basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

SENIOR CERTIFICATE EXAMINATIONS/ NATIONAL SENIOR CERTIFICATE EXAMINATIONS

INFORMATION TECHNOLOGY P1

2023

MARKING GUIDELINES

MARKS: 150

These marking guidelines consist of 24 pages.

GENERAL INFORMATION:

- These marking guidelines are to be used as the basis for the marking session. They were prepared for use by markers. All markers are required to attend a rigorous standardisation meeting to ensure that the guidelines are consistently interpreted and applied in the marking of candidates' work.

- Note that learners who provide an alternate correct solution to that given as example of a solution in the marking guidelines will be given full credit for the relevant solution, unless the specific instructions in the paper was not followed or the requirements of the question was not met

- **Annexures A, B, C** and **D** (pages 3 to 10) include the marking grid for each question.

- **Annexures E, F, G** and **H** (pages 11 to 24) contain examples of solutions for Question 1 to Question 4 in programming code.

- Copies of **Annexures A, B, C, D** and **the summary for the marks of the learner** (pages 3 to 10) should be made for each learner and completed during the marking session.

ANNEXURE A

QUESTION 1: MARKING GRID – GENERAL PROGRAMMING SKILLS

CENTRE NUMBER:	**EXAMINATION NUMBER:**		
QUESTION	**DESCRIPTION**	**MAX. MARKS**	**LEARNER'S MARKS**
1.1	**Button [1.1 – Formatting]** Set font size of edtQ1_1 to 14 ✓ Set text of edtQ1_1 to 'Hello world' ✓ Set font style of edtQ1_1 to underline ✓ Set colour of edtQ1_1 to green ✓	4	
1.2	**Button [1.2 – Random number]** Generate a random number ✓ in the correct range (1 to 99) ✓ Check if ✓ random number is a single digit (<=9) ✓ Display the number converted to string ✓ and a message indicating a single-digit in pnlQ1_2 ✓ Else ✓ Display the number converted to string and a message indicating a double-digit in pnlQ1_2 ✓	8	
1.3	**Button [1.3 – Area]** area = (3*sqrt(3) ✓ /2) ✓* sqr(rSide) ✓ - pi* ✓sqr(rSide/2) ✓ Correct values in formula ✓ Display the value of area ✓ to one decimal ✓	8	
1.4	**Button - [1.4 – Find]** Extract edtQ1_4.text ✓ Create and initialise counter ✓ While NOT EOF(text file) do ✓ Read sLine from the file ✓ If sLine = edtQ1_4.text ✓ (or variable) Ignore case ✓ Increment counter ✓ If counter > 0 ✓ Display occurrences ✓ Else ✓ (OR if counter = 0) Display “Word not found” ✓	11	

1.5	**Button [1.5 – Booster rocket]** Create and initialise counter ✓ Get total fuel from input box converted to integer/real ✓ Loop while total fuel > 200 ✓ Increment counter ✓ fuel used = total fuel / 100 ✓ * 7.5 ✓ total fuel = total fuel – fuel used ✓ Display counter, fuel used and total fuel left ✓ in neat columns ✓ formatted to two decimal places	9	
	TOTAL SECTION A:	**40**	

ANNEXURE B

QUESTION 2: MARKING GRID – SQL AND DATABASE PROGRAMMING

CENTRE NUMBER:	**EXAMINATION NUMBER:**		
QUESTION	**DESCRIPTION**	**MAX. MARKS**	**LEARNER'S MARKS**
2.1	**SQL statements**		

2.1.1	**Button [2.1.1 – iOS products]**
	`SELECT DeviceID, DeviceName ✓`
	`FROM tblDevices ✓`
	`WHERE OperatingSystem = "iOS" ✓`
		3	

2.1.2	**Button [2.1.2 – Category selected]**
	`SELECT DeviceName, Category, NumInStock`
	`FROM tblDevices ✓`
	`WHERE Category LIKE ✓ "% + sDeviceType + " ✓`
		3	

2.1.3	**Button [2.1.3 – Online support]**
	`SELECT DeviceName, Category, OperatingSystem ✓`
	`FROM tblDevices D, tblManufacturers M ✓`
	`WHERE D.ManufacturerID = M.ManufacturerID ✓`
	`AND ✓ OnlineSupport = True ✓`
	`ORDER BY DeviceName ✓`
		6	

