Province of the
EASTERN CAPE
EDUCATION

NATIONAL SENIOR CERTIFICATE

GRADE 12

SEPTEMBER 2014

INFORMATION TECHNOLOGY P1 MEMORANDUM

MARKS: **120**

This memorandum consists of 22 pages.

QUESTION 1: DELPHI PROGRAMMING

Mark allocation

Question 1 - Marking Grid				
Q	**Aspect**		**Max. marks**	**Learner's marks**
1.1	1.1.1	Function CCValidate(creditcarnumber:string):boolean;	2	

	1.1.2	CleanCardNumber:='';
		digit:=0;
		
		// Remove any non numeric value
		for I := 1 to Length(CreditCardNumber) do✓
		Begin
		if CreditCardNumber[i] in DigitsAllowed then✓ ✓
		CleanCardNumber:= CleanCardNumber + CreditCardNumber[i]; ✓✓
		End;
			5	

	1.1.3	// Check for valid card maximum length number
		if (Length(CleanCardNumber)>MaxCCSize) then ✓
		Begin
		Result:= True;
		Exit;
		End;
		//Check for valid card minimum length number
		if (Length(CleanCardNumber)<MinCCSize) then ✓
		Begin
		Result:= False; ✓
		Exit;
		End;
			3	

	1.1.4	Carrying out the Luhn Algorithm, the student may use any method as long as they accomplish the task, iterations, loops and decision making, rearranging techniques must be applied.	11	

	1.1.5	procedure TForm1.Button1Click(Sender: TObject);
		begin
		If CCValidate(Edit1.text)=true✓✓
		then
		Showmessage('Your Card is Valid') ✓
		Else
		Showmessage('Card Number is Invalid'); ✓
		
		end;
			4	

1.2	1.2.1	Type TArr=Array[1..10] integer; Var Form1:TForm1; ListA, ListB: TArr; ✓ ---------- Begin Randomize; For i:= 1 to 10 do✓	4	
		Begin ListA[i]:= random(10)+10; ✓ Richedit1.Lines.Add(IntToStr(ListA[i])); ✓		

	1.2.2	Var
		K,X, Y,iCntP :integer;
		Dup : Boolean;
		begin
		
		iCntP := 0;
		For X := 1 to 10 do✓
		begin
		Y := X + 1;
		Dup := False;
		While (Dup = False) AND (Y <= 10) DO✓
		begin
		If ListA[X] = ListA[Y] then DUp := True; ✓
		y := Y + 1;
		end;
		If Not(Dup) ✓
		then begin
		iCntP := iCntP + 1;
		ListB[iCntP] := ListA[X]; ✓
		end;
		end;
		
		For K:=1 to iCntP do
		d.Lines.Add(IntToStr(ListB[K])); ✓
		
		end;
			6	

			[35]	

