



# education

---

Department:  
Education  
**REPUBLIC OF SOUTH AFRICA**

**NATIONAL  
SENIOR CERTIFICATE**

**GRADE 12**

**INFORMATION TECHNOLOGY P1 (Practical)**

**FREE STATE**

**COMMON PAPER JUNE 2009 : MEMORANDUM**

**MARKS: 120**

**TIME: 3 hours**

**This question paper consists of 7 pages.**

Name: \_\_\_\_\_

**Question 1**

1.1 Components (2=All components present; 1=Some components; 0=None)	2	
1.2 MainMenu (2=All items correct; 1=Items present with errors; 0=None)	2	
1.3 RadioGroup (2=Correct name/contents; 1=RadioGroup with errors; 0=None)	2	
1.4 ADOQuery, DataSource and DBGrid (One mark per component)	3	
1.5 Program code closes program (Application.Terminate/Form1.Close)	1	
1.6.1 Show All (2=Query complete; 1=Query incomplete; 0=None)	2	
1.6.2 Get Difference (3=Update/Set/Select – 1 mark each)	3	
1.6.3 Zero Difference (2=Query complete; 1=Query incomplete; 0=None)	2	
1.6.4 Win Point (3=Calculation/As/Order BY – 1 mark each)	3	
1.6.5 Filter: Team with wins (2=Query complete; 1=Query incomplete; 0=None)	2	
1.6.6 Filter: Positive Difference (2=Query complete; 1=Query incomplete; 0=None)	2	
1.6.7 Sort: Name (2=Query complete; 1=Query incomplete; 0=None)	2	
1.6.8 Sort: Games Won (2=Query complete; 1=Query incomplete; 0=None)	2	
1.6.9 Sort: Points (2=Query complete; 1=Query incomplete; 0=None)	2	
1.7 Search/show team from Edit (4=Query complete; 2=Query incomplete; 0=None)	4	
1.8 RadioGroup Query – Where (1 mark per group query)	6	
<b>TOTAL</b>	<b>40</b>	

**Question 2**

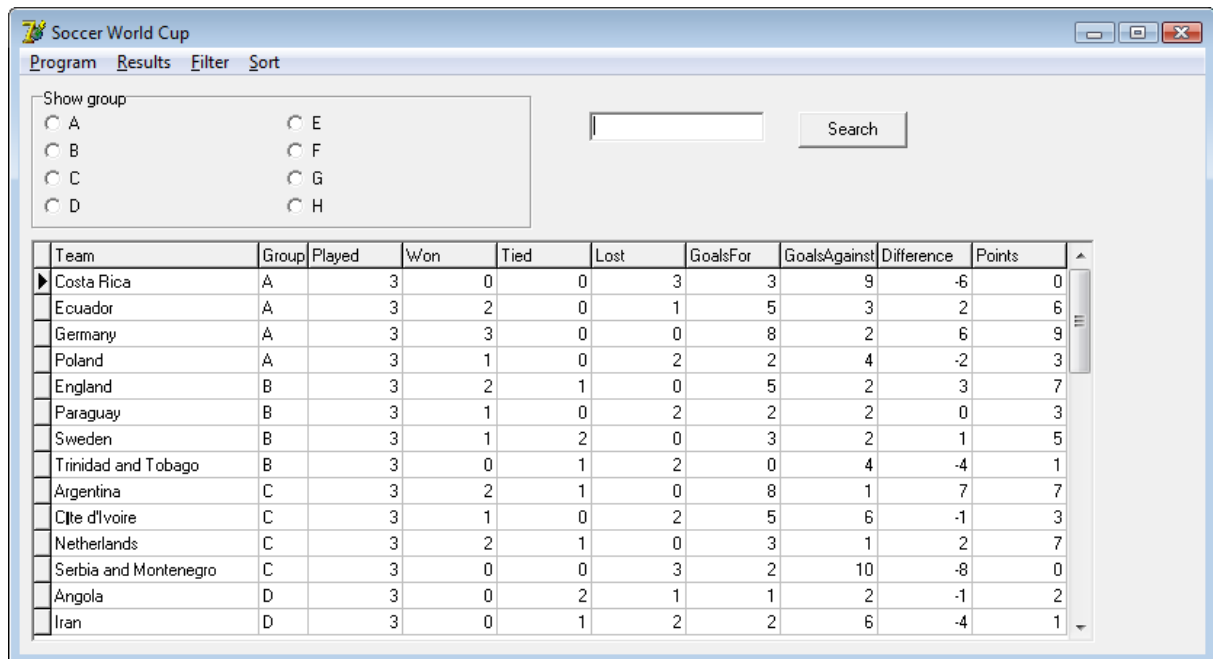
2.1.1 Private fields (2=All fields; 1=Some fields; 0=None)	2	
2.1.2 Create constructor (1 mark each: fCity, fName, fSeats and categories)	4	
2.1.3 Method: getStadium (1 mark each: fCity; fName; fSeats and categories; tabs)	4	
2.1.4 Method: getCity	1	
2.1.5 Methods: getCat (1 mark for the seats and 1 for amount x 4)	8	
2.2.1.1 Assign file (2=assigned to correct variable; 1=assigned to any variable; 0=none)	2	
2.2.1.2 Use counter (iCount)	1	
2.2.1.3 Isolate items and store in variables (While loop; Readln; Copy; Delete)	4	
2.2.1.4 TStadium object – array arrStadiums	2	
2.2.2 Button: btnShowAll (1 mark each: tab stops; headings; invoking getStadium)	3	
2.2.3 Button: btnCategory (1 mark Tab stops; 1 mark headings; 2 marks For loop)	4	
2.2.4 Button: btnSearch (1 mark Tab stops; 1 mark heading; 3 search)	5	
<b>TOTAL</b>	<b>40</b>	

