

SA's Leading Past Year

Exam Paper Portal



You have Downloaded, yet Another Great Resource to assist you with your Studies 😊

Thank You for Supporting SA Exam Papers

Your Leading Past Year Exam Paper Resource Portal

Visit us @ www.saexampapers.co.za





Province of the
EASTERN CAPE
EDUCATION

Iphondo leMpuma Kapa: Isebe leMfundo
Provinsie van die Oos Kaap: Departement van Onderwys
Porafensie Ya Kapa Botjhabela: Lefapha la Thuto

NATIONAL SENIOR CERTIFICATE

GRADE 12

SEPTEMBER 2024

INFORMATION TECHNOLOGY P1 MARKING GUIDELINE

MARKS: 150

This marking guideline consists of 21 pages.



NAME OF LEARNER:

TOTAL QUESTION 1	TOTAL QUESTION 2	TOTAL QUESTION 3	TOTAL QUESTION 4	TOTAL
/35	/45	/40	/30	/150

QUESTION 1		MAX MARK	MARKS ACHIEVED
1.1	BUTTON: [Q1.1 – Panel Properties] Set the font type to Arial. ✓ Set the font size to 16. ✓ Set the font to Bold ✓ and Italic. ✓ Set the text to display on the right-hand side of the panel. ✓ Change the caption to 'Crypto Exchange'. ✓	6	
1.2	BUTTON: [Q1.2 – Share Coins] Get input from edit box ✓ Calculate shares using the total of 12 (3+4+5) ✓ Calculation for each person (share/12*input from edit box) ✓ Use DIV/trunc/floor to discard decimals ✓ Calculate Remaining Coins ✓ Display in the label Tom's Share ✓ Jerry's Share ✓ Andile's Share ✓ Remaining Coins ✓ Correct use of enter space and display of apostrophes after each person's name ✓	10	
1.3	BUTTON: [Q1.3 – Password] Initialise Password String ✓ Loop for each letter in "CRYPTO" (6 times) ✓ Initialise Number variable to 0 ✓ Conditional Loop with correct condition ✓ Increment Number variable ✓ Randomise a number ✓ in correct range ✓ Get the corresponding letter for the randomised number ✓ Build password string: ✓ include the number ✓ and the lowercase ✓ letter of the character. Display the Password in the label. ✓	12	

1.4	BUTTON: [Q1.4 – ASCII Art] Get the length of the longest line and assign to iLines ✓ Initialise an empty string sLine ✓ // Constructing the pattern by adding ‘&’ characters Loop ✓ Add ‘&’ character to sLine ✓ Display sLine in rich edit ✓ // Removing ‘&’ characters from the beginning of each line Loop Remove the first character from sLine/Build a string ✓ Display sLine in rich edit ✓	7	
	TOTAL QUESTION 1	35	

QUESTION 2		MAX. MARK	MARKS ACHIEVED
2.1.1	Button [Q2.1.1] 'SELECT * FROM tblMembers ORDER BY DateOfBirth DESC' SELECT * (all fields) ✓ FROM correct table ✓ ORDER BY correct field DESC ✓	3	
2.1.2	Button [Q2.1.2] 'SELECT CryptoType, Sum(Amount) AS [Amount] FROM tblCrypto GROUP BY CryptoType' SELECT CryptoType ✓, Sum(correct field) ✓ FROM correct table ✓ GROUP BY correct field ✓	4	
2.1.3	Button [Q2.1.3] 'SELECT count(DateOfBirth) AS [Birthdays in ' + arrMonths[StrToInt(sLine)] + '] FROM tblMembers WHERE Month(DateOfBirth) = ' + sLine SELECT count(correct field) ✓ AS [Birthdays in correct month ✓] ✓ FROM correct table ✓ WHERE Month(correct field) ✓ = input variable ✓	6	
2.1.4	Button [Q2.1.4] 'SELECT Surname, Firstname, CryptoType, Amount, format(Amount * ' + FloatToStr(rLitecoin) + ', "Currency") AS [Value] FROM tblMembers, tblCrypto WHERE tblMembers.MemID = tblCrypto.MemID AND CryptoType = "Litecoin" AND (Amount * ' + FloatToStr(rLitecoin) + ') > 5500' SELECT Surname, Firstname, CryptoType, Amount ✓, Format (Amount * LiteCoin Value ✓, "Currency" ✓) AS [Value] ✓, FROM both tables ✓ (tblMembers, tblCrypto) WHERE link between tables ✓ (tblMembers.MemID = tblCrypto.MemID) AND CryptoType = "Litecoin" ✓ AND (Amount * Litecoin value) ✓ > 5500' ✓	9	
2.1.5	Button [Q2.1.5] 'DELETE FROM tblCrypto WHERE CryptoType = "Ripple" ' DELETE FROM correct table ✓ WHERE correct field ✓ = "Ripple" ✓	3	
2.1.6	Button [Q2.1.6] 'UPDATE tblCrypto SET Amount = Amount / 2 WHERE CryptoType = "Bitcoin" ' UPDATE correct table ✓ SET Amount = Amount / 2 ✓ WHERE CryptoType = "Bitcoin" ✓	3	