2.1.4	**Button [2.1.4 – Profit per manufacturer]**
	`SELECT ManufacturerID,`
	`FORMAT(SUM ✓ (NumInStock *`
	`(Price * 0.6)) ✓, "CURRENCY") ✓`
	`AS [Profit] ✓`
	`FROM tblDevices ✓`
	`GROUP BY ManufacturerID ✓`
		6	

2.1.5	**Button [2.1.5 – Remove devices]**
	`Delete ✓ * FROM tblDevices ✓`
	`WHERE Category = "Smart speaker" AND ✓`
	`ManufacturerID = "M104" ✓`
		4	

	Subtotal:	**22**	

QUESTION 2: MARKING GRID (CONT.)

2.2	**Database Manipulation**		
2.2.1	**Button [2.2.1 – Display products]** Go to the fist record in the tblManufacturers ✓ Use a loop to step through tblManufacturers ✓ Display the ManufacturerName and ContactNumber ✓ Display the DeviceName, Instock and Price as headings ✓ Go to the fist record in the tblDevices tabel ✓ Use a loop to step through tblDevices ✓ Test if (tblManufacturers ['ManufacturerID'] = tblDevices['ManufacturerID']) ✓ Display DeviceName, NumInStock, converted to a string, ✓ and Price, converted to currency ✓, in redQ2_2_1 ✓ tblDevices.Next ✓ End loop (tblDevices) tblManufacturers.Next ✓ End loop (tblmanufacturers)	12	
2.2.2	**Button [2.2.2 – Update stock]** Extract iNumSold sold from input dialog box ✓ Test if tblDevices['NumInStock'] - iNumSold > 0 ✓ tblDevices.Edit; ✓ tblDevices['InStock'] := tblDevices['InStock'] - iNumSold; ✓ tblDevices.Post; ✓ else ShowMessage "Not enough items in stock." ✓	6	
	Subtotal:	**18**	
	TOTAL SECTION B:	**40**	

ANNEXURE C

QUESTION 3: MARKING GRID – OBJECT-ORIENTED PROGRAMMING

CENTRE NUMBER:	**EXAMINATION NUMBER:**		
QUESTION	**DESCRIPTION**	**MAX. MARKS**	**LEARNER'S MARKS**
3.1.1	**Constructor Create:** Header with correct parameter values ✓ (correct order, data types) ✓ Assign `fSwitchID` , `fDevice` and `fPowerUsage` ✓ to correct parameters ✓ Assign False to `fSwitchStatus` ✓	5	
3.1.2	**Function getSwitchID** with the string return type ✓ Result := fSwitchID✓	2	
3.1.3	**Function energyUsed** with real return type Function heading with parameter ✓and correct return type✓ Result = fPowerUsage ✓ * iHours / 1000 ✓	4	
3.1.4	**Procedure setSwitchStatus** Correct heading✓ with Boolean parameter ✓ Set the fSwitchStatus attribute to the parameter value✓	3	
3.1.5	**Function toString** with string return type ✓ Correct labels and correct attribute names ✓ Call the determineSwitchStatus method✓ Correct formatting ✓	4	
	Subtotal: Object class	**18**	

QUESTION 3: MARKING GRID (CONT.)

QUESTION	DESCRIPTION	MAX. MARKS	LEARNER'S MARKS
3.2.1	**Button [3.2.1 – Instantiate object]** Extract the switchID from the combobox Extract the selected device from the listbox sLine := lstQ3_2_1.Items[lstQ3_2_1.ItemIndex] ✓ sDevice := Copy(sLine, 1, pos('#',sLine ✓) – 1)✓ Extract the power usage from sLine: iWatt := strToInt(Copy(sLine,pos('#',sLine)+1)) ✓ Instantiate the object with the provided values objSmartSwitch:= ✓TSmartSwitch.create(cmbQ3_2_1.Text,sDevice ✓,iWatt ✓) Display the object details in the rich edit using the toString method ✓	8	
3.2.2	**Button [3.2.2 – Change switch status]** Extract ItemIndex from the radio group Use a case/if statement and call the setSwitchStatus method with either True or False as argument: ✓ case rgpQ3_2_2.ItemIndex of 0: ✓ objSmartSwitch.setSwitchStatus(true✓) 1: objSmartSwitch.setSwitchStatus(false) ✓ Display the switch ID and status of the switch in the rich edit ✓	5	
3.2.3	**Button [3.2.3 – Write to file]** Link the internal file and external file (AssignFile) and use the Append statement to open the file: AssignFile(tFile, 'log.txt'); Append(tFile); ✓ Write the date, time, switch ID and status to the file by calling the relevant methods WriteIn(tFile, (Date(now) +'#'+ lblTime.Caption ✓ +'#'+ objSmartSwitch.getSwitchID()✓ +'#' + objSmartSwitch.determineSwitchStatus()✓)) Close file ✓	6	
3.2.4	**Button [3.2.4 – Power usage]** Extract hours from edtQ3_2_4and convert to an integer ✓ Call the energyUsed method with Hours as argument ✓ Display the enery used in the rich edit with correct text. ✓ redQ3.Lines.Add ('Energy used is: '+ FloatToStr (rEnergy) + ' kWh')	3	
	Subtotal Form class:	**22**	
	TOTAL SECTION C:	**40**	