**Question 3**

3.1 Components (1 mark each: btnGetData, btnCount, sgdSeats, pnlResult, lblEmpty, lblSeats)	6	
3.2 Letters A to J in fixed row	1	
3.3 Button: btnGetData (1 mark variables; 2 marks: assignfile, reset, for loop, readln, for loop, close file)	15	
3.4 Button: btnCount (1 mark: initial values for 2 variables; 2 marks: variables, for loop, for loop, if statement, increase of variable, adding seat values, display of empty blocks, display of seats taken)	18	
<b>TOTAL</b>	<b>40</b>	

**TOTAL: 120**

**QUESTION 1**



```
// 1.1 edtName, btnSearch, rdgGroups, mnuSoccer (2)
```

```
// 1.2 mnuSoccer items (2)
```

```
// 1.3 rgpGroups (2)
```

```
// 1.4 qryResults, datSoccer, dbgOutput (3)
```

```
unit Question1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, DB, Grids, DBGrids, ADODB, Menus, ExtCtrls;
```

```
type
```

```
TForm1 = class(TForm)  
    qryResults: TADOQuery;  
    dbgOutput: TDBGrid;  
    datSoccer: TDataSource;  
    mnuSoccer: TMainMenu;  
    Program1: TMenuItem;  
    Exit1: TMenuItem;  
    Results1: TMenuItem;  
    ShowAll1: TMenuItem;  
    DeterminedDifference1: TMenuItem;  
    Filter1: TMenuItem;  
    eamwithresults1: TMenuItem;  
    PositiveDifference1: TMenuItem;  
    Sort1: TMenuItem;  
    Name1: TMenuItem;  
    Gameswon1: TMenuItem;  
    Poinys1: TMenuItem;  
    ZeroDifference1: TMenuItem;  
    rgpGroups: TRadioGroup;  
    WinPoint1: TMenuItem;  
    edtName: TEdit;  
    btnSearch: TButton;  
    procedure Exit1Click(Sender: TObject);  
    procedure ShowAll1Click(Sender: TObject);
```

```
    procedure GetDifference1Click(Sender: TObject);
    procedure ZeroDifference1Click(Sender: TObject);
    procedure WinPoint1Click(Sender: TObject);
    procedure Teamwithwins1Click(Sender: TObject);
    procedure PositiveDifference1Click(Sender: TObject);
    procedure Name1Click(Sender: TObject);
    procedure Gameswon1Click(Sender: TObject);
    procedure Points1Click(Sender: TObject);
    procedure btnSearchClick(Sender: TObject);
    procedure rgpGroupsClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.Exit1Click(Sender: TObject);
begin
    Application.Terminate;           // 1.5 (1)
end;

procedure TForm1.ShowAll1Click(Sender: TObject);
begin
    qryResults.Active := false;     // 1.6.1 (2)
    qryResults.SQL.Text := 'Select * FROM tblResults'; ✓✓
    qryResults.Active := true;
end;

procedure TForm1.GetDifference1Click(Sender: TObject);
begin
    qryResults.Active := false;     // 1.6.2 (3)
    qryResults.SQL.Text := 'UPDATE tblResults SET Difference = GoalsFor -
GoalsAgainst'; ✓✓
    qryResults.ExecSQL;
    qryResults.SQL.Text := 'Select * FROM tblResults'; ✓
    qryResults.Active := true;
end;

procedure TForm1.ZeroDifference1Click(Sender: TObject);
begin
    qryResults.Active := false;     // 1.6.3 (2)
    qryResults.SQL.Text := 'UPDATE tblResults SET Difference = 0'; ✓
    qryResults.ExecSQL;
    qryResults.SQL.Text := 'Select * FROM tblResults'; ✓
    qryResults.Active := true;
end;

