



**education**

Department:  
Education  
PROVINCE OF KWAZULU-NATAL

# **TRIAL EXAMINATION**

## **SEPTEMBER 2020**

### **MARKING GUIDELINES & MEMORANDUM**

#### **INFORMATION TECHNOLOGY P1**

**MARKS: 150**

**EXAMINERS : Mr N Brijlal  
Mrs S Bramdaw**

**TIME: 3 hours**

**MODERATOR: Ms EC Moodley**

**This Memorandum consists of 14 pages.**

## QUESTION 1

QN	Description	Max	Learner
1.1	<p><b>Correctly assigns property settings in panel:</b>            Panel Caption : 'Welcome to Rainbow Tours' ✓            Panel Color: <i>c!Yellow</i> ✓            Panel Font Style: <i>[fsBold]</i> ✓            Panel Font Size: <i>16</i> ✓</p> <p><b>Loading Image :</b>            Use of correct code : <i>ImgPic.Picture.LoadFromFile</i> ✓            Correct Image file loaded : 'Rainbow.jpg' ✓            Property set : <i>ImgPic.Stretch := true;</i> ✓</p>	7	
1.2	<p>Extracts the amount from the edit ✓            Use of Case/Selection statement ✓            Assigning exchange rate values to corresponding option selected ✓✓            Calculation of Rand Value : <i>rAmount * rRate</i> ✓            Display Rand Value Amount ✓ in Currency Format ✓</p>	7	
1.3	<p>Initialise counters ✓            Extract password from the edit ✓            Find length of the password ✓            Set up string scan loop ✓            Check if single character ✓ is in given list ✓ then increment counter1 ✓            Check if single character is a numeric digit ✓ then increment counter2 ✓            After loop check if password contains :                ten characters ✓ AND three or more special characters ✓ AND                three or more numeric digits ✓ then display confirmation message ✓            else                display message indicating password invalid ✓</p>	14	
1.4	<p><b>Follow algorithm given :</b>            Initialise all variables ✓            Start outer loop bound by the number of months ✓                Randomly generate the number ✓                Initialise the string variable ✓                    Start inner loop bound by the randomly generated value ✓                    Generate stars and create a new string ✓                    Close inner Loop                Display month number ✓, stars ✓ as well as how many stars ✓                Calculate Total Nights booked for the year ✓            Close outer loop            Display Total Nights ✓            Calculate Total Cost ✓            Display Total Cost ✓</p>	13	
	<b>SUB-TOTAL</b>	<b>41</b>	

## POSSIBLE SOLUTION (Q1):

---

1.1

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  pnlQu1_1.Caption := 'Welcome to Rainbow Tours'; ✓
  pnlQu1_1.Color := clYellow; ✓
  pnlQu1_1.Font.Style := [fsBold]; ✓
  pnlQu1_1.Font.size := 16; ✓
  ImgPic.Picture.LoadFromFile('Rainbow.jpg'); ✓
  ImgPic.Stretch := true; ✓
end;
```

(7)

1.2

```
procedure TForm1.btnQu1_2Click(Sender: TObject);
var
  rAmount, rRandValue, rRate: real;
begin
  rAmount := strToFloat(edtAmount.Text); ✓
  case cmbCurrency.ItemIndex of ✓
    0:  rRate := 16.46;
    1:  rRate := 21.16;
    2:  rRate := 19.27;
    3:  rRate := 0.22;
  } ✓✓
end;
rRandValue := rAmount * rRate; ✓
edtRandValue.Text := floatToStrF(rRandValue, ffCurrency, 6, 2); ✓
end;
```

(7)

1.3

