***GRADE 10 PAPER 1***

***NOVEMBER 2018***

***MEMO***

***Question 1***

|  |  |  |
| --- | --- | --- |
|  | Question Mark Allocation | Mark |
| 1.1 |  |  |
| **a** | **Read in name and surname via dialog box** | (1) |
| **b** | **Display Message** | (1) |
| **c** | **Changing the font data:**  **Size =12**  **Font name Times New Roman**  **Font style =fsbold**  **Font Colour Red** | (1)  (1)  (1)  (1) |
| **d** | **Label width set to 209** | (1) |
| **e** | **Word wrap set to true** | (1) |
|  | procedure TfrmGUI.Button1Click(Sender: TObject);  begin  //1.1  sName:=InputBox('Personal Details','Enter in your Name','Jane Doe'); ✓  lblWelcomeMsg.Caption:='Welcome to Books 2018 '+sName; ✓  lblWelcomeMsg.Font.Size:=12; ✓  lblWelcomeMsg.Font.name:='Times New Roman' ; ✓  lblWelcomeMsg.Font.Style:=[fsBold]; ✓  lblWelcomeMsg.font.Color:=clRed; ✓  lblWelcomeMsg.Width:=209; ✓  lblWelcomeMsg.WordWrap:=true; ✓  end; | **[8]** |
| 1.2 |  |  |
|  | **Using case statement or if statement to change picture**  **Each if and image picture set is 2 marks (4x2)**  **Set picture to stretch** | (8)  (1) |
|  | procedure TfrmGUI.rdgRoomsClick(Sender: TObject);  begin  //1.2  case rdgRooms.ItemIndex of✓  0:imgName.Picture.LoadFromFile('double.jpg'); ✓✓  1:imgName.Picture.LoadFromFile('Triple.jpg'); ✓✓  2:imgName.Picture.LoadFromFile('Quad.jpg'); ✓✓  3:imgName.Picture.LoadFromFile('Queen.jpg'); ✓  end;  imgName.Stretch:=true; ✓  end; | **[9]** |
| **1.3** |  |  |
|  | **Add cash to combo box using code** | (1) |
|  | **Add card to combo box using code** | (1) |
|  | procedure TfrmGUI.FormCreate(Sender: TObject);  begin  //1.3  cbxPayment.Items.Add('Cash'); ✓  cbxPayment.Items.add('Card'); ✓  end; | **[2]** |
| 1.4 |  |  |
|  | **Display message using message dialog** |  |
|  | procedure TfrmGUI.btnPaymentClick(Sender: TObject);  begin  //1.4  ShowMessage✓ (sName✓+' is booking a '+rdgRooms.Items[rdgRooms.ItemIndex] ✓✓+' Room and is paying by '+cbxPayment.Text✓);  end;  suitable message✓ | **[6]** |
|  | QUESTION 1 TOTAL: | **[25]** |