ANNEXURE D

QUESTION 4: MARKING GRID – PROBLEM SOLVING

CENTRE NUMBER:	**EXAMINATION NUMBER:**		
QUESTION	**DESCRIPTION**	**MAX MARKS**	**LEARNER'S MARKS**
4.1	**Button [4.1 – Display]** Clear redQ4 ✓ Add heading and column numbers ✓ Loop from `I` to length of array ✓ (or 5) Add `I` to output string ✓ Loop `J` from 1 to length of array[`I`] ✓(or 6) Add item at array[`I`, `J`] to output string ✓ Add output string to rich edit ✓	7	
4.2	**Button [4.2 – Add access point]** Extract the row and column from the spin edits ✓ Initialise counter ✓ Loop `I` through rows of the array ✓ Check if array at index [row, `I`] = 'A' ✓ Increment counter ✓ Check if counter is 3 or less ✓ Check if character at index of the array is NOT an '`A`' ✓ Add an '`A`' to the correct index of the array ✓ If it is an '`A`' Display message that there is an access point ✓ Else Display message that there are already 3 access points ✓	10	

4.3	**Button [4.3 – Coverage]** **Determine where an access point is:** Loop through the rows of the array ✓ Loop through the columns of the array✓ Check if access point is at the current index ✓ Determine adjacent spaces **Determine the adjacent spaces:** Check the row above ✓ Check the row below ✓ Check the column left ✓ Check the column right ✓ If the index of row > 0 ✓ AND ✓ index of column > 0 ✓ If array[row, column] = '_' ✓ OR NOT 'A' Set array[row, column] = '*' ✓ **Display** Display array after signals have been added ✓	13	

	TOTAL SECTION D: **GRAND TOTAL:**	**30** **150**	

SUMMARY OF LEARNER'S MARKS:

CENTRE NUMBER:		**LEARNER'S EXAMINATION NUMBER:**			
	SECTION A	**SECTION B**	**SECTION C**	**SECTION D**	
	QUESTION 1	**QUESTION 2**	**QUESTION 3**	**QUESTION 4**	**GRAND TOTAL**
MAX. MARKS	**40**	**40**	**40**	**30**	**150**
LEARNER'S MARKS					