procedure TForm1.WinPoint1Click(Sender: TObject);
begin
    qryResults.Active := false;     // 1.6.4 (3)
    qryResults.SQL.Text := 'Select Team, [Won]*25 AS WinPoint FROM tblResults ORDER
BY Team'; ✓✓✓
    qryResults.Active := true;
end;

procedure TForm1.Teamwithwins1Click(Sender: TObject);
begin
    qryResults.Active := false;     // 1.6.5 (2)
    qryResults.SQL.Text := 'Select * FROM tblResults WHERE Won > 0'; ✓✓
    qryResults.Active := true;
end;
```

```
procedure TForm1.PositiveDifference1Click(Sender: TObject);
begin
  qryResults.Active := false; // 1.6.6 (2)
  qryResults.SQL.Text := 'Select * FROM tblResults WHERE Difference >= 0'; ✓✓
  qryResults.Active := true;
end;

procedure TForm1.Name1Click(Sender: TObject);
begin
  qryResults.Active := false; // 1.6.7 (2)
  qryResults.SQL.Text := 'Select * FROM tblResults ORDER BY Team'; ✓✓
  qryResults.Active := true;
end;

procedure TForm1.Gameswon1Click(Sender: TObject);
begin
  qryResults.Active := false; // 1.6.8 (2)
  qryResults.SQL.Text := 'Select * FROM tblResults ORDER BY Won'; ✓✓
  qryResults.Active := true;
end;

procedure TForm1.Points1Click(Sender: TObject);
begin
  qryResults.Active := false; // 1.6.9 (2)
  qryResults.SQL.Text := 'Select * FROM tblResults ORDER BY Points DESC'; ✓✓
  qryResults.Active := true;
end;

procedure TForm1.btnSearchClick(Sender: TObject);
begin
  qryResults.Active := false; // 1.7 (4)
  qryResults.SQL.Text := 'Select * FROM tblResults WHERE Team Like ' + ''' +
  edtName.Text + '%' + '''; ✓✓✓✓
  qryResults.Active := true;
end;

procedure TForm1.rgpGroupsClick(Sender: TObject);
begin
  Case rgpGroups.ItemIndex of // 1.8 (6)
    0 : begin
      qryResults.Active := false; ✓
      qryResults.SQL.Text := 'Select * FROM tblResults WHERE (Group='A')';
      qryResults.Active := true;
    end;
    1 : begin
      qryResults.Active := false; ✓
      qryResults.SQL.Text := 'Select * FROM tblResults WHERE (Group='B')';
      qryResults.Active := true;
    end;
    2 : begin
      qryResults.Active := false; ✓
      qryResults.SQL.Text := 'Select * FROM tblResults WHERE (Group='C')';
      qryResults.Active := true;
    end;
    3 : begin
      qryResults.Active := false; ✓
      qryResults.SQL.Text := 'Select * FROM tblResults WHERE (Group='D')';
      qryResults.Active := true;
    end;
    4 : begin
      qryResults.Active := false; ✓
      qryResults.SQL.Text := 'Select * FROM tblResults WHERE (Group='E')';
      qryResults.Active := true;
    end;
    5 : begin
      qryResults.Active := false; ✓
      qryResults.SQL.Text := 'Select * FROM tblResults WHERE (Group='F')';
    end;
  end;
end;
```

```
        qryResults.Active := true;
    end;
6 : begin
    qryResults.Active := false;
    qryResults.SQL.Text := 'Select * FROM tblResults WHERE (Group='G');';
    qryResults.Active := true;
    end;
7 : begin
    qryResults.Active := false;
    qryResults.SQL.Text := 'Select * FROM tblResults WHERE (Group='H');';
    qryResults.Active := true;
    end;
end;
end.
end.
```