2.2.1	Button [Q2.2.1] Test if valid condition: correct Surname ✓ = data field value ✓ Display the correct fields (Name; Surname; E-Mail; Gender) ✓ in the correct format ✓ Move to next record in table ✓	5	
2.2.2	Button [Q2.2.2] Instantiate ✓ and Initialise ✓ for Counting and Gender Types (standard variables or arrays) Test if valid condition: correct Gender = data field value ✓ Increment appropriate Counter ✓ Increment Total ✓ Move to next record in table ✓ Test if valid condition: Total = record count of Members table ✓ String Variable ← Correct ✓ else String Variable ← Incorrect ✓ Display the correct fields (Male, Female, Genderfluid, Non-binary) in the correct format ✓✓ Display String variable correct or incorrect ✓	12	
	TOTAL QUESTION 2	45	

QUESTION 3		MAX. MARK	MARKS ACHIEVED
3.1.1	Constructor Create Constructor definition with three parameters ✓ of correct data type ✓ Assign three parameters to attributes ✓✓ Set remaining attributes to zero ✓	5	
3.1.2	Accessor Method – getCryptoName Function definition with String return type ✓ If/Case statement ✓ Correct conversion of attribute fCrypto to cryptocurrency name ✓ Return the cryptocurrency name ✓	4	
3.1.3	Mutator Method – setAmount Procedure definition ✓ with Real parameter value ✓ Determine the exchange rate based on rules (any method): Bitcoin = 0.000001 1 / 1000000 (given) Ethereum = 0.00002 ✓ 1 / 50000 Litecoin = 0.0004 ✓ 1 / 2500 Assign fAmount to the received parameter * exchange rate ✓ Assign fOriginalValue to fOriginalValue ✓ + received parameter ✓	7	
3.1.4	Mutator Method – setValue Procedure definition ✓ Determine the exchange rate based on rules (any method): Bitcoin = Random: Range 400 000 – 1 600 000 (both inclusive) ✓ Ethereum = Random: Range 25 000 – 75 000 (both inclusive) ✓ Litecoin = Random: Range 500 – 3000 (both inclusive) ✓ Assign fCurrentValue to fAmount * random exchange rate ✓	5	
3.1.5	Auxiliary Method – calcProfitLoss Function definition with String return type ✓ Test if (fCurrentValue > fOriginalValue) ✓ Return "Profit" ✓ Else Test if (fCurrentValue < fOriginalValue) ✓ Return "Loss" ✓ Else Return "Even" ✓	6	
Subtotal: Object class		[27]	

3.2.1	Button [3.2.1 - Instantiate Crypto Object] Retrieve input from components ✓ Instantiate object: objCrypto := ✓ TCrypto.Create ✓ parameters of correct data type ✓ and order ✓ Display message indicating object was instantiated ✓	6	
3.2.2	Button [3.2.2 – Purchase Crypto] Extract the price from the edit box ✓ Use setAmount method with price argument ✓ to set attribute	2	
3.2.3	Timer [tmrLive] Use setValue method ✓ to set attribute Display object in rich edit ✓ using toString method ✓ Display Profit/Loss in panel ✓ using calcProfitLoss method ✓	5	
	Subtotal: Main unit	[13]	
	TOTAL QUESTION 3	40	