SAMPLE: DELPHI SOLUTION QUESTION 1

```
unit Q_1;

interface

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

type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    lblDisplay: TLabel;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}
function CCValidate(CreditCardNumber:string):boolean;

Const
DigitsAllowed = ['0'..'9'];
MaxCCSize = 19;
MinCCSize = 13;

var
i : integer;
CleanCardNumber: string;
digit: Integer;
CheckSum: Integer; { Holds the value of the operation }
Flag: Boolean; { used to indicate when ready }
Counter: Integer; { index counter }
PartNumber: string; { used to extract each digit of number }
Number: Integer; { used to convert each digit to integer}


Begin
CleanCardNumber:='';
digit:=0;

// Remove any non numeric value
for I := 1 to Length(CreditCardNumber) do
Begin
if CreditCardNumber[i] in DigitsAllowed then
CleanCardNumber:= CleanCardNumber + CreditCardNumber[i];
End;
```



```
// Check for valid card maximum length number
if (Length(CleanCardNumber)>MaxCCSize) then
Begin
Result:= False;
Exit;
End;
//Check for valid card minimum length number
if (Length(CleanCardNumber)<MinCCSize) then
Begin
Result:= False;
Exit;
End;

// get the starting value for our counter
Counter := Length(CleanCardNumber);
CheckSum := 0;
PartNumber := '';
Number := 0;
Flag := true;

while (Counter >= 1) do
begin
// get the current digit
PartNumber := Copy(CleanCardNumber, Counter, 1);
Number := StrToInt(PartNumber); // convert to integer
if (Flag) then // only do every other digit
begin
Number := Number * 2;
if (Number >= 10) then
Number := Number - 9;
end;
CheckSum := CheckSum + Number;

Flag := not (Flag);

Counter := Counter - 1;
end;

result := ((CheckSum mod 10) = 0);

End;

procedure TForm1.Button1Click(Sender: TObject);
begin
  If  CCValidate(Edit1.text)=true
  then
     Showmessage('Your Card is Valid' )
     Else
     Showmessage('Card Number is Invalid');


end;

procedure TForm1.GenerateClick(Sender: TObject);
var
i:integer;
```



```
begin
Randomize;
for i := 1 to 10 do
begin
  ListA[i]:= random(10)+10;
  Richedit1.Lines.Add(IntToStr(ListA[i]));
 end;
end;



procedure TForm1.DisplayClick(Sender: TObject);
Var
K,X, Y,iCntP :integer;
Dup : Boolean;
begin

 iCntP := 0;
 For X := 1 to 10 do
  begin
    Y := X + 1;
    Dup := False;
    While (Dup = False) AND ( Y <= 10) DO
     begin
      If ListA[X] = ListA[Y] then DUp := True;
       y := Y + 1;
     end;
    If Not(Dup)
      then begin
           iCntP := iCntP + 1;
            ListB[iCntP] := ListA[X];
           end;
      end;


For K:=1 to iCntP do
  d.Lines.Add(IntToStr(ListB[K]));

end;

end.
```


QUESTION 2: DELPHI PROGRAMMING

	Question 2 - Marking Grid			
Q	**Aspect**		**Max. marks**	**Learner's marks**
2.1	2.1.1	Defining of the class TDriver `TDriver = Class(TObject)` ✓	1	
	2.1.2	Private fields `fName:string;` ✓ `fCountry:String;` ✓ `fTeam:string;` ✓ `fTime:real;` ✓	4	
	2.1.3	Defining the parameterised constructor correctly: `constructor TDriver.Create(sName, sCountry, sTeam,sTime: string);` ✓✓ Receiving values from the parameters and equating them to the private fields: `fName:=sName;` ✓ `fCountry:=sCountry;` ✓ `fTeam:=sTeam;` ✓ `fTime:=sTime;` ✓ Adding the appropriate Uses statement in the interface section of the program `uses` `SysUtils, StdCtrls, ComCtrls;` ✓	7	
	2.1.4	Declaring The method correctly `function TDriver.getfName: string;` ✓ Getting its value correctly `Result:=copy(fName,1, pos(' ',fName)-1)` ✓ ✓	3	
	2.1.5	Declaring The method correctly, can use either string or real as long as learner uses it appropriately in the program `function TDriver.getAverage: string;` ✓ Might use an internal variable to calculate the average `sAverage,aTime:real;` Calculation and result done correctly rounding to nearest whole number `aTime:=3600* strtoint(copy(fTime,1,1))+60*strtoint(copy(fTime,3,2))+strtofloat(copy(fTime,6,6));` ✓ `sAverage:=(58*5.95*1000)/aTime;` ✓ `Result:='The Average speed of '+fName+ '` ✓ `is '+floattostr(sAverage) +'m/s';` ✓	5	

	2.1.6	Declaring The method correctly
		`function TDriver.toString: string;` ✓
		
		Getting its value correctly
		`shrs:=strtoint(copy(fTime,1,1));` ✓
		`smins:=strtoint(copy(fTime,3,2));` ✓
		`sseconds:=strtofloat(copy(fTime,6,6));` ✓
		`result :=fName`
		`+#9+fCountry+#9+fTeam+#9+inttostr(shrs)+':'+in`
		`ttostr(smins)+'":'+floattostr(sseconds);` ✓✓
			6	

