education

Department:
Education
PROVINCE OF KWAZULU-NATAL

GREENBURY SECONDARY SCHOOL

INFORMATION TECHNOLOGY P1

GRADE 11

JUNE EXAMINATIONS - 2017

MARKS: 150 TIME: 3 hours

EXAMINER(S) : MR N BRIJLAL *(Apollo Secondary)*

MODERATOR(S) : MR M GEBREAB *(Northwood Boys High)*

This question paper consists of 11 pages.

INSTRUCTIONS AND INFORMATION

1. This is a three-hour examination. Because of the nature of this examination it is important to note that you will NOT be permitted to leave the examination room before the end of the examination session.

2. You require the files listed below in order to answer the questions. They are EITHER on a stiffy disk OR CD issued to you, OR the invigilator/educator will tell you where to find them on the hard drive of the workstation you are using OR in which network folder it is. If the files are issued to you on a CD, you need to copy them onto your hard disk.

   QUESTION 1
   Customer_p.pas
   Customer_p.dfm
   Customer_p.dpr

   QUESTION 2
   delegates.txt
   strings_p.pas
   strings_p.dfm
   strings_p.dpr

   QUESTION 3
   Array_p.pas
   Array_p.dfm
   Array_p.dpr

3. Save your work at regular intervals as a precaution against power failures.

4. Read ALL the questions carefully. Do only what is required by the question.

5. During the examination you may use the manuals originally supplied with the hardware and software. You may also use the HELP functions of the software. You may NOT refer to any other resource material.

6. At the end of this examination session you will be required to hand in the stiffy or CD given to you by the invigilator with your work saved on it, or you must make sure that all your work has been saved on the network as explained to you by the invigilator/educator. Ensure that all files can be read.

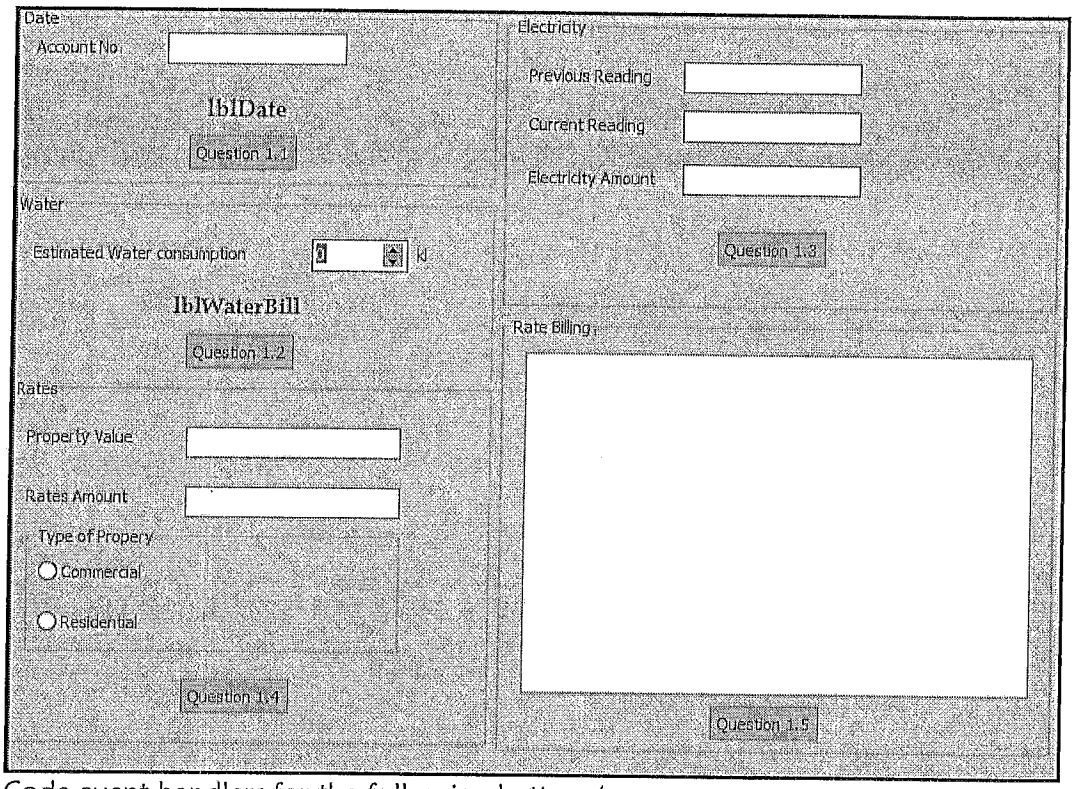
7. You also have to hand in printouts of the programming code for all the questions you have done.

8. All printing of programming questions will take place within an hour of the completion of the examination.

SCENARIO
The *Iselwini* municipality is one of the local municipalities in KZN. They are in the process of upgrading their existing software and have asked for your assistance in this regard..