QUESTION 4		MAX. MARK	MARKS ACHIEVED
4.1	Button [4.1 – Read contents] Assign the text file ✓ Test if file exists ✓ If not, show a message ✓ Reset the text file ✓ Initialise counter to 0 ✓ While loop through the text file ✓ Increment the counter ✓ Read a value from text file into string variable ✓ Loop columns from 1 to 3 ✓ Find position of comma in string variable ✓ Assign string variable to correct 2D Array[Counter,Column] ✓ Delete out of string variable up to and including the comma Assign remaining string variable to 2D Array[Counter,Column 4] ✓	12	
4.2	Button [4.2 – Display] Loop Row from 1 to Counter ✓ Loop Col from 1 to 4 ✓ String variable assigned string variable + 2D Array[Row,Col] ✓ + Tab space ✓ Display in the richedit ✓	5	
4.3	Button [4.3 – Sort] Loop Outer from 1 to Counter - 1 ✓ Loop Inner from Outer + 1 to Counter ✓ Test if 2D Array[Inner,4] ✓ is less than 2D Array[Outer,4] ✓ <i>must convert string column 4 to a number</i> ✓ Loop Col 1 to 4 ✓ String Temp ← 2D Array[Outer,Col] ✓ 2D Array[Outer,Col] ← 2D Array[Inner,Col] ✓ 2D Array[Inner,Col] ← String Temp ✓ Display sorted 2D Array in rich edit ✓	10	
4.4	Button [4.4 – Total market cap] Variable Total must be either Real or Int64 Initialise Total to 0 Loop Rows from 1 to Counter ✓ Total assigned Total + (2D Array[Row,Col 4] * 18) ✓ Display Total in rich edit in correct format ✓	3	
TOTAL QUESTION 4		30	

SAMPLE SOLUTIONS**QUESTION 1**

////////// 35 marks //////////

```
// =====
//                               Question 1.1 – 6 marks
// =====
procedure TfrmQuestion1.btn1_1Click(Sender: TObject);
begin
    /// Enter your code below ///

    with pnlOutput do
        begin
            Font.Name := 'Arial';
            Font.Size := 16;
            Font.Style := [fsBold, fsItalic];
            Alignment := taRightJustify;
            Caption := 'Crypto Exchange';
        end;
    end;
```

```
// =====
//                               Question 1.2 – 10 Marks
// =====
procedure TfrmQuestion1.btn1_2Click(Sender: TObject);
var
    iNumCoins : Integer;
    iTom, iJerry, iAndile : Integer;
    iLeftOver : Integer;
begin
    /// Enter your code below ///

    iNumCoins := StrToInt(edtInput.Text);
    iTom      := iNumCoins * 3 DIV 12;
    iJerry    := iNumCoins * 4 DIV 12;
    iAndile   := iNumCoins * 5 DIV 12;
    iLeftOver := iNumCoins – iTom – iJerry - iAndile;

    lbl1_2.Caption := 'Tom"s Share: ' + IntToStr(iTom) + #13 +
        'Jerry"s Share: ' + IntToStr(iJerry) + #13 +
        'Andile"s Share: ' + IntToStr(iAndile) + #13 +
        'Remaining Coins: ' + IntToStr(iLeftOver);
end;
```

```
// =====
//                                     Question 1.3 – 12 marks
// =====
procedure TfrmQuestion1.btn1_3Click(Sender: TObject);
// Provided code - DO NOT DELETE OR ALTER //
CONST
    PASSWORD = 'CRYPTO';
var
    sPassword : String;
    sChar : Char;
    iNum : Integer;
begin
    /// Enter your code below ///

    sPassword := '';
    for var I := 1 to Length(PASSWORD) do
        begin
            iNum := 0;
            repeat
                inc(iNum);
                sChar := Chr(Random(26) + 65);
            until sChar = PASSWORD[I];
            sPassword := sPassword + IntToStr(iNum) + Lowercase(sChar);
        end;
    lbl1_3.Caption := sPassword;
end;
```

```
// =====
//                                     Question 1.4 – 7 Marks
// =====
procedure TfrmQuestion1.btn1_4Click(Sender: TObject);
var
    sLine : String;
    iLines : Integer;
begin
    /// Enter your code below ///

    redOutput.Clear;
    sLine := '';
    iLines := StrToInt(InputBox('Lines','Enter the number of lines.',''));
    for var I := 1 to iLines do
        begin
            sLine := sLine + '&';
            redOutput.Lines.Add(sLine);
        end;
    for var I := 1 to iLines - 1 do
        begin
            delete(sLine,1,1);
            redOutput.Lines.Add(sLine);
        end;
    end;
```

QUESTION 2

////////// 45 marks //////////

// =====

// Question 2.1.1 – 3 marks

// =====

procedure TfrmQuestion2.btn2_1_1Click(Sender: TObject);

// Provided code - DO NOT DELETE OR ALTER //

var

sSQL1: String;

begin

/// Enter your code below ///

sSQL1 := 'SELECT * ' +

'FROM tblMembers ' +

'ORDER BY DateOfBirth ASC';

