MARKING GUIDELINES

EXAMINATION	NATIONAL SENIOR CERTIFICATE
GRADE	12
DATE	NOVEMBER 2025
SUBJECT	INFORMATION TECHNOLOGY
PAPER	1
MARK TOTAL	150
DURATION (HOURS)	3
NUMBER OF PAGES	26

Learner :	

QUESTION 1	41	0
QUESTION 2	38	0
QUESTION 3	40	0
QUESTION 4	31	0
TOTAL	150	0

QUESTION 1: GENERAL PROGRAMMING SKILLS			
QUESTION	Description	Max Mark	Learner

1,1	frmQuestion1.Color := clBtnFace;
	lblQ1_1.Font.Size := 36;
	lblQ1_1.Left := 200;
	lblQ1_1.Font.Name := 'Forte';
		1
		1
		1
		1
		4	

1,2	assignfile(tfile,'PollutionData.txt');
	reset(tFile);
	while not eof(tfile) do
	readln(tFile,slIne);
	memQ1_2.
	Lines.Add('* ' + sLine);
	closefile(tFile);
		1
		1
		1
		1
		1
		1
		1
		7	

1.3.1	sWord := edtQ1_3_1.Text; for I := length(sWord) downto 1 do sReverse := sReverse + sword[I]; pnlQ1_3_1.Caption := sWord + ' - ' + sReverse;	1 1 1 1 1 1 1 1	8	
1.3.2	if sReverse = sWord then edtQ1_3_2.Text := 'It is a palindrome' else edtQ1_3__2.Text := 'It is NOT a palindrome' ;	1 1 1 1 1 1	6	
1.4.1	imgQ1_4.Picture.LoadFromFile('Indicators.jpg'); imgQ1_4.Stretch := true	1 1	2	

1.4.2	rLevelCount := strtofloat
	(inputbox('Enter pollution number for ' ,cmbQ1_4_2.Text,'4.4'));
	if ((cmbQ1_4_2.ItemIndex = 0)
	and
	(rLevelCount > 250))
	or
	((cmbQ1_4_2.ItemIndex = 1) and (rLevelCount > 420)) or
	((cmbQ1_4_2.ItemIndex = 3) and (rLevelCount > 200)) or
	((cmbQ1_4_2.ItemIndex = 4) and (rLevelCount > 500)) then
	lblQ1_4_2.caption := 'Hazardous' ;
	lblQ1_4_2.Color := clred;
	else if (cmbQ1_4_2.ItemIndex = 2) and (rLevelCount > 500) or
	((cmbQ1_4_2.ItemIndex = 5) and (rLevelCount > 15)) then
	lblQ1_4_2.caption:= 'Very unhealthy'
		1
		1
		1
		1
		1
		1
		1
		1
		1
		1
		1
		1
		1
		1
		1
		1
			14	