	2.2.1	`uses` `Datam;` ✓ `AUDatam.ADOTaugp3.Active:=True;`✓	2	
	2.3.1	<u>Code on the Button</u> `uses Q2_aurgpunit;` ✓ `Q2_aurgp2.Showmodal;` ✓ <u>On the Form</u> `AUDatam.ADOTaugp3.Active:=True;` ✓ `AUDatam.ADOTaugp3.Sort:='Time';` ✓	4	
	2.3.2	<u>Reading the database</u> `AUDatam.ADOTaugp3.Open;` ✓ `AUDatam.ADOTaugp3.First;` ✓ <u>CREATING THE OBJECT</u> `Winner:=Tdriver.Create`✓ ✓ `(AUDatam.ADOTaugp3['Name'],` ✓ `AUDatam.ADOTaugp3['Country'],` ✓ `AUDatam.ADOTaugp3['Team'],` ✓ `AUDatam.ADOTaugp3['Time']);` ✓ `showmessage(Winner.toString);` ✓✓	10	
	2.4.1	`Uses` `Q2_aurgp3unit;` ✓ `Q2_aurgp3unit;.showmodal;` ✓	2	
	2.4.2	Placing all the components required	3	

	2.4.3	Call Procedure load and Procedure heading;
		begin
		if ((Button = nbFirst) or (Button = nbNext) ✓✓
		or (Button = nbPrior) or (Button = nbLast)) ✓✓
		then
		begin
		Mdriver:=Tdriver.Create(DBedit2.Text, ✓✓
		DBEdit3.Text, ✓
		DBEdit4.Text✓,
		DBEdit5.Text); ✓
		lblavgdisp.Caption:=Mdriver.getAverage; ✓✓
		end;
		end;
			11	

	2.5	Call the procedure load and procedure heading
		begin
		var
		myFile:textfile;
		DrvID , Name, Country : String;
		iTotal : Integer;
		begin
		AssignFile(myFile,'Drivers.txt'); ✓
		Rewrite(myFile); ✓
		
		Writeln(myFile,'DRIVERS AND THEIR COUNTRIES':50); ✓
		
		Writeln(myFile,'==========':50); ✓
		Writeln(myFile,'');
		Writeln(myFile,'DriverID':20, 'Name and Surname':30, 'Country':20); ✓✓
		Writeln(myFile,'------':20,'----------':30,'----------':20);
		Writeln(myFile,'');
		AUDatam.ADOTaugp3.Open; ✓
		AUDatam.ADOTaugp3.First; ✓
		while not AUDatam.ADOTaugp3.Eof do✓
		begin
		DrvID:=AUDatam.ADOTaugp3['DRID']; ✓
		Name:= AUDatam.ADOTaugp3['Name']; ✓
		Country:= AUDatam.ADOTaugp3['Country']; ✓
		if Country <>'Great Britain' then✓
		
		Writeln(myFile,DrvID:20,Name:30,Country:20); ✓
		AUDatam.ADOTaugp3.Next; ✓
		end;
		CloseFile(myFile); ✓
			16	

			[74]	

SAMPLE: DELPHI SOLUTION QUESTION 2

Object Class (Games_U)

```
unit clsDriver;

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, ComCtrls, Buttons;
type
//=========================================================================
=======
TDriver = Class(TObject)
  Private
  fName:string;
  fCountry:String;
  fTeam:String;
  fTime:string;
  Public

  Constructor Create(sName,sCountry,sTeam,sTime:string);
  Function toString:string;
  Function getfName:string;
  Function getAverage:string;

End;

implementation

{ TDriver }

//=========================================================================
=====
constructor TDriver.Create(sName, sCountry, sTeam,sTime: string);
begin
  fName:=sName;
  fCountry:=sCountry;
  fTeam:=sTeam;
  fTime:=sTime;
end;
//=========================================================================
=======
function TDriver.getAverage: string;
 Var
   sAverage,aTime:real;
begin
  aTime:=3600*
strtoint(copy(fTime,1,1))+60*strtoint(copy(fTime,3,2))+strtofloat(copy(fTi
me,6,6));
  sAverage:=(58*5.95*1000)/aTime;
  Result:='The Average speed of '+fName+ ' is '+floattostr(sAverage)
+'m/s';
end;

//=========================================================================
=====
function TDriver.getfName: string;
Var
cpos:integer;
begin
 cpos:=pos(' ',fName);
 Result:=copy(fName,1,cpos-1);
```



```
end;


//=========================================================================
=======
function TDriver.toString: string;
var
shrs, smins:integer;
sseconds:real;
begin
  shrs:=strtoint(copy(fTime,1,1));
  smins:=strtoint(copy(fTime,3,2));
  sseconds:=strtofloat(copy(fTime,6,6));
result :=fName
+#9+fCountry+#9+fTeam+#9+inttostr(shrs)+':'+inttostr(smins)+'":'+floattost
r(sseconds);
end;

end.
```

