**JUNE EXAMINATIONS-2017**

**INFORMATION TECHNOLOGY : GRADE 11**

**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 | 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; | **5** |
| 2.1.2 | 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; | **6** |
| 2.1.3 | 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; | **8** |
| 2.1.4 | 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; | **8** |
| 2.2.1 | 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; | **4** |
| 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));  if totCost > 400000 then**🗸**  begin  pnlBudget.Visible := true; **🗸**  end;  end; | **7** |
|  | **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  redDisplay.Lines.Add(arrHotel[x]+#9+IntToStr(arrRooms[x])+#9+  FloatToStrF(arrRates[x],ffCurrency,7,2)); **🗸**  end;  end;  end; | **5** |
| 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  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; | **10** |
| 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** |