

```

{OR
if (iNum1 = iNum2) then
    edtQ1_2_1.Text := 'Equal'
else
    edtQ1_2_1.Text := IntToStr(Max(iNum1, iNum2));
end;
//=====
// Question 1.2.2 (5 marks)
//=====
procedure TFormQuestion1.btnQ1_2_2Click(Sender: TObject);
// Provided code
var
    sWord1, sWord2: String;
    sTempWord: String;
begin
    sWord1 := edtWord1.Text;
    sWord2 := edtWord2.Text;
    sTempWord := sWord1;
    sWord1 := sWord2;
    sWord2 := sTempWord;
    edtWord1.Text := sWord1;
    edtWord2.Text := sWord2;
end;
//=====
// Question 1.3.1 (5 Marks)
//=====
procedure TFormQuestion1.cmbNumCakesChange(Sender: TObject);
// Provided code
const
    PRICE = 159.50;
var
    rCost: Real;
begin
    iNumCakes := cmbNumCakes.ItemIndex + 1;
    imgCakePic.Picture.LoadFromFile('Pict' + IntToStr(iNumCakes) +
        '.PNG');
    rCost := iNumCakes * PRICE;
    edtCost.Text := FloatToStrF(rCost, ffCurrency, 6, 2);
end;
//=====
// Question 1.3.2 (5 marks)
//=====
procedure TFormQuestion1.btnQ1_3Click(Sender: TObject);
// Provided code
const
    SUGAR = 375;
var
    iSugarGrams, iSugarPacks: integer;
begin
    iSugarGrams := iNumCakes * SUGAR;
    edtSugarInGrams.Text := IntToStr(iSugarGrams);
    iSugarPacks := Ceil(iSugarGrams / 1000);
    edtSugarPacks.Text := IntToStr(iSugarPacks);
end;

```

Round up:

(5)

Please turn over

```

procedure btnQ1_2_2Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure btnQ1_3Click(Sender: TObject);
procedure btnQ1_5_2Click(Sender: TObject);
procedure btnQ1_5_1Click(Sender: TObject);
procedure cmbNumCakesChange(Sender: TObject);
procedure btnQ1_2_1Click(Sender: TObject);
procedure rgpQ1_4_1Click(Sender: TObject);
procedure btnQ1_4_2Click(Sender: TObject);
procedure btnQ1_4_3Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    frmQuestion1: TFormQuestion1;
    iNumCakes: integer;
    sPassword: String;
implementation
($R *.dfm)
//=====
// Question 1.1 (3 marks)
//=====
procedure TFormQuestion1.FormCreate(Sender: TObject);
begin
    btnQ1_4_Ride:
    btnQ1_4_3.Enabled := false;
    pnlQ1_1.Color := clLime;
    pnlQ1_1.Font.Size := 15;
    pnlQ1_1.Caption := 'IT is FUN!';
end;
//=====
// Question 1.2.1 (4 marks)
//=====
procedure TFormQuestion1.btnQ1_2_1Click(Sender: TObject);
var
    iNum1, iNum2: integer;
begin
    iNum1 := StrToInt(edtNum1.Text);
    iNum2 := StrToInt(edtNum2.Text);
    if (iNum1 > iNum2) then
        edtQ1_2_1.Text := IntToStr(iNum1)
    else
        if (iNum2 > iNum1) then
            edtQ1_2_1.Text := IntToStr(iNum2)
        else
            edtQ1_2_1.Text := 'Equal';
    end;
end;

```

Any colour / number.

(3)

(4)

Please turn over

Copyright reserved

cmbNumCakes.Text
cmbNumCakes.Items[3]

```

procedure btnQ1_2_2Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure btnQ1_3Click(Sender: TObject);
procedure btnQ1_5_2Click(Sender: TObject);
procedure btnQ1_5_1Click(Sender: TObject);
procedure cmbNumCakesChange(Sender: TObject);
procedure btnQ1_2_1Click(Sender: TObject);
procedure rgpQ1_4_1Click(Sender: TObject);
procedure btnQ1_4_2Click(Sender: TObject);
procedure btnQ1_4_3Click(Sender: TObject);

private
{ Private declarations }
public
{ Public declarations }
end;

var
frmQuestion1: TfrmQuestion1;
iNumCakes: integer;
sPassword: String;

implementation

{$R *.dfm}

//=====
// Question 1.1      (3 marks)
//=====
procedure TfrmQuestion1.FormCreate(Sender: TObject);
begin
  pnlQ1_4.Hide;
  btnQ1_4_3.Enabled := false;

  pnlQ1_1.Color := clLime; ✓
  pnlQ1_1.Font.Size := 15; ✓
  pnlQ1_1.Caption := 'IT is FUN!';
end;

//=====
// Question 1.2.1    (4 marks)
//=====
procedure TfrmQuestion1.btnQ1_2_1Click(Sender: TObject);
var
  iNum1, iNum2: integer;
begin
  iNum1 := StrToInt(edtNum1.Text); } ✓
  iNum2 := StrToInt(edtNum2.Text); } ✓
  if (iNum1 > iNum2) then
    edtQ1_2_1.Text := IntToStr(iNum1) ✓
  else
    if (iNum2 > iNum1) then
      edtQ1_2_1.Text := IntToStr(iNum2) ✓
    else
      edtQ1_2_1.Text := 'Equal'; ✓