Main Form

```
unit Q2_aurgp;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, ComCtrls, Buttons, clsDriver, Grids, DBGrids, Datam;

type
  TfrmAustralia = class(TForm)
    btnDisplay: TButton;
    btnResults: TButton;
    BitBtn1: TBitBtn;
    DBGrid1: TDBGrid;
    btnNavigate: TButton;
    btnSave: TButton;
    procedure btnDisplayClick(Sender: TObject);
    procedure btnResultsClick(Sender: TObject);


    procedure btnNavigateClick(Sender: TObject);
    procedure btnSaveClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }


  end;

var
  frmAustralia: TfrmAustralia;
implementation

uses Q2_aurgpunit, Q2_aurgp3unit;

{$R *.dfm}

//=========================================================================
======
procedure TfrmAustralia.btnDisplayClick(Sender: TObject);
 begin
AUDatam.ADOTaugp3.Active:=True;

end;

//=========================================================================
======
procedure TfrmAustralia.btnResultsClick(Sender: TObject);
begin
Q2_aurgp2.Showmodal;
end;

procedure TfrmAustralia.btnSaveClick(Sender: TObject);
var
myFile:textfile;
DrvID , Name, Country : String;
iTotal : Integer;
begin
  AssignFile(myFile,'Divers.txt');
  Rewrite(myFile);
```

```
  Writeln(myFile,'DRIVERS AND THEIR COUNTRIES':50);
  Writeln(myFile,'==============================':50);
  Writeln(myFile,'');
  Writeln(myFile,'DriverID':20, 'Name and Surname':30, 'Country':20);
  Writeln(myFile,'------':20,'---------------':30,'--------------':20);
  Writeln(myFile,'');
     AUDatam.ADOTaugp3.Open;
     AUDatam.ADOTaugp3.First;
      while not AUDatam.ADOTaugp3.Eof do
        begin
         DrvID:=AUDatam.ADOTaugp3['DRID'];
         Name:= AUDatam.ADOTaugp3['Name'];
         Country:= AUDatam.ADOTaugp3['Country'];
         if Driver.Country <>'Great Britain' then
          Writeln(myFile,Driver.Getname +#9+Driver.getcountry);
         AUDatam.ADOTaugp3.Next;
        end;
    CloseFile(myFile);

end;

//=========================================================================
======
procedure TfrmAustralia.btnNavigateClick(Sender: TObject);
begin
Q2_aurgp3.Showmodal;
end;

//=========================================================================
======




end.
```

Form2// Button Display Race Results

```
unit Q2_aurgpunit;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, Buttons,Datam, Grids, DBGrids,clsDriver;

type
  TQ2_aurgp2 = class(TForm)
    BitBtn1: TBitBtn;
    DBGrid1: TDBGrid;
    btnWinner: TButton;
    procedure FormActivate(Sender: TObject);
    procedure btnWinnerClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Q2_aurgp2: TQ2_aurgp2;
  Winner:Tdriver;

implementation

{$R *.dfm}

procedure TQ2_aurgp2.btnWinnerClick(Sender: TObject);
begin

      AUDatam.ADOTaugp3.Open;
      AUDatam.ADOTaugp3.First;
      Winner:=Tdriver.Create(AUDatam.ADOTaugp3['Name'],
      AUDatam.ADOTaugp3['Country'],
      AUDatam.ADOTaugp3['Team'],
      AUDatam.ADOTaugp3['Time']);
      showmessage(Winner.toString);


end;

procedure TQ2_aurgp2.FormActivate(Sender: TObject);
begin
   AUDatam.ADOTaugp3.Active:=True;
   AUDatam.ADOTaugp3.Sort:='Time';
end;

end.
```