QUESTION 1- GENERAL PROGRAMMING

The municipality would like to develop an application that could be used to calculate and view customer's utility bills.
You have been supplied with the following form in the project called *Customer_p*



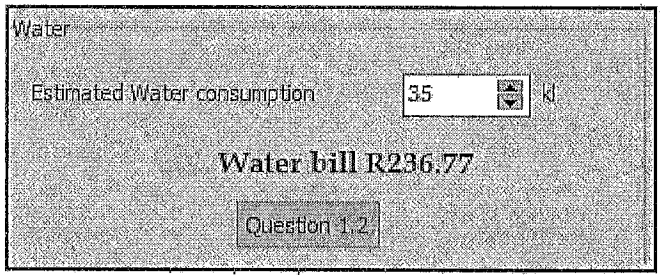
Code event handlers for the following buttons/components

1.1 **Button[Question 1.1]**
Write Delphi code to display the current/system date in the label *lblDate* in the format eg 24 May 2017 (2)

1.2 **Button [Question 1.2]**
The estimated water consumption in kilolitres(kl) for the consumer is currently displayed in the spin edit (*spnWaterCons*)
The water bill amount is determined in the following way :

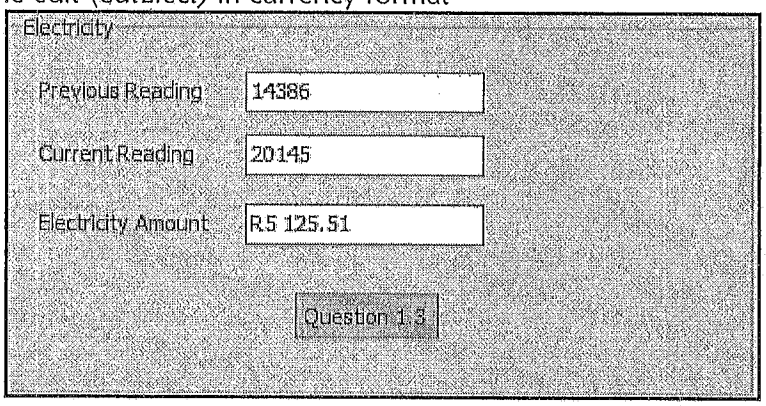
- The first 8kl is free (no charge)
- The next 22kl is charged at R7.56 per kl
- The remainder is charged at R14.09 per kl

Using the above criteria and the estimated water consumption value, determine the water bill amount. Display this value in the label (*lblWaterBill*) (5)
(See sample output on next page)



1.3 Button [Question 1.3]

- Randomly generate a previous meter reading in the range 10100 to 20200 and display this value in the edit (*edtPrev*)
- Retrieve the current meter reading from the edit (*edtCurr*)
- If the current meter reading is less than previous reading, then
    - clear the edit (*edtCurr*)
    - prompt the user, with a dialog box, to enter a valid value in the edit to proceed
- The number of kiloWatthours(kWh) consumed is represented by the difference in the meter readings while the electricity tariff is 89c per kWh.
  Calculate the electricity bill amount using the data supplied and display this value in the edit (*edtElect)* in currency format



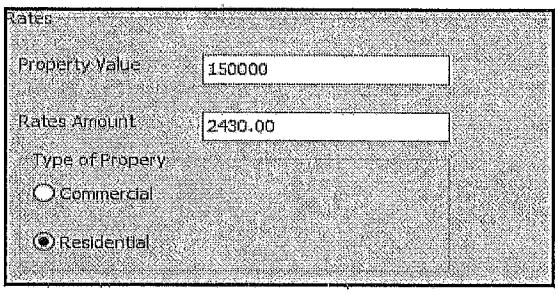
(8)

1.4 Button [Question 1.4]

The annual rates amount charged is based on the current value of the property and the type of property
The property can either be a commercial property or residential property
The following is applied in the calculation of the rates :

- No rates are applied to properties under R80 000
- The rates tariff is 1.8 cents per rand, based on the value of the property
- For residential properties, the first R15 000 is exempted from the rates calculation

Read in the property value from the edit (*edtPropVal*) and the type of property from the radio group, then calculate the rates amount due. Display this value in the edit (*edtRate*)



(12)

1.5 Button [Question 1.5]

The payment of the rates is usually done in the following manner :

- The first payment is calculated as 15% of the rates amount
- Thereafter this amount is deducted from the rates amount with the remaining balance to be paid in equal instalments over 11 months

Retrieve the account number and rates value, from the respective edits.
Display the account number at the top , then :
Calculate and display the month, the amount to be paid monthly and the decreasing balance in neat columns with suitable headings. The amounts must be displayed in currency format, correct to two decimal places

Account No : 7452187

Month	Monthly payment	Balance
1	R364.50	R2 065.50
2	R187.77	R1 877.73
3	R187.77	R1 689.95
4	R187.77	R1 502.18
5	R187.77	R1 314.41
6	R187.77	R1 126.64
7	R187.77	R938.86
8	R187.77	R751.09
9	R187.77	R563.32
10	R187.77	R375.55
11	R187.77	R187.77
12	R187.77	R0.00