**QUESTION 2**

---

```
// CREATED UNIT

unit StadiumUnit;

interface

uses SysUtils;

type
    TStadium = Class
    private
        fCity : String;
        fName : String;
        fSeats : Integer;
        fCategory1 : Integer;
        fCategory2 : Integer;
        fCategory3 : Integer;
        fCategory4 : Integer;

    public
        constructor create(City, Name : String; Seats, Cat1, Cat2, Cat3, Cat4 :
            Integer);
        function getStadium : String;
        function getCity : String;
        function getCat1 : String;
        function getCat2 : String;
        function getCat3 : String;
        function getCat4 : String;
    end;

implementation

{ TStadium }

constructor TStadium.create(City, Name : String; Seats, Cat1, Cat2, Cat3, Cat4 :
Integer);
begin
    fCity := City;
    fName := Name;
    fSeats := Seats;
    fCategory1 := Cat1;
    fCategory2 := Cat2;
    fCategory3 := Cat3;
    fCategory4 := Cat4;
end;
```

```
function TStadium.getStadium: String; // 2.1.3 (4)
begin
  Result := fCity + #9 + fName + #9 + IntToStr(fSeats) + ✓✓✓✓
  #9 + IntToStr(fCategory1) + #9 + IntToStr(fCategory2) + #9 +
  IntToStr(fCategory3) + #9 + IntToStr(fCategory4);
end;

function TStadium.getCity: String;
begin
  Result := fCity; ✓ // 2.1.4 (1)
end;

function TStadium.getCat1: String; // 2.1.5 (8)
begin
  Result := fName + #9 + FloatToStr((fCategory1/100)*fSeats) + #9 + 'R' +
  FloatToStr((fCategory1/100)*fSeats)*1120 ✓✓
end;

function TStadium.getCat2: String;
begin
  Result := fName + #9 + FloatToStr((fCategory2/100)*fSeats) + #9 + 'R' +
  FloatToStr((fCategory2/100)*fSeats)*840 ✓✓
end;

function TStadium.getCat3: String;
begin
  Result := fName + #9 + FloatToStr((fCategory3/100)*fSeats) + #9 + 'R' +
  FloatToStr((fCategory3/100)*fSeats)*560 ✓✓
end;

function TStadium.getCat4: String;
begin
  Result := fName + #9 + FloatToStr((fCategory4/100)*fSeats) + #9 + 'R' +
  FloatToStr((fCategory4/100)*fSeats)*140 ✓✓
end;

end.

// MAIN PROGRAM
unit Question2u;

interface

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

type
  TfrmStadiums = class(TForm)
    btnShowAll: TButton;
    redOutput: TRichEdit;
    btnCategory: TButton;
    cmbCategory: TComboBox;
    edtCity: TEdit;
    btnSearch: TButton;
    procedure FormActivate(Sender: TObject);
    procedure btnShowAllClick(Sender: TObject);
    procedure btnCategoryClick(Sender: TObject);
    procedure btnSearchClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmStadiums: TfrmStadiums;
```

```

arrStadiums : array[1..10] of TStadium;
iCount : Integer;

implementation

{$R *.dfm}

procedure TfrmStadiums.FormActivate(Sender: TObject);
var
  fFile : TextFile;
  sTemp,sCity,sName : String;
  iSeats, iCat1, iCat2, iCat3, iCat4 : Integer;
begin
  iCount := 0;                               ✓ // 2.2.1.2 (1)
  AssignFile(fFile, 'stadiums.txt');         ✓✓ // 2.2.1.1 (2)
  Reset(fFile);
  While not EOF(fFile) do                    ✓ // 2.2.1.3 (4)
  begin
    Readln(fFile, sTemp);                    ✓
    sCity := copy(sTemp, 1, pos(',',sTemp)-1); ✓
    Delete(sTemp, 1, pos(',',sTemp));         ✓
    sName := copy(sTemp, 1, pos(',',sTemp)-1);
    Delete(sTemp, 1, pos(',',sTemp));
    iSeats := StrToInt(copy(sTemp, 1, pos(',',sTemp)-1));
    Delete(sTemp, 1, pos(',',sTemp));
    iCat1 := StrToInt(copy(sTemp, 1, pos(',',sTemp)-1));
    Delete(sTemp, 1, pos(',',sTemp));
    iCat2 := StrToInt(copy(sTemp, 1, pos(',',sTemp)-1));
    Delete(sTemp, 1, pos(',',sTemp));
    iCat3 := StrToInt(copy(sTemp, 1, pos(',',sTemp)-1));
    Delete(sTemp, 1, pos(',',sTemp));
    iCat4 := StrToInt(copy(sTemp, 1, length(sTemp)));
    inc(iCount);
    arrStadiums[iCount] :=
      TStadium.create(sCity,sName,iSeats,iCat1,iCat2,iCat3,iCat4); // 2.2.1.4 (2)
  end;                                       ✓✓
  CloseFile(fFile);
end;

procedure TfrmStadiums.btnShowAllClick(Sender: TObject);
var
  iFCount : Integer;
begin
  redOutput.Clear;                          // 2.2.2 (3)
  redOutput.Paragraph.TabCount := 6;        ✓
  redOutput.Paragraph.Tab[0] := 75;
  redOutput.Paragraph.Tab[1] := 200;
  redOutput.Paragraph.Tab[2] := 250;
  redOutput.Paragraph.Tab[3] := 275;
  redOutput.Paragraph.Tab[4] := 300;
  redOutput.Paragraph.Tab[5] := 325;
  redOutput.Lines.Add('CITY' + #9 + 'STADIUM' + #9 + 'SEATS' + #9 + 'CAT1' + #9 +
  'CAT2' + #9+ 'CAT3' + #9+ 'CAT4');        ✓
  FOR iFCount := 1 to iCount do
  begin
    redOutput.Lines.Add(arrStadiums[iFCount].getStadium); ✓
  end;
end;

procedure TfrmStadiums.btnCategoryClick(Sender: TObject); // 2.2.3 (4)
var
  iFCount : Integer;
begin
  redOutput.Clear;
  redOutput.Paragraph.TabCount := 2;        ✓
  redOutput.Paragraph.Tab[0] := 125;
  redOutput.Paragraph.Tab[1] := 170;