Form 3// Button Navigate

```
unit Q2_aurgp3unit;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs,Datam, ExtCtrls, DBCtrls, StdCtrls, Mask, Buttons, clsDriver;

type
  TQ2_aurgp3 = class(TForm)
    DBNavigator1: TDBNavigator;
    Label1: TLabel;
    DBEdit1: TDBEdit;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    DBEdit4: TDBEdit;
    Label5: TLabel;
    DBEdit5: TDBEdit;
    BitBtn1: TBitBtn;
    lblAverage: TLabel;
    lblavgdisp: TLabel;
    procedure FormActivate(Sender: TObject);
    procedure DBNavigator1Click(Sender: TObject; Button: TNavigateBtn);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Q2_aurgp3: TQ2_aurgp3;
  Mdriver:Tdriver;


implementation

{$R *.dfm}

procedure TQ2_aurgp3.DBNavigator1Click(Sender: TObject; Button:
TNavigateBtn);
begin
   if ((Button = nbFirst) or (Button = nbNext)
      or (Button = nbPrior) or (Button = nbLast)) then
    begin
       Mdriver:=Tdriver.Create(DBedit2.Text,
       DBEdit3.Text,
       DBEdit4.Text,
       DBEdit5.Text);
       lblavgdisp.Caption:=Mdriver.getAverage;
    end;
end;

procedure TQ2_aurgp3.FormActivate(Sender: TObject);
begin
   AUDatam.ADOTaugp3.Active:=True;
end;

end.
```


QUESTION 3: DELPHI PROGRAMMING

Mark allocation

Question 3 - Marking Grid			
Q	**Aspect**	**Max. mark**	**Learner's marks**

3.1	Introduce a counter to count the number of entries in the file
	`k:=0;` ✓
	Check to see if file exists and exit if it does not exist
	`if FileExists('abrivs.txt')<> true then`✓
	`Begin`
	`ShowMessage('File Does not exist');`
	`Exit;` ✓
	`End;`
	Assign and reset file
	`AssignFile(Myfile,'abrivs.txt' );` ✓
	`Reset(Myfile);` ✓
	Start a Loop
	`While Not EOF(Myfile) do`✓
	`inc(k);` ✓
	Read line by line
	`Readln( Myfile, SOneline);` ✓
	Enter correct values into array
	`ArrAbrevs[1,k]:=copy(SOneline,1,3);` ✓
	
	`ArrAbrevs[2,k]:=copy(SOneline,5,length(SOneline));` ✓
		10	

3.2	Loop to Display in Richedit
	`For z:=1 to k do`✓
	`Begin`
	Display correctly
	`redAbrevs.Lines.add(ArrAbrevs[1,z]+' '+ ArrAbrevs[2,z]);` ✓
	`end;`
	Nested loop to display positions in stringgrid
	`For i:=1 to 12 do`✓
	`begin`
	`For j:=1 to 10 do`✓
	`begin`
	ArrPositions displayed correctly
	`strgdPositions.Cells[j,i]:=ArrPositions[j,i];`
	`end;` ✓
	`end;`
	Loop to display circuits and numbers
	`For i:=1 to 10 do`✓
	`Begin`
	
	Circuits and Positions displayed correctly
	`strgdPositions.Cells[0,i]:= arrCircuits[i];` ✓
	`strgdPositions.Cells[i,0]:= inttostr(i);`
	`End;` ✓
		8	

3.3	Variable to count number of podium appearances to be initialised
	nump:=0; ✓
	Use inputbox correctly to get values
	iAbrev:=UpperCase(Inputbox('Driver Podium Search','Enter Abbreviations of Driver here','')); ✓✓
	Start loop to search abbreviation
	For i:=1 to 25 do✓
	Begin
	Link Abbreviation to Name
	if ArrAbrevs[1,i]=iAbrev then
	iName:=ArrAbrevs[2,i]; ✓✓
	End;
	Search podium positions on stringgrid
	for i := 1 to 10 do ✓
	Begin
	For j:=1 to 3 do
	Begin
	if strgdPositions.Cells[j,i]=iAbrev✓ **then**
	inc(nump); ✓
	End;
	End;
	if nump>0 then✓
	Display Podium podium positions using a message box
	ShowMessage(iName+' has been on the podium '+inttostr(nump)+' times so far') ✓
	Else✓**ShowMessage(iName+' has not been on the podium so far')** ✓
		13	

