \) term, the required products are:

Term 1: \( 3 \times 2 \)  
Term 2: \( 5 \times 4 \)  
Term 3: \( 7 \times 6 \)

Given the term counter \( t \), the two numbers must be \( 2t \) and \( 2t+1 \)

Hence suitable pseudocode would be

```
Set sum and term to the initial term x  
Set term number t to 1  
For as many terms as we need  
    \[ \text{term} = - \frac{\text{term} \times x^2}{(2t \times (2t+1))} \]  
    sum = sum + term  
    t = t + 1  
Return sum
```
Converting this to VBA is pretty straightforward. \( t \) can be controlled by a \texttt{For} statement:

```vba
Function MySin(x As Double, numTerms As Integer) As Double
    Dim t As Integer
    Dim term As Double, sum As Double
    term = x: sum = x ' first term
    For t = 1 To numTerms - 1 ' note the -1
        term = -term * x^2 / (2*t*(2*t+1))
        sum = sum + term
    Next t
    MySin = Sum
End Function
```

\textit{Note:} This is probably the most difficult of the programming questions, but within the expected capabilities of most students.

\textbf{C4.} Write a VBA subprogram that draws a thin red cross using the two diagonal borders in all empty cells in row 1 of the active sheet. Your subprogram should consider all 256 columns.

To help you to complete this task, this is what the Macro Recorder saves when the user puts a applies a thin red border to the left of cell E15:

```vba
Range("E15").Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = 3
End With
Selection.Borders(xlEdgeTop).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
```

Your subprogram should manipulate the cells directly rather than use the Selection object.

\textit{(Answer overleaf)}
Const MAX_COLS = 256

Sub RedX()
    Dim col As Integer

    For col = 1 To MAX_COLS
        If ActiveSheet.Cells(1,col).Value = "" Then
            With ActiveSheet.Cells(1,col)
                With .Borders(xlDiagonalDown)
                    .LineStyle = xlContinuous
                    .Weight = xlThin
                    .ColorIndex = 3
                End With
                .Borders(xlDiagonalUp) = .Borders(xlDiagonalDown)
                .Borders(xlEdgeTop).LineStyle = xlNone
                .Borders(xlEdgeBottom).LineStyle = xlNone
                .Borders(xlEdgeLeft).LineStyle = xlNone
                .Borders(xlEdgeRight).LineStyle = xlNone
            End With
        End If
    Next col
End Sub

Remarks: The core idea is picking the right detail out of the sample code. Cell processing is very simple, though it’s OK to repeat the code for the second diagonal border rather than to copy the properties in one assignment statement as shown. Mark deductions:

- poor procedure structure, missing declarations
- no use of With
- use of Selection object (implying that you don’t know how to apply these changes to objects directly, minor deduction only)
- wrong border references
- forgetting to do the empty cell test
- poor loop structure
- no constants
- simply copying the code given without showing any understanding at all (100% deduction!)
- etc etc