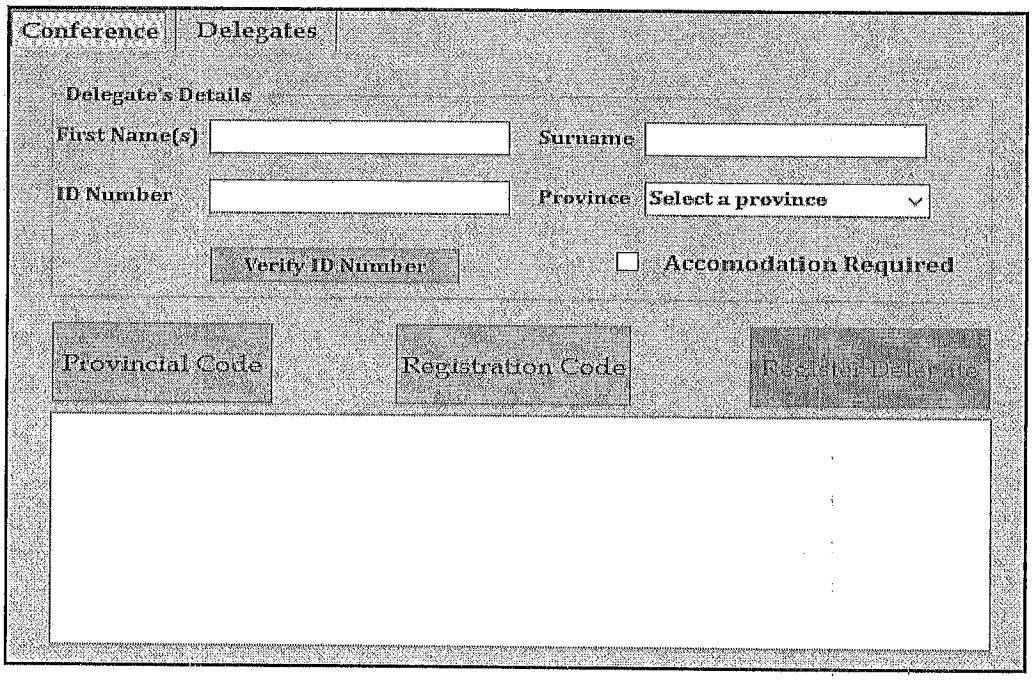
(10)

SUB TOTAL 37

QUESTION 2- STRING HANDLING & TEXT FILES

2.1 Tabsheet : Conference

The municipality will be holding a national conference at the Durban ICC in July, where delegates at the conference will be representing their provinces.
Details of delegates attending the conference, will be captured using the following form :



2.1.1 Button[Verify ID Number]

Validate the ID number in the edit (*edtID*), as only a 13 digit ID number will be acceptable. If an ID number does not meet this requirement then, the edit should be cleared and the user should be continuously prompted(via a dialog box) to enter their ID number, until a valid ID number is entered.
Display the valid ID number in the edit box *edtID* (5)

2.1.2 Button[Provincial Code]

Declare a global variable *provCode* of type string. Retrieve the province selected by the user from the combo-box, and use it to determine the provincial code of the delegate. The provincial code is derived as follows :

- If the province consists of a single word, then the code is the first three letters of the province in uppercase eg. Gauteng = GAU
- If the province consists of two words, the code becomes the first letter of each word in the name in uppercase eg. Kwazulu Natal = KN

Store the code in the variable *provCode* (6)

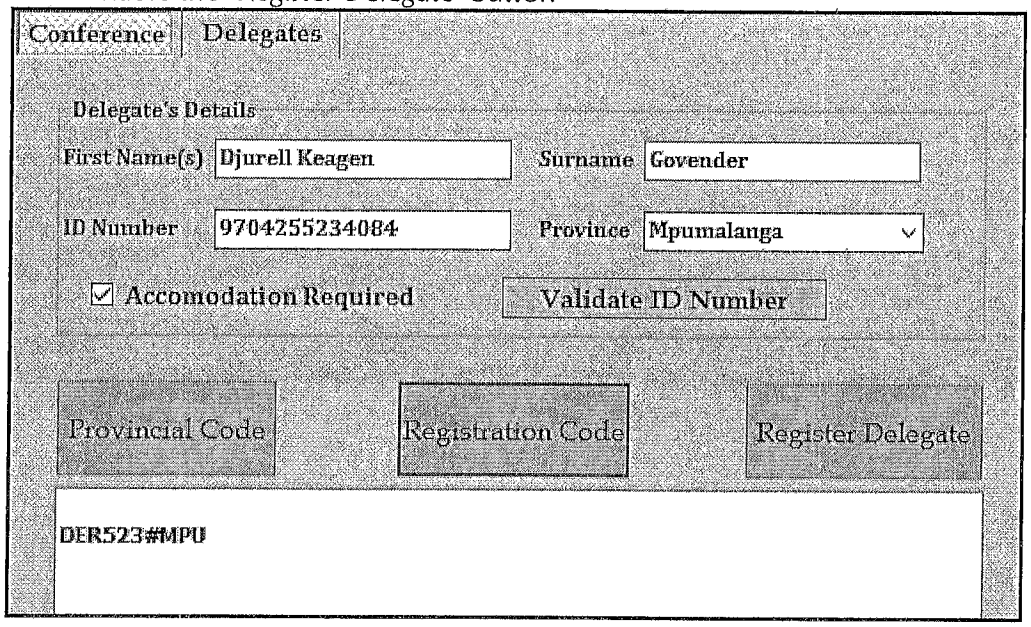
2.1.3 Button[Registration Code]