3.4	Use of Input box correct to get Abbreviation
	iAbrev:=UpperCase(Inputbox('Points Calculator','Enter Abbreviations of Driver here',''));✓✓
	Start loop to count
	for j := 1 to 10 do✓
	Begin
	for i:= 1 to 10 do✓
	if strgdPositions.Cells[i,j]= iAbrev then
	Correct use of Case statement or If Statement
	Begin
	Case i of
	1:iPoint:=25;
	2:iPoint:=18;
	3:iPoint:=15;
	4:iPoint:=12;
	5:iPoint:=10; ✓
	6:iPoint:=8; ✓
	7:iPoint:=6;
	8:iPoint:=4;
	9:iPoint:=2;
	10:iPoint:=1
	Else
	iPoint:=0; ✓
	End;
		10	

	End;
	Add Total
	iTotal:=iTotal+iPoint; ✓
	End;
	Link Name to Abbreviation
	For i:=1 to 25 do
	Begin
	if ArrAbrevs[1,i]=iAbrev then
	iName:=ArrAbrevs[2,i]; ✓
	End;
	Display Points
	ShowMessage(iName+' has '+inttostr(iTotal)+' points so far'); ✓
	end;		

		[41]	

SAMPLE: DELPHI SOLUTION QUESTION 3

```
unit Results;

interface

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

type
  TfrmSummaries = class(TForm)
    MainMenu1: TMainMenu;
    sRATRT1: TMenuItem;
    Load: TMenuItem;
    Display: TMenuItem;
    Podium: TMenuItem;
    Points: TMenuItem;
    redAbrevs: TRichEdit;
    strgdPositions: TStringGrid;
    Exit1: TMenuItem;
    Label1: TLabel;
    procedure LoadClick(Sender: TObject);
    procedure DisplayClick(Sender: TObject);
    procedure PodiumClick(Sender: TObject);
    procedure PointsClick(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmSummaries: TfrmSummaries;
  ArrAbrevs:Array[1..2,1..25] of string;
  ArrPositions:Array[1..10,1..15] of string =(('VET', 'VET', 'HAM',
'ROS', 'ROS', 'ROS', 'VET', 'HAM', 'HAM', 'HAM', 'HAM', 'VET', '',
'', ''),
  ('WEB', 'MAS', 'RAI', 'VET', 'HAM', 'HAM', 'HAM', 'ROS', 'VET',
'VET', 'VET', 'WEB', '', '', ''),
   ('HAM', 'ALO', 'ALO', 'ALO', 'VET', 'VET', 'BOT', 'VET', 'WEB',
'GRO', 'WEB', 'HUL', '', '', ''),
    ('MAS', 'HAM', 'ROS', 'MAS', 'RAI', 'WEB', 'ROS', 'WEB', 'RAI',
'ROS', 'ROS', 'MAS', '', '', ''),
     ('ALO', 'WEB', 'MAS', 'DIR', 'ALO', 'RAI', 'WEB', 'RIC',
'GRO', 'ALO', 'DIR', 'ALO', '', '', ''),
      ('ROS', 'ROS', 'GRO', 'SUT', 'GRO', 'ALO', 'ALO', 'SUT',
'RIC', 'RAI', 'BUT', 'ROS', '', '', ''),
       ('RAI', 'BUT', 'RIC', 'WEB', 'WEB', 'PER', 'VER', 'GRO',
'MAS', 'MAS', 'GRO', 'RIC', '', '', ''),
        ('GRO', 'SUT', 'BUT', 'RAI', 'PER', 'SUT', 'SUT', 'RAI',
'ALO', 'RIC', 'RAI', 'PER', '', '', ''),
         ('DIR', 'PER', 'VET', 'HAM', 'MAS', 'BUT', 'HUL', 'ALO',
'BUT', 'PER', 'ALO', 'BUT', '', '', ''),
          ('BUT', 'RAI', 'HUL', 'BUT', 'DIR', 'VER', 'RAI', 'BUT',
```



```
'HUL', 'WEB', 'MAS', 'VER', '', '', ''));
  ArrCircuits: Array[1..12] of string = ('Rolex Australian Grand
Prix','Petronas Malaysia Grand Prix','UBS Chinese Grand Prix','Gulf
Air Bahrain Grand Prix','57° Gran Premio de España','71e Grand Prix
de Monaco','Grand Prix du Canada','Santander British Grand
Prix','Grosser Preis Santander von Deutschland','Magyar
Nagydíj','Shell Belgian Grand Prix','84° Gran Premio dItalia');
  k:integer;

implementation

{$R *.dfm}

//=================================================================
=============