***Question 2***

|  |  |  |
| --- | --- | --- |
|  | Question Mark Allocation | **Mark** |
| 2.1 |  |  |
| **a** | **Read in name, surname, email** | (3) |
| **b** | **Determine if the fields are blank** | (3) |
|  | **Display suitable message** | (2) |
| **c** | **Enable the 2 panels if user is registered** | (2) |
|  | procedure TfrmBooking.btnRegisterClick(Sender: TObject);  begin  sname := edtName.Text; ✓  ssurname := edtSurname.Text; ✓  semail := edtEmail.Text; ✓  if (sname = '') ✓ or (ssurname = '') ✓or (semail = '') ✓then  begin  lblRegister.Caption := 'Registration Failed!'; ✓  ShowMessage('Complete all fields correctly!');  end  else  begin  lblRegister.Caption := 'Registration Successful'; ✓  pnlSelectRoom.Enabled := true; ✓  pnlSelectBooking.Enabled := true; ✓  end;  end; | **[10]** |
| 2.2 |  |  |
|  | **Display cost depending on radio button selected** |  |
|  | procedure TfrmBooking.rgpRoomsClick(Sender: TObject);  begin    case rgpRooms.ItemIndex of✓  0: rRoomPrice:=600;  1: rRoomPrice:=750;  2: rRoomPrice:=1100; ✓✓  3: rRoomPrice:=1600;  end;  lblCostPerRoom.Caption := 'R '+floatToStr (rRoomPrice,,ffcurrency,8,2); ✓  end; | **[4]** |
|  |  |  |
| 2.3 | **Retrieve room selected and number of people staying**  **Determine how many rooms needed , rounded up.** | (2)  (6) |
|  | procedure TfrmBooking.btnRoomsRequiredClick(Sender: TObject);  var  iAdults:integer;  begin  iAdults := sedtAdults.Value; ✓  case rgpRooms.ItemIndex ✓of✓  0:  rNumberOfRooms:= Ceil(iAdults / 2); ✓  1:  rNumberOfRooms:= Ceil(iAdults / 3); ✓  2:  rNumberOfRooms:= Ceil(iAdults / 4); ✓  3:  rNumberOfRooms:= Ceil(iAdults / 2); ✓  end;  lblNumberOfRooms.Caption:=’Rooms Needed is ‘+IntToStr(rNumberOfRooms); ✓  **NOTE: *number of rooms is named rNumberOfRooms but is an integer data type.***  End; | **[8]** |
|  |  |  |
| 2.4 |  |  |
|  | **Determine days staying** | (1) |
|  | **Calculate total cost** | (3) |
|  | **Display as currency** | (3) |
|  | procedure TfrmBooking.btnTotalCostClick(Sender: Tobject);  var  idays:integer;  begin  //2.4  idays:=sedNumberDays.Value; ✓  rTotalCost:= rRoomPrice✓\* rNumberOfRooms ✓\*idays✓;  lblTotalCost.Caption✓:=FloatToStrF✓ (rTotalCost,ffCurrency,8,2✓);  end; | **[7]** |
| 2.5 |  |  |
|  | Display data in correct format |  |
|  | procedure TfrmBooking.btnConfirmBookingClick(Sender: Tobject);  begin  //2.5 code here  redtDisplay.Clear;  redtDisplay.Lines.Add(edtName.Text+’ ‘+sName); ; ✓  redtDisplay.Lines.Add(‘Email: ‘+sSurname); ; ✓  redtDisplay.Lines.Add(‘Type of Room: ‘+sRoomType; ✓  redtDisplay.Lines.Add(‘Rooms needed’+floattostr( rNumberOfRooms)) ; ✓  redtDisplay.Lines.Add(‘Total cost: ‘+ floattostr( rTotalCost)) ; ✓  end; | **[5]** |
| 2.6 |  |  |
|  | Clear all fields and set the panel enable false |  |
|  | procedure TfrmBooking.btnNewBookingClick(Sender: Tobject);  begin  //2.6 code here  edtName.Clear;  edtSurname.Clear; ✓✓  edtEmail.Clear;  redtDisplay.Clear;  pnlSelectBooking.Enabled:=false;  pnlSelectRoom.Enabled:=false; ✓✓  sedtAdults.Value:=0;  lblTotalCost.Caption:=’’;  sedNumberDays.Value:=0;  rgpRooms.ItemIndex:=-1; ✓✓  lblRegister.Caption:=’’;  end; | **[6]** |
|  | **QUESITON 2 TOTAL:** | **[40]** |