Each delegate will be issued with a registration code, which is derived as follows :

- First three digits are the last three letters of surname
- Next three digits are the 7th , 8th and 9th digits of the ID number
- Next character is a hyphen (-) if the delegate is not requiring accommodation or a hash(#) if they are requiring accommodation
- Followed by the provincial code

NOTE :

- Registration code must be in uppercase !
- Store the registration code in a global variable *RegCode*
- Enable the 'Register Delegate' button



(8)

2.1.4 Button[Register Delegate]

This button will display the details of the delegate in the rich edit as confirmation of his registration for the conference. Display the following details :

Initial & Surname : Eg RD Pillay
Province : Eg Gauteng
Accomodation Required : Yes/ No
Registration Code : Eg LAY453-GAU

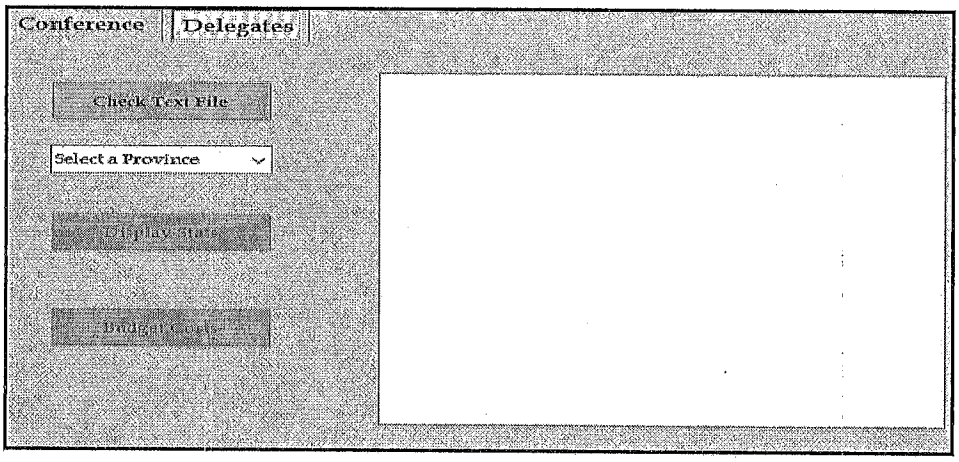
(8)

2.2 Tabsheet : Delegates

You are supplied with a text file called *delegates.txt* which contains details of registered delegates attending the conference. Each line of this delimited text file contains details of the delegate in the following format :
Initial & Surname@Province@Y or N (indicating if accommodation is required)

Eg TR Bantho@Western Cape@N
T Harridas@Gauteng@Y

Code event handlers for the following :
Note that the buttons : '*Display Stats*' and '*Budget Costs*' are disabled



2.2.1 Button[Check Text File]

Check that the text-file '*delegates.txt*' exists, then using dialogue boxes :

- display a confirmation message and enable the remaining buttons, if the text file is located

OR

- display an appropriate error message in the event that the text file cannot be found (4)

2.2.2 Button[Display Stats]

Retrieve the selected province from the combo-box
Read the text-file and separate and display the details of the text-file in the rich edit in tabular form with suitable headings
In addition determine and display the :

a) total number of delegates for the conference
b) number of delegates requiring accommodation
c) number of delegates from the selected province

HINT : *Use global variables !*

NAME	PROVINCE	ACCOMODATION
TR Bantho	Western Cape	N
T Harridas	Gauteng	Y
L Mariah	Gauteng	Y
S Govender	Kwazulu Natal	N
T Msobe	Limpopo	Y
BT Mtshali	Mpumalanga	Y
F Shaik	Gauteng	Y
E Doraivelu	Free State	Y
RD Pillay	Kwazulu Natal	N
MJ Rambali	Western Cape	Y
JG Pine	North West	N
M Marimuthu	Limpopo	N
M Abdullah	Western Cape	Y
A Hiralal	North West	N
D Chetty	Free State	Y
A Mohamed	North West	N
L Pillay	Mpumalanga	Y
T Myrtyle	Northern Cape	Y
GB Smith	Kwazulu Natal	Y
RE Pillay	Gauteng	N
KM Vasinathan	North West	Y

Number of delegates 22
Number requiring accomodation 12
Number from Kwazulu Natal is 3

(12)

2.2.3 Button[Budget Costs]

The municipality will spend the following costs towards the hosting of the delegates for this conference :

Aspect	Cost per delegate
Subsistence cost per delegate	R135.00
Accommodation cost	R560.00