procedure TfrmSummaries.Exit1Click(Sender: TObject);
begin
Application.Terminate;
end;
 procedure TfrmSummaries.FormCreate(Sender: TObject);
begin
 strgdPositions.ColWidths[0]:=220;
end;

//3.1=============================================================
===============
procedure TfrmSummaries.LoadClick(Sender: TObject);
Var
  Myfile:TextFile;
  i:integer;
  SOneline:string;
begin
k:=0;
if FileExists('abrivs.txt')<> true then
  Begin
    ShowMessage('File Does not exist');
  Exit;
  End;
  AssignFile(Myfile,'abrivs.txt' );
  Reset(Myfile);
 While Not EOF(Myfile) do
     Begin
      inc(k);
      Readln( Myfile, SOneline);
      ArrAbrevs[1,k]:=copy(SOneline,1,3);
      ArrAbrevs[2,k]:=copy(SOneline,5,length(SOneline));

     End;


End;




//3.2=============================================================
===============
```

```
procedure TfrmSummaries.DisplayClick(Sender: TObject);
  Var
    z,i,j:integer;
begin
  For z:=1 to k do
    begin
      redAbrevs.Lines.add(ArrAbrevs[1,z]+'  '+ ArrAbrevs[2,z]);
    end;

   For i:=1 to 12 do
   begin
      For j:=1 to 10 do
         begin
            strgdPositions.Cells[j,i]:=ArrPositions[j,i];
         end;
   end;
  For i:=1 to 10  do
  Begin
    strgdPositions.Cells[0,i]:= arrCircuits[i];
    strgdPositions.Cells[i,0]:= inttostr(i);
  End;


end;

//3.3=============================================================
================
procedure TfrmSummaries.PodiumClick(Sender: TObject);
  Var
    i,j,nump:integer;
    iAbrev,iName:string;
begin
nump:=0;
iAbrev:=UpperCase(Inputbox('Driver Podium Search','Enter
Abbreviations of Driver here',''));

  For i:=1 to 25 do
      Begin
        if ArrAbrevs[1,i]=iAbrev then
          iName:=ArrAbrevs[2,i];
      End;

for i := 1 to 10  do
  Begin
    For j:=1 to 3 do
      Begin
        if strgdPositions.Cells[j,i]=iAbrev then
          inc(nump);
      End;
  End;




  if nump>0 then

  ShowMessage(iName+' has been on the podium '+inttostr(nump)+'
times so far')
```

```
  Else
  ShowMessage(iName+' has not been on the podium so far')

end;
//3.4================================================================
===============
procedure TfrmSummaries.PointsClick(Sender: TObject);
  Var
  i,j,iPoint,iTotal:integer;
  iName,iAbrev:string;

begin

 iAbrev:=UpperCase(Inputbox('Points Calculator','Enter
Abbreviations of Driver here',''));
 for j := 1 to 10 do
  Begin
    for i:= 1 to 10 do
      if strgdPositions.Cells[i,j]= iAbrev then
        Begin
          Case i of
            1:iPoint:=25;
            2:iPoint:=18;
            3:iPoint:=15;
            4:iPoint:=12;
            5:iPoint:=10;
            6:iPoint:=8;
            7:iPoint:=6;
            8:iPoint:=4;
            9:iPoint:=2;
            10:iPoint:=1
            Else
            iPoint:=0;
          End;
        End;

     iTotal:=iTotal+iPoint;
  End;
      For i:=1 to 25 do
      Begin
        if ArrAbrevs[1,i]=iAbrev then
          iName:=ArrAbrevs[2,i];
      End;
   ShowMessage(iName+' has  '+inttostr(iTotal)+'  points so far');
end;


end.
```

TOTAL: 150