```
procedure TForm1.btnQu1_3Click(Sender: TObject);
var
  iChar, k, iNum, iLen: integer;
  scode: string;
begin
  iChar := 0;
  iNum := 0;
  scode := edtPassword.Text; ✓
  iLen := length(scode); ✓
  for k := 1 to iLen do ✓
  begin
    if scode[k] in ['*', '?', '#', '%'] then ✓
      inc(iChar); ✓
    if (scode[k] in ['0' .. '9']) then ✓
      inc(iNum); ✓
  end;
  if (iLen = 10) ✓ and (iChar >= 3) ✓ and (iNum >= 3) ✓ then
    ShowMessage('Valid Password') ✓
```

```

else
  ShowMessage('Invalid Password'); ✓
end;

```

(14)

1.4

```

procedure TForm1.btnQu1_4Click(Sender: TObject);
var
  iRan, iNights, k, l: integer;
  sStars: string;
  rTotal: real;
begin
  // Given Code - Do NOT Alter !
  redDisplay.clear;
  redDisplay.Paragraph.tabcount := 1;
  redDisplay.Paragraph.tab[0] := 50;
  redDisplay.Lines.add('Month' + #9 + 'No of Nights');

  rTotal := 0;
  iNights := 0; } ✓
  for k := 1 to 12 do ✓
  begin
    sStars := ""; ✓
    iRan := Random(20)+1; ✓
    for l := 1 to iRan do ✓
    begin
      sStars := sStars + '*'; ✓
    end;
    redDisplay.Lines.add(inttostr(k) ✓ + #9 + sStars ✓ + inttostr(iRan)); ✓
    iNights := iNights + iRan; ✓
  end;
  rTotal := iNights * 450; ✓
  redDisplay.Lines.add('Total No of Nights booked: ' + inttostr(iNights)); ✓
  redDisplay.Lines.add('Total Income for the year: ' + floatToStrF(rTotal, ffCurrency, 7, 2)); ✓
end;

```

(14)

**TOTAL QUESTION 1 : 41**

## QUESTION 2

### 2.1 : SQL statements

#### 2.1.1

sSQL := 'Select GuestName, GuestSurname, DepositAmount ✓ From tblAccom ✓ Where DepositAmount between ✓ 2000 and 3000'; ✓ (4)

#### 2.1.2

sSQL := 'Select AccomName,AccomType,StarRating ✓ From tblRates ✓ Where len ✓ (StarRating) >=3'; ✓ (4)

#### 2.1.3

sSQL := 'Select Count(\*) ✓ as [Number of guests with NO deposit] ✓ from tblAccom Where Depositpaid = false ✓ and YEAR ✓ (DateIn) = 2019 ✓'; (5)

#### 2.1.4

sSQL := 'Select AccomName, SUM ✓ (DateOut-DateIn) ✓ As [No of days Booked] ✓ From tblAccom,tblRates ✓ Where tblAccom.AccomID = tblRates.AccomID ✓ Group by ✓ AccomName'; ✓ (7)

## ALTERNATIVE SOLUTIONS

#### 2.1.1

sSQL := 'Select GuestName, GuestSurname, DepositAmount ✓ From tblAccom ✓ Where DepositAmount >= 2000 ✓ and DepositAmount <= 3000'; ✓

## 2.2 Code Construct

QN	Description	Max	Learner
2.2.1	Sets ado-table to first record ✓ Loops through table ✓ Extracts first 2 digits of ID number(year of birth) ✓ Adds the century to the year of birth ✓ Use current year ✓ to calculate age ✓ Displays details with calculated age ✓ Moves to next record ✓ End loop	8	
2.2.2	Sets ado-table to first record Loops through table ✓ Sets ado-table to edit mode ✓ Finds the position of key words ("Country", "Guest") in accommodation name ✓ If "Country" ✓ found or "Guest" ✓ Found Update ✓ the rate field using appropriate formula ✓ Commit the transaction ✓ Move to next record ✓ End loop Display confirmation message ✓	10	

## POSSIBLE SOLUTION (Q2.2):

---

2.2.1