ANNEXURE E: SOLUTION FOR QUESTION 1

```
unit Question1_u;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, ExtCtrls, ComCtrls, pngimage;

type
  TfrmQuestion1 = class(TForm)
    grpQ1_1: TGroupBox;
    edtQ1_1: TEdit;
    btnQ1_1: TButton;
    grpQ1_2: TGroupBox;
    btnQ1_2: TButton;
    pnlQ1_2: TPanel;
    grpQ1_5: TGroupBox;
    edtQ1_5: TEdit;
    redQ1_5: TRichEdit;
    Label1: TLabel;
    btn1_5: TButton;
    grpQ1_3: TGroupBox;
    grpQ1_4: TGroupBox;
    Image1: TImage;
    Label2: TLabel;
    edtQ1_3: TEdit;
    Label3: TLabel;
    btnQ1_3: TButton;
    pnlQ1_3: TPanel;
    btnQ1_4: TButton;
    redQ1_4: TRichEdit;
    procedure btnQ1_1Click(Sender: TObject);
    procedure btnQ1_2Click(Sender: TObject);
    procedure btn1_5Click(Sender: TObject);
    procedure btnQ1_3Click(Sender: TObject);
    procedure btnQ1_4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmQuestion1: TfrmQuestion1;

implementation

{$R *.dfm}
```

```
// ============================================================
// 1.1  Formatting      4 marks
// ============================================================

procedure TfrmQuestion1.btnQ1_1Click(Sender: TObject);
begin
  edtQ1_1.Font.Size := 14;
  edtQ1_1.Text := 'Hello world';
  edtQ1_1.Font.Style := [fsUnderline];
  edtQ1_1.Color := clGreen;
end;

// ============================================================
// 1.2  Random number   8 marks
// ============================================================

procedure TfrmQuestion1.btnQ1_2Click(Sender: TObject);
var
  iRandom: integer;
begin
  iRandom := Random(99) + 1;
  if iRandom <= 9 then
    pnlQ1_2.Caption := IntToStr(iRandom) + ' is a single digit value'
  else
    pnlQ1_2.Caption := IntToStr(iRandom) + ' is a two digit value';
end;


// ============================================================
// 1.3  Area          8 marks
// ============================================================

procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);
var
  rSide, rArea: real;
const
  pi: real = 22 / 7;
begin
  rSide := StrToFloat(edtQ1_3.Text);
  rArea := (3 * sqrt(3) / 2) * sqr(rSide) -  pi * sqr(rSide / 2);
  pnlQ1_3.Caption := FloatToStrF(rArea, ffFixed, 7, 1) + ' cm squared';
end;

// ============================================================
// 1.4  Find          11 marks
// ============================================================

procedure TfrmQuestion1.btn1_4Click(Sender: TObject);
var
  tFile: textfile;
  sLine, sWord: String;
  iCount: integer;
begin
  redQ1_4.Clear;
  AssignFile(tFile, 'Words.txt');
  Reset(tFile);
```

```
  sWord := (edtQ1_4.Text);
  iCount := 0;
  while NOT EOF(tFile) do
  begin
    Readln(tFile, sLine);
    if UpperCase(sWord) = UpperCase(sLine) then
    begin
      inc(iCount);
    end;
  end;
  if iCount > 0 then
  begin
    redQ1_4.Lines.Add('Occurrences: ' + IntToStr(iCount));
  end
  else
  begin
    redQ1_4.Lines.Add('Word not found');
  end;
  CloseFile(tFile);
end;

// ============================================================
// 1.5  Booster rocket       9 marks
// ============================================================

procedure TfrmQuestion1.btnQ1_5Click(Sender: TObject);
var
  rTotalFuel, rFuel: real;
  iCounter: integer;
begin

  // Provided code
  redQ1_5.Paragraph.TabCount := 3;
  redQ1_5.Paragraph.tab[0] := 1;
  redQ1_5.Paragraph.tab[1] := 50;
  redQ1_5.Paragraph.tab[2] := 150;

  redQ1_5.Lines.Add('Second' + #9 + 'Fuel used' + #9 +'Fuel left ' );
  //1.5 Booster rocket

  rTotalFuel := StrToFloat(inputbox('Fuel', 'Total litres of fuel: ',
'550'));

  iCounter := 0;
  while rTotalFuel > 200 do
  begin
    inc(iCounter);
    rFuel := rTotalFuel / 100 * 7.5;
    rTotalFuel := rTotalFuel - rFuel;
    redQ1_5.Lines.Add(IntToStr(iCounter)+ #9+
                      FloatToStrF(rFuel,FFFixed,5,2)+#9+
                      FloatToStrF(rTotalFuel,FFFixed,5,2));
  end;
end;
end.
```