The total allocated budget for this is R400 000. Determine and display the following:

a) the total subsistence cost
b) the accommodation cost
c) total cost for hosting the delegates

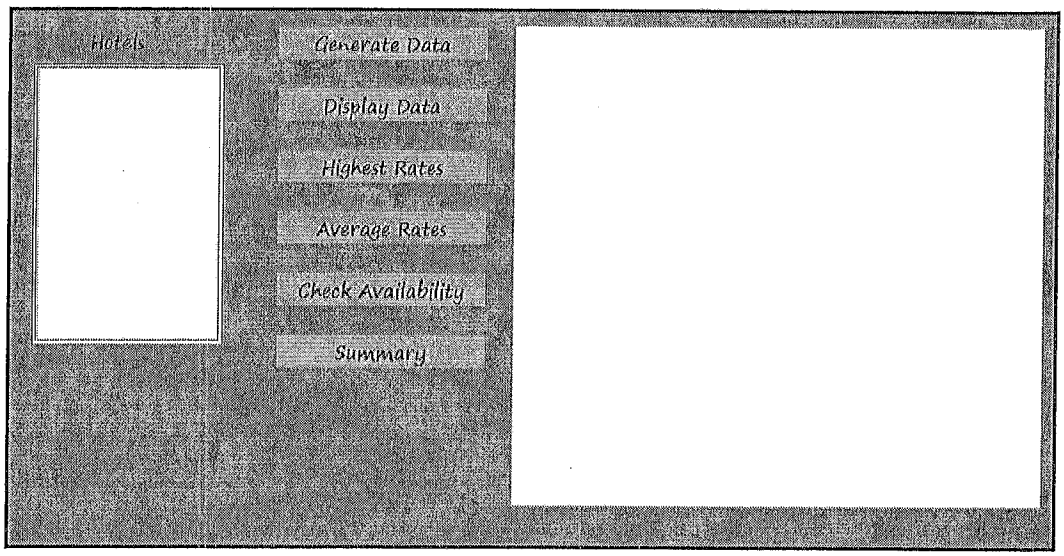
If the total cost exceeds the allocated budget then activate the hidden panel, *pnlBudget* (7)

SUB TOTAL 50

QUESTION 3 – ARRAYS

3.1 The admin manager needs to make reservations and bookings for the delegates who will be requiring accommodation. Each delegate will only be accommodated for 2 nights at any local hotel. The names of 8 local hotels and the rates they charge per night are stored in two arrays. An application has to be developed to control bookings at the hotels.

The form has the following layout :



3.1.1 Declare a one-dimensional array called *arrRooms* to store the number of rooms available in each hotel. (2)

3.1.2 Declare two global variables called ***nBookings****(counts the number of bookings to be made)*, and ***total****(totals the amount paid to hotels for the bookings)* (2)

3.1.3 Create an onActivate event for the form to display the hotels (from the given array *arrHotels*) in the list box.
You should also initialise the variables *nBookings* and *total* (3)

3.2 Now code the following as instructed :

3.2.1 Button[Generate Data]
Randomly generate values in the range 0 to 10 and store in the array *arrRooms*.
These values represent the number of rooms available for booking per hotel !
Display a message in a dialog box, indicating that the values have been generated (5)

3.2.2 Button [Display Data]
Display the data from the arrays in tabular format in the rich-edit, with suitable headings as shown below. Display an additional message indicating that the hotel is fully booked , alongside any hotel that has no rooms available,

HOTEL	ROOMS	RATE PER NIGHT	
Parkly Hotel	8	R534.65	
Moses Gardens	7	R456.87	
The Selby	4	R575.23	
The Maharaja	4	R643.12	
Hotel Bartlet	9	R654.92	
The Delphine	0	R562.00	Fully booked!
Holiday Inn	8	R765.92	
Protea Sands	9	R438.43	

(5)

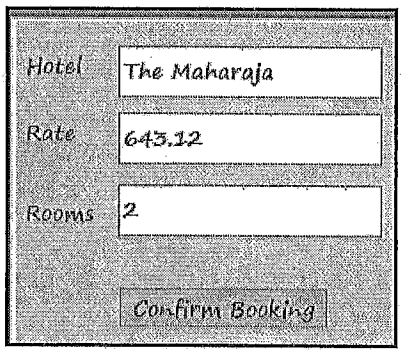
3.2.3 Button[Highest Rates]
Determine and display in the rich edit, the hotel together with the rates amount, of the hotel with the highest rates.