```
procedure TfrmQuestion2.btn2_2_1Click(Sender: TObject);
```

```
var
```

```
  sDriverTel: String;
```

```
  iAge,iYr : integer;
```

```
  sYear,sY : string;
```

```
begin
```

```
  //Provided code - DO NOT CHANGE
```

```
  redQ2.SelAttributes.Style := [fsBold];
```

```
  redQ2.Lines.Add('Name'+#9+'Surname'+#9+' Age');
```

```
  // QUESTION 2.2.1
```

```
  tblAccom.Open;
```

```
  tblAccom.First; ✓
```

```
  iYr := StrToInt(copy(IntToStr(CurrentYear),3));
```

```
  while not tblAccom.Eof do ✓
```

```
  begin
```

```
    sYear := copy(tblAccom['GuestIDNo'],1,2); ✓
```

```
    if StrToInt(sYear) > iYr then
```

```
      begin
```

```
        sYear := '19'+sYear; ✓
```

```
      end
```

```
    else
```

```
      begin
```

```
        sYear := '20'+sYear;
```

```
      end;
```

```
    iAge := CurrentYear✓-StrToInt(sYear); ✓
```

```
    redQ2.Lines.Add(tblAccom['GuestName']+#9+tblAccom['GuestSurname']+#9+  
                    IntToStr(iAge)); ✓
```

```
    tblAccom.Next; ✓
```

```
  end;
```

```
end;
```

(8)

```

2.2.2
procedure TfrmQuestion2.btn2_2_2Click(Sender: TObject);
var
iPos1, iPos2 : integer;
begin
// QUESTION 2.2.2
tblRates.First;
while not TblRates.Eof do ✓
begin
tblRates.Edit; ✓
iPos1 := Pos('Country',tblRates['AccomType'] ); } ✓
iPos2 := Pos('Guest',tblRates['AccomType']); }
if (iPos1 > 0) ✓ OR (iPos2 >0) ✓ then
begin
tblRates['Ratepernight'] ✓ := tblRates['Ratepernight']*1.05; ✓
end;
tblRates.Post; ✓
tblRates.Refresh;
tblRates.Next; ✓
end;
ShowMessage('Updated Completed !'); ✓

end;

```

(10)

**TOTAL QUESTION 2 : 38**

## QUESTION 3

### 3.1 Object Class

QN	Description	Max	Learner
3.1.1	Constructor header correct (including parameters) ✓ String attributes set correctly ✓ Integer attributes set correctly ✓ Calculate the Total Cost attribute ✓ using rate value passed Assigning Discount attribute to zero ✓	5	
3.1.2	Extract the month ✓ from the Date attribute ✓ Check if month ✓ is one ✓ of the designated months ✓ then assign a discount /percentage ✓ set Discount ✓ attribute to calculated value ✓ based on discount value/percentage ✓	9	
3.1.3	Procedure header correct (including parameter) ✓ Update Total Cost ✓ attribute by subtracting discount ✓	3	
3.1.4	Function header correct (including return type) ✓ Returns the Discount ✓ attribute	2	
3.1.5	Displays the correct captions ✓ With corresponding attribute values ✓ On separate lines (using token) Correct conversions for : Number of people attribute ✓ Discount attribute ✓ Total Cost attribute ✓	5	
3.2.1	Extracts tour date and number in group from components ✓ Extracts search destination from Combo Box ✓ Check if file exists ✓ then Links logical and physical files ✓ Opens file for reading ✓ Sets up loop ✓ Reads a line from textfile ✓ Locates the position of destination in the line ✓ If destination found then ( <i>using string handling</i> ) Isolates booking number ✓ Isolates destination ✓ Isolates rate ✓ End loop Else Error message / Close application ✓ Create method called correctly ✓ with correct parameters sent ✓ Confirmation message displayed ✓	15	
3.2.2	Method(s) called correctly ✓ and discount displayed ✓	2	
3.2.3	Method(s) called correctly ✓ and displayed ✓	2	
	<b>SUB-TOTAL</b>	<b>43</b>	



## POSSIBLE SOLUTION

### 3.1.1

constructor TTravel.create(sBooking, sDestination,sDate:string; iPeople: integer; rRate: real);✓

begin