// Provided code - DO NOT DELETE OR ALTER //

dbCONN.runSQL(sSQL1);

if length(sSQL1) <> 0 then

SetGridColumnWidths(dbgSQL);

end;

// =====

// Question 2.1.2 – 4 Marks

// =====

procedure TfrmQuestion2.btn2_1_2Click(Sender: TObject);

// Provided code - DO NOT DELETE OR ALTER //

var

sSQL2: String;

begin

/// Enter your code below ///

sSQL2 := 'SELECT CryptoType, Sum(Amount) AS [Amount] ' +

'FROM tblCrypto ' +

'GROUP BY CryptoType';

// Provided code - DO NOT DELETE OR ALTER //

dbCONN.runSQL(sSQL2);

if length(sSQL2) <> 0 then

SetGridColumnWidths(dbgSQL);

end;

```
// =====
//                                     Question 2.1.3 – 6 marks
// =====
procedure TfrmQuestion2.btn2_1_3Click(Sender: TObject);
  // Provided code - DO NOT DELETE OR ALTER //
  CONST
    ARRMONTHS : array[1..12] of String = ('January','February','March','April',
                                           'May','June','July','August','September',
                                           'October','November','December');

  var
    sSQL3 : String;
    sLine : String;
  begin
    sLine := inputbox('Month','Enter your month (1-12)','1');
    /// Enter your code below ///
    //Alternate solution
    //Case statement or If statement to determine month in string format

    sSQL3 := 'SELECT count(*) AS [Birthdays in ' + ARRMONTHS[StrToInt(sLine)] + ']' +
             'FROM tblMembers ' +
             'WHERE Month(DateOfBirth) = ' + sLine;

    // Provided code - DO NOT DELETE OR ALTER //
    dbCONN.runSQL(sSQL3);
    if length(sSQL3) <> 0 then
      SetGridColumnWidths(dbgSQL);
  end;
```

```
// =====
//                                     Question 2.1.4 – 9 Marks
// =====
procedure TfrmQuestion2.btn2_1_4Click(Sender: TObject);
  // Provided code - DO NOT DELETE OR ALTER //
  var
    sSQL4: String;
    rLiteCoin : Real;
  begin
    rLitecoin := 1339.30;
    /// Enter your code below ///

    sSQL4 := 'SELECT Surname, Firstname, CryptoType, Amount, ' +
             'format(Amount * ' + FloatToStr(rLitecoin) + ', "Currency") AS [Value] ' +
             'FROM tblMembers, tblCrypto ' +
             'WHERE tblMembers.MemID = tblCrypto.MemID ' +
             'AND CryptoType = "Litecoin" ' +
             'AND (Amount * ' + FloatToStr(rLitecoin) + ') > 5500';

    // Provided code - DO NOT DELETE OR ALTER //
    dbCONN.runSQL(sSQL4);
    if length(sSQL4) <> 0 then
      SetGridColumnWidths(dbgSQL);
  end;
```

```
// =====
//                                     Question 2.1.5 – 3 marks
// =====
procedure TfrmQuestion2.btn2_1_5Click(Sender: TObject);
    // Provided code - DO NOT DELETE OR ALTER //
var
    sSQL5: String;
begin
    /// Enter your code below ///

    sSQL5 := 'DELETE FROM tblCrypto ' +
              'WHERE CryptoType = "Ripple" ';

    // Provided code - DO NOT DELETE OR ALTER //
    dbCONN.executeSQL(sSQL5,dbgMembers,dbgCrypto,dbgSQL);
    if length(sSQL5) <> 0 then
        SetGridColumnWidths(dbgSQL);
end;
```

```
// =====
//                                     Question 2.1.6 – 3 Marks
// =====
procedure TfrmQuestion2.btn2_1_6Click(Sender: TObject);