Highest Rate is R765.92 charged by Holiday Inn

(6)

3.2.4 Function avgRates
Code a function called *avgRates*, that will calculate and return the average rates amount charged by the hotels
Button[Average rates]
Display the average rates amount in the rich-edit , with a suitable message, by making a call to the function *avgRates* (7)

3.2.5 Button[Check Availability]
This must allow the user to select a hotel from the list box, input the number of rooms to book (via a dialogue box), and determine if there are rooms available in the selected hotel to accommodate the booking request. Display a message accordingly in the rich edit.
If the booking can be made, also enable the panel '*pnlBooking*', which must reflect the name of the selected hotel, the number of rooms to be booked as well as the rate charged by that hotel



(10)

3.2.6 Button[Confirm Booking]

Write code that will simulate the reservation of the booking at the requested hotel. Do the following :

- update the number of rooms available (*arrRooms*), for the selected hotel.
- keep a tally of the total number of rooms being booked (use global variable declared)
- determine an accumulated total of the amount being paid for the bookings
- display a confirmation message in a dialogue box, indicating that the reservation has been made
- hide the panel *pnlBooking*

Note : *this process can then continue, so as to book all delegates into hotels, and thereby obtain the totals for number of delgates/rooms being booked !* (6)

3.2.7 Button[Summary]

Display the details of the bookings as follow :

- The total number of rooms booked
- The total cost of the booking reservations (2)

SUB TOTAL 48

TOTAL = (135 *10) / 9 = 150

HOD - MCM/JC
06/06/2017

JUNE EXAMINATIONS-2017

INFORMATION TECHNOLOGY : GRADE 11 – P1

MARKING MEMO

QUESTION 1

1.1

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  lblDate.Caption:=FormatDateTime✓('d mmmm yyyy',date); ✓
end;
```

2

1.2

```
procedure TForm1.Button4Click(Sender: TObject);
var
wRead : integer;
wChrg : real;
begin
  wRead := spnWaterCons.Value;
  if wRead >=30 then✓
  begin
    wChrg := 22*7.56+(wRead-30)*14.09; ✓
  end;
  if (wRead >8 ) and (wRead <=22) then✓
  begin
    wChrg := (wRead-8)*7.56; ✓
  end;
  if wRead<=8 then
  begin
    wChrg := 0;
  end;
  lblWaterBill.Caption:= 'Water bill'+ FloatToStrF
                         (wChrg,  ffCurrency,7,2);   ✓
end;
```

5

1.3

```
procedure TForm1.Button1Click(Sender: TObject);
var
pR, cR,kwh : integer;
eBill : real;
begin
  pR := RandomRange(10100,20201);      ✓
  edtPrev.Text:= IntToStr(pR);
  cR := StrToInt(edtCurr.Text);                ✓
  if cR < pR then ✓
  begin
    edtCurr.Clear;  ✓
    cR := StrToInt(InputBox('Data','Enter a valid amount',''));  ✓
  end;
   kwh := cR - pR; ✓
   eBill := kwh*0.89; ✓
   edtElect.Text:= FloatToStrF(eBill,ffCurrency,7,2); ✓

end;
```

8

1.4

```
procedure TForm1.Button2Click(Sender: TObject);
var
pVal,rates : real;
pType : string;
begin
 pVal := StrToFloat(edtPropVal.Text); ✓

 if (rgpProp.ItemIndex=0) ✓and (pVal >= 80000) ✓then
 begin
   rates := 0.018*pVal; ✓✓
 end
 else
   if (rgpProp.ItemIndex=1) ✓and (pVal>=80000) ✓then
   begin
     rates := (pVal-15000) ✓*0.018; ✓
   end
   else                                        // nested if…else ✓
   begin
     rates := 0; ✓
   end;
   edtRate.Text := FloatToStrf(rates,ffFixed,7,2); ✓
end;
```

12

1.5

```
procedure TForm1.Button3Click(Sender: TObject);
var
rates,bal,pay,monPay : real;
  I: Integer;
begin
  rates := StrToFloat(edtRate.Text);
  pay := rates*0.15;  ✓
  bal := rates-pay; ✓
  monPay := bal/11; ✓
  redDisplay.Paragraph.TabCount :=3;
  redDisplay.Paragraph.Tab[0] := 50;
  redDisplay.Paragraph.Tab[1]:=100;          ✓
  redDisplay.Paragraph.Tab[2] :=160;
  redDisplay.Lines.Add('Account number '+edtAcc.Text);
  redDisplay.Lines.Add('Month'+#9+'Monthly payment'+#9+'Balance');
✓

redDisplay.Lines.add('1'+#9+FloatToStrf(pay,ffCurrency,7,2)+#9#9+
     FloatToStrF(bal,ffCurrency,7,2)); ✓
  for I := 2 to 12 do ✓
  begin
    bal := bal-monPay; ✓
    redDisplay.Lines.Add(IntToStr(i)+#9+
                FloatToStrF(monPay,ffCurrency,7,2)+#9#9+
                FloatToStrF(bal,ffCurrency,7,2)); ✓✓*
  end;
end;
```

** Currency Format*

10

SUB TOTAL QUESTION 1 — 37

QUESTION 2

2.1.1 — 5

```
procedure TForm1.Button7Click(Sender: TObject);
var
id :string;
begin
   id := edtID.Text; ✓
  while length(id)<> 13 do ✓
  begin
    edtID.Clear; ✓
    id := InputBox('Data','Enter valid ID number',''); ✓
  end;
  edtID.Text:= id; ✓
end;
```

2.1.2 — 6