```
fBooking:=sBooking;
fDestination:=sDestination; } ✓
fDate:=sDate;
fPeople:=iPeople; ✓
fTotal:=rRate*iPeople; ✓
fDiscount:=0; ✓
```

(5)

end;

### 3.1.2

procedure TTravel.calcDiscount;

var

iMonth:integer;

rDiscount:real;

begin

iMonth✓:=strToInt(copy(fDate,4,2)); ✓

if iMonth✓ IN✓ [2,5,8,11] ✓ then

begin

rDiscount:=5; ✓

end

else

rDiscount:=0;

fDiscount✓ := rDiscount/100✓ \* fTotal; ✓

end;

(9)

### 3.1.3

procedure TTravel.calcTotal; ✓

begin

fTotal✓ := fTotal-fDiscount; ✓

end;

(3)

### 3.1.4

function TTravel.getDiscount: real; ✓

begin

result := fDiscount; ✓

end;

(2)

### 3.1.5

```
function TTravel.toString: string;
```

```
var
```

```
sLine : string;
```

```
begin
```

```
sLine:= 'Booking No :'+#9+fBooking+#13 +'Destination :'+#9+ fDestination+ #13; ✓
```

```
sLine := sLine + 'Date of Booking :'+#9+fDate +#13+'No in Group :'+intToStr
```

```
(fPeople)+#13;
```

```
sLine := sLine+ 'Discount :'+floatToStrF(fDiscount,ffCurrency,8,2) ✓+#13+'Final Amount :'
```

```
+#9+floatToStrF(fTotal,ffCurrency,8,2); ✓
```

```
result := sLine;
```

```
end;
```

(5)

### 3.2.1

```
procedure TForm2.Button1Click(Sender: TObject);
```

```
var
```

```
sTourDate,sBookNo : string;
```

```
iNumG ,iPos: integer;
```

```
sDest ,sLine: string;
```

```
rRate : real;
```

```
tFile : textfile;
```

```
begin
```

```
sTourDate := edtTourDate.Text; } ✓
```

```
iNumG := sedPeople.Value; } ✓
```

```
sDest := cmbDest.Items[cmbDest.ItemIndex]; ✓
```

```
if FileExists('Destinations.txt') then ✓
```

```
begin
```

```
AssignFile(tFile,'Destinations.txt'); ✓
```

```
reset(tfile); ✓
```

```
while not eof(tfile) do ✓
```

```
begin
```

```
readln(tfile,sLine); ✓
```

```
iPos := pos(sDest,sLine); ✓
```

```
if iPos >0 then ✓
```

```
begin
```

```
iPos := pos(';',sLine);
```

```
sBookNo := copy(sLine,1,iPos-1); } ✓
```

```
delete(sLine,1,iPos);
```

```
iPos := pos(';',sLine);
```

```
sDest := copy(sLine,1,iPos-1); } ✓
```

```
delete(sLine,1,iPos);
```

```
rRate := StrToFloat(sLine); ✓
```

```
end;
```

```

end;
closefile(tfile);
end
else
begin
  ShowMessage('Error !');
  Application.Terminate; } ✓
end;
ObjTravel := TTravel.create(sBookNo,sDest,sTourDate,iNumG,rRate); ✓
ShowMessage('Booking Reservation Made'); ✓
end;

```

(15)

### 3.2.2

```

procedure TForm2.Button2Click(Sender: TObject);
begin
  travelObj.calcDiscount; ✓
  redOut.Lines.Add('Discount:'+FloatToStrF(travelObj.getDiscount ✓,ffCurrency,7,2));
end;

```

(2)

### 3.2.3