ANNEXURE F: SOLUTION FOR QUESTION 2

```
//=========================================================================
// 2.1 – Section: SQL statements
//=========================================================================

//=========================================================================
//  2.1.1   iOS devices               3 marks
//=========================================================================

     sSQL1 := 'SELECT DeviceID, DeviceName FROM tblDevices ' +
              'WHERE OperatingSystem = "iOS"';
//=========================================================================
//  2.1.2   Category selected         3 marks
//=========================================================================

     sSQL2 := 'SELECT DeviceName, Category, NumInStock ' +
              'FROM tblDevices ' +
              'WHERE Category LIKE "%' + sDeviceType + '"';
//=========================================================================
//  2.1.3  Online support            6 marks
//=========================================================================

     sSQL3 := 'SELECT DeviceName, Category, OperatingSystem ' +
              'FROM tblDevices D, tblManufacturers M ' +
              'WHERE D.ManufacturerID = M.ManufacturerID ' +
              'AND OnlineSupport = True ' +
              'ORDER BY DeviceName';
//=========================================================================
//  2.1.4  Profit per manufacturer   6 marks
//=========================================================================

     sSQL4 := 'SELECT ManufacturerID, ' +
              'FORMAT(SUM(NumInStock * (Price * 0.6)), "CURRENCY") ' +
              'AS [Profit] ' +
              'FROM tblDevices GROUP BY ManufacturerID';
// ========================================================================
//  2.1.5  Remove devices             4 marks
//=========================================================================

     sSQL5 := 'Delete * FROM tblDevices ' +
              'WHERE Category = "Smart speaker" ' +
              'AND ManufacturerID = "M104"';
```

```
//=============================================================================
// 2.2 – Section: Delphi code
//=============================================================================

//=============================================================================
// 2.2.1   Display products        12 marks
// ============================================================================
procedure TfrmQuestion2.btnQ2_2_1Click(Sender: TObject);
begin
  // Provided code
  redQ2_2_1.Clear;
  // Question 2.2.1
    tblManufacturers.First;
    while NOT tblManufacturers.Eof do
      begin
        redQ2_2_1.Lines.Add(tblManufacturers['ManufacturerName'] + ': ' +
                            tblManufacturers['ContactNumber']);
        redQ2_2_1.Lines.Add(#9 + 'Device name' + #9 + 'In stock' + #9 +
                            'Price');
        tblDevices.First;
        while NOT tblDevices.Eof do
          begin
            if (tblDevices['ManufacturerID'] =
                tblManufacturers['ManufacturerID']) then
              begin
                redQ2_2_1.Lines.Add(#9 + tblDevices['DeviceName'] + #9
                                    + IntToStr(tblDevices['NumInStock']) +
                                    #9 +
                                    FloatToStrF(tblDevices['Price'],
                                    ffCurrency, 8, 2));
            end;
          tblDevices.Next;
        end;
        redQ2_2_1.Lines.Add('');
      tblManufacturers.Next;
      end;
end;
// ============================================================================
// 2.2.2   Update stock      6 marks
// ============================================================================
procedure TfrmQuestion2.btnQ2_2_2Click(Sender: TObject);
var
  iNumSold: integer;
begin
  // Question 2.2.2
  iNumSold := StrToInt(InputBox('Products sold', 'Amount:', '50'));
    if tblDevices['NumInStock'] - iNumSold > 0 then
      begin
        tblDevices.Edit;
        tblDevices['NumInStock'] := tblDevices['NumInStock'] - iNumSold;
        tblDevices.Post;
      end
    else
        ShowMessage('Not enougth items in stock.');
end;
```

```
// =====================================================
// {$ENDREGION}
// =====================================================
// {$REGION 'Provided code: Setup DB connections - DO NOT CHANGE!'}
// =====================================================

procedure TfrmQuestion2.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
// Disconnects from database and closes all open connections
  dbCONN.dbDisconnect;
end;

procedure TfrmQuestion2.FormCreate(Sender: TObject);
begin
// Provided code
  redQ2_2_1.Paragraph.TabCount := 2;
  redQ2_2_1.Paragraph.Tab[0] := 100;
  redQ2_2_1.Paragraph.Tab[1] := 150;
  redQ2_2_1.Paragraph.Tab[2] := 200;
end;

procedure TfrmQuestion2.FormShow(Sender: TObject);
begin
// Sets up the connection to database and opens the tables.
  dbCONN := TConnection.Create;
  dbCONN.dbConnect;
  tblManufacturers := dbCONN.tblOne;
  tblProducts := dbCONN.tblMany;
  dbCONN.setupGrids(dbgManufacturers, dbgProducts, dbgrdSQL);
  pgcDBAdmin.ActivePageIndex := 0;
end;
// =====================================================
// {$ENDREGION}

end.
```