    // Provided code - DO NOT DELETE OR ALTER //
var
    sSQL6: String;
begin
    /// Enter your code below ///

    sSQL6 := 'UPDATE tblCrypto ' +
              'SET Amount = Amount / 2 ' +
              'WHERE CryptoType = "Bitcoin"';

    // Provided code - DO NOT DELETE OR ALTER //
    dbCONN.executeSQL(sSQL6,dbgMembers,dbgCrypto,dbgSQL);
    if length(sSQL6) <> 0 then
        SetGridColumnWidths(dbgSQL);
end;
```

```
// =====
//                               Question 2.2.1 – 5 marks
// =====
procedure TfrmQuestion2.btn2_2_1Click(Sender: TObject);
// Provided code - DO NOT DELETE OR ALTER //
var
    sSurname : String;
    sString : String;
begin
    with redOutput do
        begin
            Clear;
            SelAttributes.Style := [fsBold];
            Lines.Add('Member"s details');
        end;
    with tblMembers do
        begin
            Open;
            First;
            sSurname := edtSurname.Text;
            sString := sSurname + ' was not found in database';
            while not (eof) do
                begin
                    /// Enter your code below ///

                    if UpperCase(FieldByName('Surname').AsString) = UpperCase(sSurname) then
                        begin
                            sString := 'Name: ' + FieldByName('FirstName').AsString + #13 +
                                'Surname: ' + FieldByName('Surname').AsString + #13 +
                                'E-Mail: ' + FieldByName('E-mail').AsString + #13 +
                                'Gender: ' + FieldByName('Gender').AsString;
                        end;
                    Next;
                end;
            redOutput.Lines.Add(sString);
        end;
    end;
end;
```

```
// =====
//                                     Question 2.2.2 – 12 Marks
// =====
procedure TfrmQuestion2.btn2_2_2Click(Sender: TObject);
var
  sGender, sLine : String;
  arrGender : Array[1..4] of String;
  arrCryptoCount : Array[1..4] of Integer;
  iTotal, K : Integer;
begin
  // Provided code - DO NOT DELETE OR ALTER //
  with redOutput do
    begin
      Clear;
      SelAttributes.Style := [fsBold];
      Lines.Add('Total members = ' + IntToStr(tblMembers.RecordCount));
      Lines.Add('-----');
    end;
  with tblMembers do
    begin
      Open;
      First;
      /// Enter your code below ///

      arrGender[1] := 'Male';
      arrGender[2] := 'Female';
      arrGender[3] := 'Genderfluid';
      arrGender[4] := 'Non-binary';
      iTotal := 0;
      for K := 1 to 4 do
        arrCryptoCount[K] := 0;

      while not (eof) do
        begin
          for K := 1 to 4 do
            begin
              if FieldByName('Gender').AsString = arrGender[K] then
                begin
                  inc(arrCryptoCount[K]);
                  inc(iTotal);
                end;
            end;
          Next;
        end;
      if iTotal = RecordCount then
        sLine := 'Correct'
      else
        sLine := 'Incorrect';
      for K := 1 to 4 do
        redOutput.Lines.Add(arrGender[K] + #9 + IntToStr(arrCryptoCount[K]));
      redOutput.Lines.Add(#13 + sLine);
    end;
  end;
end;
```