```

procedure TForm2.Button3Click(Sender: TObject);
begin
  redOut.Clear;
  travelObj.calcTotal; ✓
  redOut.Lines.Add(travelObj.toString); ✓
end;

```

(2)

**TOTAL QUESTION 3 : 43**

## QUESTION 4

QN	Description	Max	Learner
4.1.	<p><b>Creates a procedure display that:</b></p> <p>Displays correct header ✓            Sets up outer loop ✓                Initialise string to name of flight ✓                Sets up inner loop ✓                Builds row ✓                End inner loop            Displays row ✓            End outer loop</p> <p>Calls procedure ✓ in button</p>	7	
4.2	<p>Links logical file and physical file ✓            Creates a new file to read for the first time ✓            Set up outer loop                Sets up inner loop } ✓                Checks if array value less than 70 then ✓                    Builds string with name of flight and correct time (column) ✓                    Writes to text file ✓                Ends inner loop            Ends outer loop            Closes file ✓            Display confirmation message ✓            Code to load textfile in memo ✓</p>	9	
4.3	<p>Read the position (row) of option selected from combo-box ✓            Set up sort outer loop ✓                Set up sort inner loop ✓                Compare values ✓                Swop flight values ✓✓✓                Set up column loop ✓                    Swop rows of 2D array ✓✓✓                End column loop            End inner loop            End outer loop            Display message ✓</p>	12	
		13	

## POSSIBLE SOLUTION

4.1

```
procedure TForm2.display;
var
  r: Integer;
  c: Integer;
  str : string;
  I: Integer;
begin
  redDisplay.Clear;
  str := 'FLIGHTS'+#9;
  for I := 1 to 6 do
  begin
    str := str+IntToStr(i+11)+':00'+#9; ✓
  end;
  redDisplay.Lines.Add(str);
  redDisplay.Lines.Add('-----');
  for r := 1 to 10 do ✓
  begin
    str := arrFlights[r]+#9#9; ✓
    for c := 1 to 6 do ✓
    begin
      str := str + IntToStr(arrPass[r,c])+#9; ✓
    end;
    redDisplay.Lines.Add(str); ✓
  end;
end;
```

```
procedure TForm2.btnQ4_1Click(Sender: TObject);
begin
  display; ✓
end;
```

(7)

4.2

```
procedure TForm2.btnQ4_2Click(Sender: TObject);
var
  r: Integer;
  c: Integer;
  str : string;
  tFile : TextFile;
  sLine : string;
begin
  AssignFile(tFile,'CancFlights.txt'); ✓
  rewrite(tfile); ✓
  for r := 1 to 10 do
  begin
    for c := 1 to 6 do } ✓
  end;
```

```

begin
  if arrPass[r,c] < 70 then ✓
  begin
    sLine := (arrFlights[r]+#9+IntToStr(c+11)+' :00'); ✓
    writeln(tFile,sLine); ✓
  end;
end;
closeFile(tFile); ✓
showMessage('Contents written to File !'); ✓
MemOut.Lines.LoadFromFile('CancFlights.txt'); ✓
end;

```

(9)

4.3

```

procedure TForm2.btnQ4_3Click(Sender: TObject);
var
  iNum,i,j,k : integer;
  sTemp : string;
  iTemp : integer;
begin
  iNum := cmbFlight.ItemIndex+1; ✓
  for I := 1 to 9 do ✓
  begin
    for j := i+1 to 10 do ✓
    begin
      if arrPass[i,iNum]> arrPass[j,iNum] then ✓
      begin
        sTemp := arrFlights[i]; ✓
        arrFlights[i] := arrFlights[j]; ✓
        arrFlights[j] := sTemp; ✓

        for k := 1 to 6 do ✓
        begin
          iTemp := arrPass[i,k]; ✓
          arrPass[i,k] := arrPass[j,k]; ✓
          arrPass[j,k] := iTemp; ✓
        end;
      end;
    end;
  end;
  ShowMessage('Flight Details for '+cmbFlight.Text+' sorted according to capacity :'); ✓
  display;
end;

```

(12)

**TOTAL QUESTION 4 : 28**

**TOTAL = 150**