```
procedure TForm1.Button4Click(Sender: TObject);
var
prov :string;
posSpc : integer;
begin
 prov:= cmbProv.text; ✓
 posSpc := pos(' ',prov); ✓
 if posSpc=0 then ✓
 begin
   provCode := UpperCase✓ (copy(prov,1,3) ✓);
 end
 else
 begin
   provCode := uppercase(prov[1]+prov[posSpc+1]); ✓
 end;
end;
```

2.1.3 — 8

```
procedure TForm1.Button5Click(Sender: TObject);
var
surN ,id,last3 : string;
begin
  surN := edtSN.Text;
  id := edtID.Text;  ✓
  last3 := copy(surN,length(surn)-2); ✓✓
  regCode := last3+id[7]+id[8]+id[9]; ✓
 if chbAcc.Checked then ✓
  begin
    regCode := uppercase(regCode+'#'+provCode); ✓
  end
  else
  begin
    regCode := uppercase✓ (regCode+'-'+provCode); ✓
  end;
  redDisplay.Lines.add(regCode);
  btnRegCand.Enabled := true;
end;
```

2.1.4 — 8

```
procedure TForm1.btnRegCandClick(Sender: TObject);
var
init,firstN : string;
  I: Integer;
begin
  firstN := edtFN.Text;
  if pos(' ',firstN)=0 then
  begin
    init :=uppercase(firstN[1])+' '+edtSN.Text; ✓
  end
  else
  begin
    init := firstN[1];
    for I := 2 to length(firstN)do ✓
    begin
      if firstN[i]=' ' then ✓
      begin
         init := uppercase(init+firstN[i+1]); ✓
      end;
    end;
     init := init+' '+edtSN.Text; ✓
  end;
  redDisplay.Lines.Add('Initials & Surname : '+init);
  redDisplay.Lines.Add('Province :'+cmbProv.Text); ✓
  if chbAcc.Checked then
  begin
    redDisplay.Lines.Add('Accomodation required : Yes');
  end
  else
  begin
    redDisplay.Lines.Add('Accomodation required : No');   ✓✓
  end;
  redDisplay.Lines.Add('Registration Code :'+regCode);
end;
```

2.2.1 — 4

```
procedure TForm1.Button1Click(Sender: TObject);
var
tfile : textfile;
begin
  if FileExists('delegates.txt') ✓ then
  begin
    ShowMessage('File located !'); ✓
    btnStats.Enabled:=true;
    btnBudget.Enabled := true;          ✓
  end
  else
  begin
     ShowMessage('File cannot be located'); ✓
  end;
end;
```

2.2.2

```
procedure TForm1.btnStatsClick(Sender: TObject);
var
tfile :textfile;
posAt :integer;
prov,sProv,oneL,naam : string;
```

```
begin
  nDel :=0;
  nAcc := 0;
  nProv :=0;                    ✓
  redOutput.Paragraph.TabCount := 3;
  redOutput.Paragraph.Tab[0]:=80;
  redOutput.Paragraph.Tab[1]:=150;
  redOutput.Paragraph.Tab[2] := 70;
  redOutput.Lines.Add('NAME'+#9+'PROVINCE'+#9+'ACCOMODATION');
  sProv:= cmboProv.Items[cmboProv.ItemIndex]; ✓
  AssignFile(tfile,'delegates.txt'); ✓
  reset(tfile); ✓
  while not eof(tfile) do✓
  begin
    Readln(tfile,onel); ✓
    Inc(nDel);
    posAt := pos('@',onel);
    naam := copy(onel,1,posAt-1);  ✓
    delete(onel,1,posAt); ✓
    posAt := pos('@',onel);
    prov := copy(onel,1,posAt-1);  ✓
    if prov = sProv then
    begin
      inc(nProv);                  ✓
    end;
    delete(onel,1,posAt);
    if onel='Y' then
    begin
      inc(nAcc);                   ✓
    end;
    redOutput.Lines.Add(naam+#9+prov+#9+onel); ✓
  end;
  CloseFile(tfile);
  redOutput.Lines.add('Number of delegates '+IntToStr(nDel));
  redOutput.Lines.Add('Number requiring accomodation
'+IntToStr(nAcc));
  redOutput.Lines.Add('Number from '+sprov+' is
'+IntToStr(nProv));
end;
```

12

2.2.3

```
procedure TForm1.btnBudgetClick(Sender: TObject);
var
subCost,accCost,totCost : real;
begin
 redOutput.Clear;
 subCost := nDel*135; ✓
 accCost := nAcc*560; ✓
 totCost := subCost+accCost; ✓
 redOutput.Lines.Add('The subsistence cost is
'+FloatToStrF(subCost,ffCurrency,7,2)) ;
 redOutput.Lines.Add('The accomodation cost is
'+FloatToStrF(accCost,ffCurrency,7,2));
 redOutput.Lines.Add('Total cost is                ✓✓
'+FloatToStrF(totCost,ffCurrency,7,2));
```

7