```

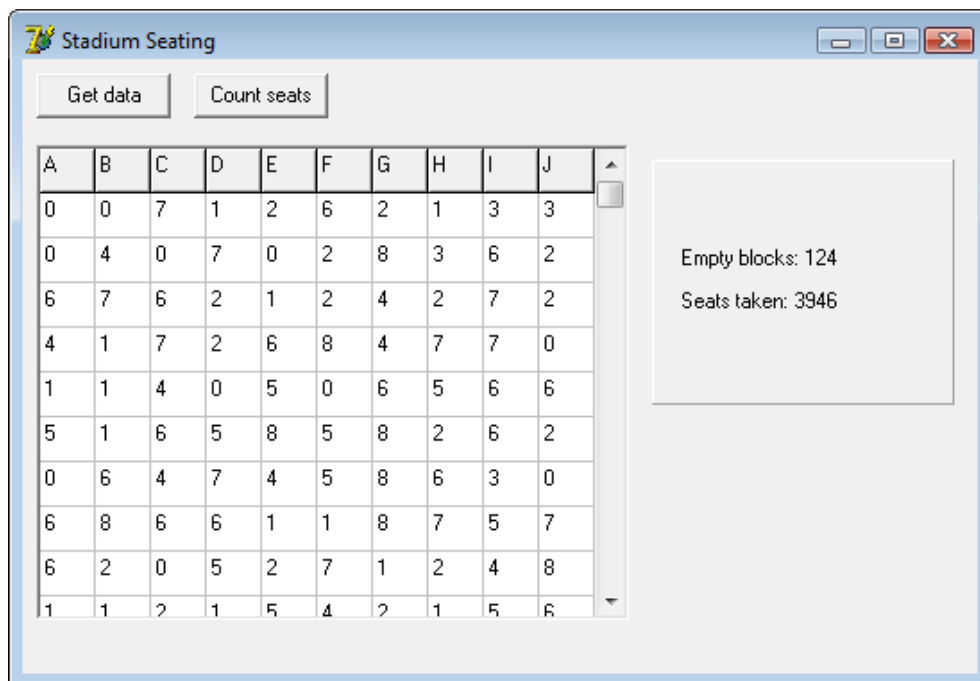
```

redOutput.Lines.Add('STADIUM' + #9 + 'SEATS' + #9 + 'AMOUNT'); ✓
FOR iFCount := 1 to iCount do ✓✓
begin
  if cmbCategory.ItemIndex = 0 then
    redOutput.Lines.Add(arrStadiums[iFCount].GetCat1);
  if cmbCategory.ItemIndex = 1 then
    redOutput.Lines.Add(arrStadiums[iFCount].GetCat2);
  if cmbCategory.ItemIndex = 2 then
    redOutput.Lines.Add(arrStadiums[iFCount].GetCat3);
  if cmbCategory.ItemIndex = 3 then
    redOutput.Lines.Add(arrStadiums[iFCount].GetCat4);
  end;
end;

procedure TfrmStadiums.btnSearchClick(Sender: TObject); // 2.2.4 (5)
var
  iFCount : Integer;
begin
  redOutput.Clear;
  redOutput.Paragraph.TabCount := 6; ✓
  redOutput.Paragraph.Tab[0] := 75;
  redOutput.Paragraph.Tab[1] := 200;
  redOutput.Paragraph.Tab[2] := 250;
  redOutput.Paragraph.Tab[3] := 275;
  redOutput.Paragraph.Tab[4] := 300;
  redOutput.Paragraph.Tab[5] := 325;
  redOutput.Lines.Add('CITY' + #9 + 'STADIUM' + #9 + 'SEATS' + #9 + 'CAT1' + #9 +
  'CAT2' + #9 + 'CAT3' + #9 + 'CAT4'); ✓
  FOR iFCount := 1 to iCount do
  begin
    if edtCity.Text = arrStadiums[iFCount].GetCity then
      redOutput.Lines.Add(arrStadiums[iFCount].GetStadium); ✓✓✓
    end;
  end;
end.
end.