```

```

{OR
if (iNum1 = iNum2) then
  edtQ1_2_1.Text := 'Equal'
else
  edtQ1_2_1.Text := intToStr(Max(iNum1, iNum2));
end;

```

```

//=====
// Question 1.2.2      (5 marks)
//=====
procedure TfrmQuestion1.btnQ1_2_2Click(Sender: TObject);
// Provided code
var
  sWord1, sWord2: String;
  sTempWord: String;
begin
  sWord1 := edtWord1.Text; } ✓
  sWord2 := edtWord2.Text; } ✓
  sTempWord := sWord1; ✓
  sWord1 := sWord2; ✓
  sWord2 := sTempWord; ✓

  edtWord1.Text := sWord1; } ✓
  edtWord2.Text := sWord2; } ✓
end;

```

(5)

```

//=====
// Question 1.3.1      (5 Marks)
//=====
procedure TfrmQuestion1.cmbNumCakesChange(Sender: TObject);
// Provided code
const
  PRICE = 159.50;
var
  rCost: Real;
begin
  iNumCakes := cmbNumCakes.ItemIndex + 1; ✓
  imgCakePic.Picture.LoadFromFile('Pict' + IntToStr(iNumCakes) + ✓
  '.PNG');
  rCost := iNumCakes * PRICE; ✓
  edtCost.Text := FloatToStrF(rCost, ffCurrency, 6, 2); ✓
end;

```

*cmbNumCakes.Text
cmbNumCakes.Items[3];*

(5)

```

//=====
// Question 1.3.2      (5 marks)
//=====
procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);
// Provided code
const
  SUGAR = 375;
var
  iSugarGrams, iSugarPacks: integer;
begin
  iSugarGrams := iNumCakes * SUGAR; ✓
  edtSugarInGrams.Text := IntToStr(iSugarGrams); ✓
  iSugarPacks := Ceil(iSugarGrams / 1000); ✓
  edtSugarPacks.Text := IntToStr(iSugarPacks); ✓
end;
Copyright reserved

```

Round Up:

(5)

Please turn over

```

=====
// Question 1.5.1      (6 marks)
//=====
procedure TfrmQuestion1.btnQ1_5_1Click(Sender: TObject);
var
  iNum: integer;
  rSquareRoot: Real;
begin
  redQ1_5_1.Clear;
  iNum := StrToInt(TextBox('Perfect Square', 'Enter number', '')); float
  rSquareRoot := Sqrt(iNum);
  if rSquareRoot = trunc(rSquareRoot) then
    redQ1_5_1.Lines.Add(IntToStr(iNum) + ' is a perfect square.')
  else
    redQ1_5_1.Lines.Add(IntToStr(iNum) + ' is not a perfect square.');
```

(6)

```

=====
// Question 1.5.2      (7 marks)
//=====
procedure TfrmQuestion1.btnQ1_5_2Click(Sender: TObject);
// Provided code
const
  MULTIPLIER = 3;
var
  iSum, iNum: integer;
  sOutput: String;
begin
  redQ1_5_2.Clear;
  sOutput := '';
  iSum := 0;
  iNum := 1;
  repeat
    sOutput := sOutput + IntToStr(iNum) + ' ';
    iSum := iSum + iNum;
    iNum := iNum * MULTIPLIER;
  until iSum > 1000;
  redQ1_5_2.Lines.Add(sOutput);
```

(7)

```

end.

```



```
//=====
// Question 1.4.1      (4 marks)
//=====
procedure TfrmQuestion1.rgpQ1_4_1Click(Sender: TObject);
begin
  if (rgpQ1_4_1.ItemIndex = 0) OR (rgpQ1_4_1.ItemIndex = 2) then
    pnlQ1_4.Show;
  else
    pnlQ1_4.Hide;
end;
```

(4)

```
//=====
// Question 1.4.2      (11 marks)
//=====
procedure TfrmQuestion1.btnQ1_4_2Click(Sender: TObject);
var
  i, iCountChar: Integer;
  bValid: Boolean;
begin
  bValid := false;
  iCountChar := 0;
  sPassword := edtPassword.Text;
  if length(sPassword) >= 6 then
    begin
      if sPassword[1] in ['A' .. 'Z'] then
        for i := 2 to length(sPassword) do
          if sPassword[i] in ['$', '@', '#', '&'] then
            Inc(iCountChar);
          if iCountChar >= 2 then
            begin
              ShowMessage('Valid Password');
              btnQ1_4_3.Enabled := true;
              bValid := true;
            end;
        end;
      if (bValid = false) then
        begin
          ShowMessage('Invalid Password');
          edtPassword.Text := '';
        end;
    end;
end;
```



(11)

```
//=====
// Question 1.4.3      (5 marks)
//=====
procedure TfrmQuestion1.btnQ1_4_3Click(Sender: TObject);
begin
  if sPassword[1] = 'Z' then
    sPassword[1] := 'A';
  else
    sPassword[1] := char(ord(sPassword[1])+1);
  edtPassword.Text := sPassword;
end;
```

(5)