***QUESTION 3***

|  |  |  |
| --- | --- | --- |
| 3.1 | **Retrieve the hours per day and number of emails sent.**  **To determine how much data the guest would need the following formula’s must be used.( The data below represents MB)**   * **Internet browsing data= Number of hours browsing \* 450** * **Email data= (number of emails/25) \* 2048** * **Media data = number of hours on media sites \* 1024** * **Stream data= number of hours streaming \* 5120**   **Display the number of GB the user would need, rounded up.**  **Calculate Cost (GB \* 49,99)**  **Display the cost of the data in currency format and enable the button Display Discount** | (4)  (5)  (2)  (1)  (1) |
|  | procedure TForm1.Button1Click(Sender: TObject);  var  ibrows, iemail, imedia, istream, iGB, iMB: integer;  rcost, rdata: real;  begin  redOutput.clear;  ibrows := sedInternetHours.Value; ✓  iemail := sedEmail.Value; ✓  imedia := sedMedia.Value; ✓  istream := sedStream.Value; ✓  rdata := (ibrows \* 450) ✓+ (iemail / 25✓ \* 2048) ✓+ (imedia \* 1024) ✓+  (istream \* 5120) ✓;  iGB := Ceil(rdata / 1024); ✓  redOutput.Lines.Add(intToStr(iGB) + ' GB needed'); ✓  rcost := iGB \* 49.99; ✓  redOutput.Lines.Add('To use the internet at our hotel will cost you ' +  FloatToStrF(rcost, ffcurrency, 8, 2));  btnDiscount.Enabled := true; ✓  end; | **[13]** |
| 3.2.1 | Extract the selected hotel from the combo box and the number of nights the guest plan on staying over.  The cost of a night at each hotel is as follows:   * The Oyster Box Hotel – R6215 * Tsogo Sun Elangeni Hotel – R2245 * Fairmont Zimbali Lodge – R1627 * Beverly Hills Hotel – R4071 * Hilton Durban Hotel – R1429 |  |
|  | procedure TForm1.btnDiscountClick(Sender: TObject);  var  iHotelIndex, iNum, I: integer;  rTotal, rPrice, rDiscount, rOriginal: real;  begin  iHotelIndex := cmbHotel.ItemIndex; ✓  iNum := StrToInt(edtNumNights.Text); ✓  case iHotelIndex ✓of  0: rPrice := 6215;  1: rPrice := 2245;  2: rPrice := 1627; ✓✓  3: rPrice := 4071;  4: rPrice := 1429;  end; | **[5]** |
| 3.2.2 | * **1 to 2 nights** - **does not qualify for a discount.** Display a message dialog to the user to indicate that they do not qualify for any discounts. * **3 to 4 nights - once off 15% discount** from the cost of their stay. Display the discount of the guest’s stay and their final total cost. All monetary figures must be formatted as currency to 2 decimal places. * **5 nights or more** - **10% discount that is deducted each day**. The first day of a guests stay has the 10 % discount deducted. Every day after is charged at the previous discount amount with a further 10% deduction. This is done for every day of the guest’s stay. Clear the TRichedit. (Use a for loop to calculate and display the cost of each night’s stay). All monetary figures must be formatted as currency to 2 decimal places. | (3)  (6)  (8) |
|  | if (iNum = 1) ✓or (iNum = 2) ✓then  begin  ShowMessage('Sorry, you do not qualify for any discounts'); ✓  End  else if (iNum = 3) ✓ or (iNum = 4) ✓ then  begin  rDiscount := rPrice \* iNum \* 0.15; ✓  rTotal := (rPrice \* iNum) - rDiscount; ✓  redOutput.Lines.Add('Your 15% discount earned on your stay is: ' +  FloatToStrF(rDiscount, ffcurrency, 5, 2)); ✓  redOutput.Lines.Add('The total cost of your stay: ' + FloatToStrF(rTotal,  ffcurrency, 5, 2)); ✓  end;  if iNum >= 5 then✓  begin  redOutput.Lines.clear;  redOutput.Lines.Add('Day' + #9 + 'Original Price' + #9 + #9 +  'Discounted Balance');  for I := 1 to iNum do✓  begin  rNewPrice:=rPrice; ✓  rDiscount := rPrice \* 0.1; ✓  rPrice := rPrice - rDiscount; ✓  redOutput.Lines.Add(intToStr(I) + #9 + FloatToStrF(rNewPrice✓, ffcurrency,  5, 2) + #9 + #9 + FloatToStrF(rPrice✓, ffcurrency, 5, 2)); ✓//***2 decimal throughout***  end; end;  end; | **[17]** |
|  | QUESTION 3 TOTAL: | **[35]** |