```
 if totCost > 400000 then✓
 begin
    pnlBudget.Visible := true; ✓
 end;

end;
```

SUB TOTAL QUESTION 2 — 50

QUESTION 3

3.1.1

```
arrRooms✓ : array[1..8] of integer; ✓
```

2

3.1.2

```
nBookings : integer; ✓
total : real; ✓
```

2

3.1.3

```
procedure TForm1.FormActivate(Sender: TObject);
var
  I: Integer;
begin
nBookings := 0;
total :=0;                      ✓
for I := 1 to 8 do ✓
begin
  lst1.Items.Add(arrHotel[i]); ✓
end;
end;
```

2

3.2.1

```
procedure TForm1.Button1Click(Sender: TObject);
var
  I: Integer;
begin
  for I := 1 to 8 do ✓
  begin
    arrRooms[i] ✓:= RandomRange✓ (0,11); ✓
  end;
  ShowMessage('Values generated'); ✓
end;
```

5

3.2.2

```
procedure TForm1.Button2Click(Sender: TObject);
var
  x: Integer;
begin
  redDisplay.Paragraph.TabCount :=3;
  redDisplay.Paragraph.Tab[0] :=100;
  redDisplay.Paragraph.Tab[1] :=80;
  redDisplay.Paragraph.Tab[2] :=150;
  redDisplay.Lines.Add('HOTEL'+#9+'ROOMS'+#9+'RATE PER NIGHT'); ✓
  for x := 1 to 8 do✓
  begin
  if arrRooms[x]=0 then ✓
  begin
    redDisplay.Lines.Add(arrHotel[x]+#9+IntToStr(arrRooms[x])+#9+
   FloatToStrF(arrRates[x],ffCurrency,7,2)+#9+'Fully booked!'); ✓
  end
  else
    begin
```

5

	redDisplay.Lines.Add(arrHotel[x]+#9+IntToStr(arrRooms[x])+#9+ FloatToStrF(arrRates[x],ffCurrency,7,2)); ✓ end; end; end;	

3.2.3	procedure TForm1.Button4Click(Sender: TObject);
	var
	maxR : real;
	maxH : string;
	I: Integer;
	begin
	maxR := 0; ✓
	for I := 1 to 8 do✓
	begin
	if arrRates[i]>maxR ✓then
	begin
	maxR := arrRates[i]; ✓
	maxH := arrHotel[i]; ✓
	end;
	end;
	redDisplay.Lines.add('Highest Rate is
	'+FloatToStrF(maxR,ffCurrency,7,2)+' charged by '+maxH); ✓
	end;
		6

3.2.4	function TForm1.avgRates: real; ✓
	var
	sum : real;
	I: Integer;
	begin
	sum :=0; ✓
	for I := 1 to 8 do✓
	begin
	sum := sum+arrRates[i]; ✓
	end;
	result ✓:= sum/8; ✓
	end;
	procedure TForm1.Button5Click(Sender: TObject);
	begin
	redDisplay.Lines.Add('The average rate'+ FloatToStrF
	(avgRates,ffCurrency ,7,2));✓
	end;
		7

3.2.5	procedure TForm1.Button3Click(Sender: TObject);
	var
	hotel : string;
	avail : boolean;
	rate : real;
	nRoom : integer;
	I: Integer;
	begin
	avail := false;
	hotel := lst1.Items[lst1.ItemIndex]; ✓
	nRoom := StrToInt(InputBox('Data','Enter number rooms to
	book','1')); ✓
	for I := 1 to 8 do ✓
	begin
		10

	if (hotel=arrHotel[i]) ✓ and (arrRooms[i]>=nRoom) ✓ then
	begin
	avail := true; ✓
	redDisplay.Lines.Add('Booking can be made !');
	pnlBooking.Visible := true; ✓
	edtHotel.Text := hotel;
	edtRate.Text := FloatToStr(arrRates[i]);
	edtRoom.Text:= IntToStr(nRoom); ✓
	end;
	end;
	if avail = false then ✓
	begin
	redDisplay.Lines.Add('Insufficient rooms !'); ✓
	end;
	end;	

3.2.6	procedure TForm1.Button7Click(Sender: TObject);
	var
	nRm,iPos : integer;
	nRt : real;
	
	begin
	nRm := StrToInt(edtRoom.Text);
	nRt := StrToFloat(edtRate.text); ✓
	iPos:= lst1.ItemIndex+1; ✓
	arrRooms[iPos] := arrRooms[iPos]-nRm; ✓
	nBookings := nBookings+nRm; ✓
	total := total+(nRm*nRt*2); ✓✓
	ShowMessage('Reservation has been made !');
	pnlBooking.Visible:= false;
	end;
		6

3.2.7	procedure TForm1.Button6Click(Sender: TObject);
	begin
	redDisplay.Lines.Add('Total bookings '+IntToStr(nBookings)); ✓
	redDisplay.Lines.Add('Total cost
	'+FloatToStrF(total,ffCurrency,7,2)); ✓
	end;
		2

	SUB TOTAL QUESTION 3	48