```

**QUESTION 3**



// 3.1 Components (6)

```
unit Question3u;

interface

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

type
  TfrmSeating = class(TForm)
    sgdSeats: TStringGrid;
    btnGetData: TButton;
    btnCount: TButton;
    pnlResult: TPanel;
    lblEmpty: TLabel;
    lblSeats: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure btnGetDataClick(Sender: TObject);
    procedure btnCountClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmSeating: TfrmSeating;

implementation

{$R *.dfm}
procedure TfrmSeating.FormCreate(Sender: TObject);           // 3.2 (1)
begin
  sgdSeats.Cells[0,0] := 'A';
  sgdSeats.Cells[1,0] := 'B';
  sgdSeats.Cells[2,0] := 'C';
  sgdSeats.Cells[3,0] := 'D';
  sgdSeats.Cells[4,0] := 'E';
  sgdSeats.Cells[5,0] := 'F';
  sgdSeats.Cells[6,0] := 'G';
  sgdSeats.Cells[7,0] := 'H';
  sgdSeats.Cells[8,0] := 'I';
  sgdSeats.Cells[9,0] := 'J';
end;

procedure TfrmSeating.btnGetDataClick(Sender: TObject);     // 3.3 (15)
var
  fFile : TextFile;
  sTemp : String;
  iCount, iRow : Integer;
begin
  AssignFile(fFile, 'seats.dat');
  Reset(fFile);
  for iCount := 1 to 100 do
  begin
    readln(fFile, sTemp);
    For iRow := 1 to 10 do
    begin
      sgdSeats.Cells[iRow-1, iCount] := sTemp[iRow];
    end;
  end;
  CloseFile(fFile);
end;

procedure TfrmSeating.btnCountClick(Sender: TObject);      // 3.4 (18)
var
  iCol, iRow, iEmpty, iSeats : Integer;
begin
```

```
iEmpty := 0; ✓  
iSeats := 0; ✓  
for iRow := 1 to 100 do ✓✓  
begin  
  For iCol := 0 to 9 do ✓✓  
  begin  
    if sgdSeats.Cells[iCol, iRow] = '0' then ✓✓  
    begin  
      iEmpty := iEmpty + 1; ✓✓  
    end;  
    iSeats := iSeats + StrToInt(sgdSeats.Cells[iCol, iRow]); ✓✓  
  end;  
end;  
lblEmpty.Caption := 'Empty blocks: ' + IntToStr(iEmpty); ✓✓  
lblSeats.Caption := 'Seats taken: ' + IntToStr(iSeats); ✓✓  
end;  
  
end.
```

**TOTAL: 120**