***Question 4(1)***

|  |  |  |
| --- | --- | --- |
|  | Question Mark Allocation | **Mark** |
| 4.1 |  |  |
|  | **Retrieve full name and ID** | (2) |
|  | **Take the 1st 3 characters of the guest name, make sure it is in upper case.** | (3) |
|  | **Take the 1st characters of the guest surname, make sure it is a upper case.** | (3) |
|  | **Take the last 2 characters of the guest surname, make sure it is in lower case.** | (2) |
|  |  |  |
|  | **Gender is determined by taking the 7th – 10th characters,**  **if the number is from 0000 to 4999 than the guest is female else the guest is male.** | (5) |
|  | **Randomly generate a 4 digit number.** | (3) |
|  | **Display the code generated.** | (2) |
|  |  |  |
|  | procedure TfrmPassCode.btnPassCodeClick(Sender: TObject);  var  sFullname, sID, sA, sB, sC, sD, sE,scode: String;  iID, irandom: integer;  begin  sFullname := edtFullName.Text; ✓  sID := edtIDNumber.Text; ✓  sA := UpperCase✓ (copy(sFullname, 1✓, 3)); ✓  sB := UpperCase(copy(sFullname, Pos(' ', sFullname) ✓+1✓, 1)); ✓//***corerct use of copy***  sC := LowerCase✓ (copy(sFullname, length(sFullname) - 1)); ✓  iID := StrToInt(copy(sID, 7✓, 4✓));  if (iID < 5000) then✓  begin  sD := 'Female'; ✓  end  else  begin  sD := 'Male'; ✓  end;  sE:=IntToStr ✓ (RandomRange(1000 ✓,10000 ✓));  scode:=sa+sb+sc+sd+se; ✓  lblPassCode.Caption:=scode; ✓  end; |  |
|  | QUESTION 4(1) TOTAL: | **[20]** |

***Question 4(2)***

|  |  |  |
| --- | --- | --- |
| 4.1 |  |  |
|  | loop 1-15 | (2) |
|  | Randomly generate and store numbers | (2) |
|  | procedure TForm1.btnBillClick(Sender: TObject);  var  i:integer;  begin  //4.1  for I := 1 ✓to 15✓ do  begin  arrBill[i] ✓:=RandomRange(1000,27001); ✓//**correct range**  end;  end; | **[4]** |
| 4.2 |  |  |
|  | Loop 1-15 | (1) |
|  | Display guest name and bill | (3) |
|  | procedure TForm1.btnDisplayClick(Sender: TObject);  var  i:integer;  begin  //code 4.2 here  redtDisplay.Clear;  redtDisplay.Lines.Add('Guest Name'+#9#9+'Bill Amount');  for I := 1 to 15 do✓  begin  redtDisplay.Lines.Add(arrNames[i] ✓+#9#9+FloatToStrF(arrBill[i] ✓,ffCurrency,8,1✓));  end;  end; | **[4]** |
| 4.2 |  |  |
|  | Loop 1-15 | (1) |
|  | Calculate and store bill after discount | (3) |
|  | procedure TForm1.btnDiscountClick(Sender: TObject);  var  i:integer;  begin  //4.3  for I := 1 to 15 do✓  begin  arrBill[i]:=arrBill[i] ✓-(arrBill[i] ✓\*0.35); ✓  end;  end; | **[4]** |
|  |  |  |
| 4.4 |  |  |
|  | Loop 1-15 | (1) |
|  | Sum of numbers | (1) |
|  | After loop calculate average | (1) |
|  | Display average | (1) |
|  | procedure TForm1.btnAverageClick(Sender: TObject);  var  i:integer;  rsum:real;  begin  //4.3  for I := 1 to 15 do✓  begin  rsum:=rsum+arrBill[i]; ✓  end;  ravg:=rsum/15; ✓  ShowMessage('The average bill is'+FloatToStrF(ravg,ffCurrency,8,2)); ✓  end; | **[4]** |
| 4.5 |  |  |
|  | Loop 1-15 | (1) |
|  | If statement to determine if mark is above average | (1) |
|  | Display guest name and bill amount on a new line | (2) |
|  | procedure TForm1.btnAboveAvgClick(Sender: TObject);  var  i:integer;  begin  //4.5 code here  redtDisplay.Clear;  for I := 1 to 15 do✓  begin  if(arrBill[i]>ravg)then✓  begin  redtDisplay.Lines.Add(arrNames[i] ✓+#9+FloatToStr(arrbill[i] ✓));  end;  end;  end; | **[4]** |
|  | Question 4(2)TOTAL: | **[20]** |