ANNEXURE F: SOLUTION FOR QUESTION 3

Object class

```
// ========================================================================
//  3.1.1   Constructor      5 marks
// ========================================================================
constructor TSmartSwitch.create(sSwitchID: String; sDevice: String;
iPowerUsage: Integer);
begin
  fSwitchID := sSwitchID;
  fDevice:=sDevice;
  fPowerUsage := iPowerUsage;
  fSwitchStatus := False;
end;
// ========================================================================
//  3.1.2  getSwitchID       2 marks
// ========================================================================
function TSmartSwitch.getSwitchID: String;
begin
  Result := fSwitchID;
end;

// ========================================================================
//  3.1.3  energyUsed         4 marks
// ========================================================================
function TSmartSwitch.energyUsed(iHours: Integer): Real;
begin
 Result := fPowerUsage * iHours / 1000;
end;

// ========================================================================
//  3.1.4  setSwitchStatus    3 marks
// ========================================================================
procedure TSmartSwitch.setSwitchStatus(bStatus: Boolean);
begin
  fSwitchStatus := bStatus;
end;

// ========================================================================
//  3.1.5  toString           4 marks
// ========================================================================
function TSmartSwitch.toString: String;
begin
   Result := 'Switch ID: ' + fSwitchID + #13 +
    'Device: ' + fDevice + #13 +
    'Power usage: ' + intToStr(fPowerUsage) + 'W' + #13 +
    'Switch status:' +  determineSwitchStatus;
end;
```

```
// ========================================================================
//                      Provided Code
// ========================================================================

function TSwitch.determineSwitchStatus: String;
var
  sStatus: String;
begin
case fSwitchStatus of
  True:sStatus := 'ON';
  False: sStatus := 'OFF';
end;
 Result := sStatus;
end;

//========================================================================
//                      End of provided Code
// ========================================================================
```

Main form unit

```
unit Question3_u;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs,SmartSwitch_u, StdCtrls, ComCtrls, Spin, ExtCtrls;

type
  TfrmQuestion3 = class(TForm)
    redQ3: TRichEdit;
    btnQ3_2_1: TButton;
    Panel1: TPanel;
    btnQ3_2_2: TButton;
    GroupBox3: TGroupBox;
    Label3: TLabel;
    GroupBox2: TGroupBox;
    rgpQ3_2_2: TRadioGroup;
    Panel2: TPanel;
    GroupBox1: TGroupBox;
    lstQ3_2_1: TListBox;
    btnQ3_2_4: TButton;
    cmbQ3_2_1: TComboBox;
    Label1: TLabel;
    edtQ3_2_4: TEdit;
    Label2: TLabel;
    GroupBox4: TGroupBox;
    btnQ3_2_3: TButton;
    lblTime: TLabel;
    procedure btnQ3_2_1Click(Sender: TObject);
    procedure btnQ3_2_2Click(Sender: TObject);
    procedure btnQ3_2_4Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmQuestion3: TfrmQuestion3;
  objSmartSwitch:TSmartSwitch;


implementation

{$R *.dfm}
```

```
// ========================================================================
//  3.2.1   Instantiate object    8 marks
// ========================================================================
procedure TfrmQuestion3.btnQ3_2_1Click(Sender: TObject);
Var
  sLine, sDevice:String;
  iWatt:Integer;

begin
  redQ3.Clear;
  sLine := lstQ3_2_1.Items[lstQ3_2_1.ItemIndex];
  sDevice := Copy(sLine, 1,  pos('#',sLine) - 1);
  iWatt := strToInt(Copy(sLine,pos('#',sLine)+1

  objSmartSwitch:=TSmartSwitch.create( cmbQ3_2_1.Text,sDevice,iWatt);
  redQ3.Lines.Add(objSmartSwitch.toString);
end;

// ========================================================================
//  3.2.2  Change switch status  5 marks
// ========================================================================
procedure TfrmQuestion3.btnQ3_2_2Click(Sender: TObject);
begin
  redQ3.Lines.Clear;
  case rgpQ3_2_2.ItemIndex of
    0: objSmartSwitch.setSwitchStatus(True);
    1: objSmartSwitch.setSwitchStatus(False);
  end;
  redQ3.Lines.Add(objSmartSwitch.getSwitchID + ': ' +
objSmartSwitch.determineSwitchStatus);
end;

// ========================================================================
//  3.2.3  Write to file        6 marks
// ========================================================================

procedure TfrmQuestion3.Button1Click(Sender: TObject);
var
  tFile : textFile;
begin
  AssignFile(tFile, 'log.txt');
  Append(tFile);

writeln(tFile,DateToStr(now)+'#'+lblTime.Caption+'#'+objSmartSwitch.getSwit
chID+'#'+ objSmartSwitch.determineSwitchStatus);
  CloseFile(tFile);
end;
```

```
// ========================================================================
// 3.2.4  Power usage        3 marks
// ========================================================================

procedure TfrmQuestion3.btnQ3_2_4Click(Sender: TObject);
var
  iHours : Integer;
  rEnergy:Real;
begin
  redQ3.Lines.Clear;
  iHours := StrToInt(edtQ3_2_4.Text);
  rEnergy := objSmartSwitch.energyUsed(iHours);
  redQ3.Lines.Add('Energy used is: '+ FloatToStr(rEnergy) +' kWh');
end;

end.
```