QUESTION 3

////////// 40 marks //////////

// =====

// Question 3.1.1 – 5 marks

// =====

constructor TCrypto.create(sFirstName, sSurname: String; iCrypto: Integer);

begin

fFirstName := sFirstName;

fSurname := sSurname;

fCrypto := iCrypto;

fAmount := 0;

fOriginalValue := 0;

fCurrentValue := 0;

end;

// =====

// Question 3.1.2 – 4 Marks

// =====

function TCrypto.getCryptoName: String;

begin

case fCrypto of

0 : Result := 'Bitcoin';

1 : Result := 'Ethereum';

2 : Result := 'Litecoin';

end;

end;

// =====

// Question 3.1.3 – 7 Marks

// =====

procedure TCrypto.setAmount(rMoney : Real);

var

rConversion : Real;

begin

rConversion := 0;

case fCrypto of

0 : rConversion := 1 / 1000000;

1 : rConversion := 1 / 50000;

2 : rConversion := 1 / 2500;

end;

fAmount := rMoney * rConversion;

fOriginalValue := fOriginalValue + rMoney;

end;

```
// =====  
//                                     Question 3.1.4 – 5 marks  
// =====  
procedure TCrypto.setValue;  
begin  
    case fCrypto of  
        0 : fCurrentValue := fAmount * randomRange(400000,1600001);  
        1 : fCurrentValue := fAmount * randomRange(25000,75001);  
        2 : fCurrentValue := fAmount * randomRange(500,3001);  
    end;  
end;
```

```
// =====  
//                                     Question 3.1.5 – 6 Marks  
// =====  
function TCrypto.calcProfitLoss: String;  
begin  
    if fCurrentValue > fOriginalValue then  
        Result := 'Profit'  
    else  
        if fCurrentValue < fOriginalValue then  
            Result := 'Loss'  
        else  
            Result := 'Even';  
        end;  
    end;  
end;
```

```
// =====
//                                     Question 3.2.1 – 6 marks
// =====
procedure TfrmQuestion3.btn3_2_1Click(Sender: TObject);
begin
    /// Enter your code below ///
    objCrypto := TCrypto.Create(edtName.Text, edtSurname.Text, rgpCrypto.ItemIndex);
    ShowMessage('Crypto account created successfully');

    // Provided code - DO NOT DELETE OR ALTER //
    btn3_2_2.Enabled := True;
end;
```

```
// =====
//                                     Question 3.2.2 – 2 Marks
// =====
procedure TfrmQuestion3.btn3_2_2Click(Sender: TObject);
begin
    /// Enter your code below ///
    objCrypto.setAmount(StrToFloat(edtMoney.Text));

    // Provided code - DO NOT DELETE OR ALTER //
    btnLive.Enabled := True;
end;
```

```
// =====
//                                     Question 3.2.3 – 5 Marks
// =====
procedure TfrmQuestion3.tmrLiveTimer(Sender: TObject);
begin
    /// Enter your code below ///
    objCrypto.setValue;
    redOutput.Lines.Add(objCrypto.toString);
    pnlOutput.Caption := objCrypto.calcProfitLoss;
end;
```

QUESTION 4

////////// 30 marks //////////

```
// =====
//                               Question 4.1 – 12 marks
// =====
```

```
procedure TfrmQuestion4.btn4_1Click(Sender: TObject);
var
    MyFile : TextFile;
    sLine  : String;
    iPos   : Integer;
    iCol   : Integer;
begin
    /// Enter your code below ///
    AssignFile(MyFile, 'Crypto.txt');
    try
        Reset(MyFile);
    except
        ShowMessage('File not found');
        Exit;
    end;

    iCount := 0;
    while not eof(MyFile) do
        begin
            inc(iCount);
            ReadLn(MyFile, sLine);
            for iCol := 1 to 3 do
                begin
                    iPos := pos(',', sLine);
                    ar2Crypto[iCount, iCol] := copy(sLine, 1, iPos - 1);
                    delete(sLine, 1, iPos);
                end;
            ar2Crypto[iCount, 4] := sLine;
        end;

        CloseFile(MyFile);
    end;
```

```
// =====
//                                     Question 4.2 – 5 marks
// =====
procedure TfrmQuestion4.btn4_2Click(Sender: TObject);
var
    iRow, iCol : Integer;
    sLine : String;
begin
    // Provided code - DO NOT DELETE OR ALTER //
    with redOutput do
        begin
            Clear;
            Paragraph.TabCount := 3;
            Paragraph.Tab[0] := 60;
            Paragraph.Tab[1] := 120;
            Paragraph.Tab[2] := 200;
            SelAttributes.Style := [fsBold];
            Lines.Add('Name' + #9 + 'Symbol' + #9 + 'Price (ZAR)' + #9 + 'Market Cap (USD)');
        end;
        /// Enter your code below ///
        for iRow := 1 to iCount do
            begin
                sLine := "";
                for iCol := 1 to 4 do
                    sLine := sLine + (ar2Crypto[iRow,iCol] + #9);
                redOutput.Lines.Add(sLine);
            end;
        end;
    end;

// =====
//                                     Question 4.3 – 10 Marks
// =====
procedure TfrmQuestion4.btn4_3Click(Sender: TObject);
var
    K, L, J : Integer;
    sTemp : String;
begin
    /// Enter your code below ///
    for K := 1 to iCount - 1 do
        for L := K + 1 to iCount do
            begin
                if StrToFloat(ar2Crypto[K,4]) < StrToFloat(ar2Crypto[L,4]) then
                    begin
                        for Col := 1 to 4 do
                            begin
                                sTemp := ar2Crypto[K,Col];
                                ar2Crypto[K,Col] := ar2Crypto[L,Col];
                                ar2Crypto[L,Col] := sTemp;
                            end;
                        end;
                    end;
            end;
        end;
    btn4_2.Click;
end;
```

```
// =====  
//                               Question 4.4 – 3 marks  
// =====  
procedure TfrmQuestion4.btn4_4Click(Sender: TObject);  
var  
    rTotalCap : Real;  
    iRow : Integer;  
begin  
    /// Enter your code below ///  
    rTotalCap := 0;  
    for iRow := 1 to iCount do  
        rTotalCap := rTotalCap + StrToFloat(ar2Crypto[iRow,4]);  
        redOutput.Lines.Add(#13 + 'Total Market Cap: ' +  
            (FloatToStrF(rTotalCap,ffCurrency,25,2) * 18));  
end;
```