ANNEXURE H: SOLUTION FOR QUESTION 4

```
unit Question4_u;

interface

uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, ComCtrls,
  ExtCtrls, Buttons, Spin, pngimage;

type
  TfrmQuestion4 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    btnQ4_1: TButton;
    redQ4: TRichEdit;
    btnQ4_2: TButton;
    btnQ4_3: TButton;
    gbxQ4_3: TGroupBox;
    sedQ4_2_Row: TSpinEdit;
    sedQ4_3_Col: TSpinEdit;
    Label1: TLabel;
    Label2: TLabel;
    gbxQ4_1: TGroupBox;
    gbxQ4_2: TGroupBox;
    Image1: TImage;
    procedure btnQ4_1Click(Sender: TObject);
    procedure btnQ4_2Click(Sender: TObject);
    procedure btnQ4_3Click(Sender: TObject);
  private
    { Private declarations }

  public
    { Public declarations }
  end;

var
  Form1: TfrmQuestion4;

  arrNetwork: array [1 .. 5, 1 .. 6] of char =
    (('_', 'A', '_', '_', '_', '_'), ('_', '_', '_', '_', '_', '_'),
    ('_', '_', '_', '_', 'A', '_'), ('_', '_', '_', '_', '_', '_'),
    ('_', 'A', '_', '_', '_', '_'));

implementation

{$R *.dfm}
```

```
// ============================================================================
//  4.1 Display                7 marks
// ============================================================================
procedure TfrmQuestion4.btnQ4_1Click(Sender: TObject);
var
  I: Integer;
  J: Integer;
  sLine: String;
begin
  redQ4.Clear;
  redQ4.Lines.Add('Access points');
  sLine := '  1 2 3 4 5 6' + #13;
  for I := 1 to Length(arrNetwork) do
  begin
    sLine := sLine + intToStr(I) + ' ';
    for J := 1 to Length(arrNetwork[I]) do
    begin
      sLine := sLine + arrNetwork[I, J] + ' ';
    end;

    sLine := sLine + #13;

  end;
  redQ4.Lines.Add(sLine);
end;

// ============================================================================
//  4.2  Add access point   10 marks
// ============================================================================
procedure TfrmQuestion4.btnQ4_2Click(Sender: TObject);
var
  I, iCounter, iRow, iCol: Integer;
begin
  redQ4.Clear;
  iRow := sedQ4_2_Row.Value;
  iCol := sedQ4_3_Col.Value;

  iCounter := 0;

  for I := 1 to Length(arrNetwork[iCol]) do
  begin
    if arrNetwork[iRow, I] = 'A' then
      inc(iCounter);
  end;

  if iCounter < 3 then
  begin
    if arrNetwork[iRow, iCol] in ['*', '_'] then
    begin
      arrNetwork[iRow, iCol] := 'A';
    end
    else
    begin
      ShowMessage('Access point already on this location.');
    end;
  end
```

```
  else
  begin
    ShowMessage('There are already 3 access points in the row.');
  end;

  btnQ4_1.Click;

end;

// ============================================================================
//  4.3 Coverage                13 marks
// ============================================================================
procedure TfrmQuestion4.btnQ4_3Click(Sender: TObject);
var
  I: Integer;
  J: Integer;
  K: Integer;
  L: Integer;
begin

  for I := 1 to Length(arrNetwork) do
  begin
    for J := 1 to Length(arrNetwork[I]) do
    begin
      if arrNetwork[I, J] = 'A' then
      begin
        for K := J - 1 to J + 1 do
        begin
          for L := I - 1 to I + 1 do
          begin
            if (K > 0) AND (L > 0) then
            begin
              if arrNetwork[L, K] = '_' then
              begin
                arrNetwork[L, K] := '*';
              end;
            end;
          end;
        end;
      end;
    end;
  end;

  btnQ4_1.Click;
end;
end.
```