	TOTAL SECTION A:	**41**		**0**

QUESTION 2 – DATABASE PROGRAMMING

QUESTION	Description	Max Mark		Learner
2.1	**SQL STATEMENTS**			
2.1.1	Select *	1	2	
	from tblDrivers	1		
2.1.2	Select SystemID, Surname , Suburb	1	4	
	From tblDrivers	1		
	Where Suburb Like	1		
	''M%''	1		
2.1.3	Select DateIssued, GroupCode from tblOffences	1	3	
	where month(DateIssued) =	1		
	+ spnQ2.text	1		
2.1.4	Select SystemID, format(Amount,"currency")	1	5	
	as FineAmount, DateIssued from tblOffences	1		
	where not Ispaid	1		
	and date() -	1		
	dateIssued > 45	1		
2.1.5	select Suburb,count(*)	1	5	
	as [Number of offences per suburb]	1		
	from tblOffences,tblDrivers	1		
	tblOffences.SystemID = tblDrivers.SystemID	1		
	group by Suburb	1		
2.2	**DATABASE MANIPULATION USING DELPHI CODE**			
2.2.1	icount := 0;	1	11	
	tblOffences.first;	1		
	while not tblOffences.eof do	1		
	if tblOffences['SystemID'] = tblDrivers['SystemID'] then	1		
	if not tblOffences['IsPaid'] then	1		
	redQ2.Lines.Add(inttostr(tblOffences['FineID'])+#9 +	1		
	Datetostr(tblOffences['DateIssued']) + #9	1		
	+ floattostrf(tblOffences['Amount'],ffcurrency,10,2))	1		
	else inc(iCount);	1		
	tblOffences.next;	1		
	redQ2.Lines.Add('Number of paid fines: '+ inttostr(icount));	1		
2.2.2	tblGroupCodes.First;	1	8	
	for i := 1	1		
	to 3 do	1		
	redQ2.SelAttributes.Style := [fsbold];	1		
	scode := tblgroupCodes['groupcode'] ;	1		
	redQ2.Lines.Add(scode);	1		
	redq2.Lines.Add(tblgroupCodes['description']);	1		
	tblgroupCodes.Next;	1		
	TOTAL SECTION B:	**38**		**0**

QUESTION 3 – OOP				
QUESTION	**Description**	**Max Mark**		**Learner Mark**
3.1.1	**constructor Create(sServiceType : String; iDuration : Integer ;** Heading 4 parameters 2 integers, 2 strings assign parameters to attributes Assign true to fIsPlanned	1 1 1 1 1	5	
3.1.2	**function TDisruption.b_getLevel: Integer;** Result := fLevel;	1 1	2	

3.1.3	**function TDisruption.c_calcPriority: Integer;**
	case of
	fServiceType[1]
	'E': Result := fLevel * 30; // Electrical
	'W': Result := fLevel * 25; // Water
	'G': Result := fLevel * 10; // Garbage
		1
		1
		1
		1
		1
		1
			6	

3.1.4	**procedure TDisruption.d_setEmergency;**
	fIsPlanned := false;
	if fLevel < 5 then
	fLevel := fLevel + 1;
		1
		1
		1
			3	

3.1.5	**function TDisruption.e_determineStatus: String;** if not fIsPlanned then sStatus := 'Emergency' else begin else if (fLevel >= 3) and (fLevel <= 4) then sStatus := 'Moderate' s Result := sStatus;	1 1 1 1 1 1 1	7	

3.2.1a	sedQ3_2_1Duration.Value := Random(12) + 1; // 1-12 hours
	sWeekday := arrWeekdays[Random(7) + 1];
	iLevel := Random(5) + 1; // 1-5
		1
		1
		1
			3	

3.2.1b	sType := cmbQ3_2_1ServiceType.Text; objDisruption := TDisruption.Create (sType, iDuration, sWeekday, iLevel);	1 1 1	3	
3.2.2	shpQ3_2_2.Left := 7 + (objDisruption.calcPriority * 2); lblQ3_2_2Priority.Caption := 'Priority: ' + IntToStr(objDisruption.calcPriority);	1 1 1 1	4	

3.2.3	if (sEnteredType = sType)
	and (sEnteredStatus <> 'Emergency') then
	begin
	objDisruption.setEmergency;
	redQ3.Lines.Add(objDisruption.toString);
	redQ3.Lines.Add('Status: ' +
	objDisruption.determineStatus);
	end
	else
	redQ3.Lines.Add('Cannot upgrade: Service type mismatch or already emergency');
		1
		1
		1
		1
		1
		1
		1
			7	

	TOTAL SECTION C	**40**		**0**

QUESTION 4 – PROBLEM SOLVING				
QUESTION	**Description**	**Max Mark**		**Learner Mark**
4,1	for iCol := 0 to 3 do	**1**	**5**	
	stgQ4_1.Cells[icol,0] := arrHeadings[iCol+1];	**1**		
	for iRow := 1 to iCount do	**1**		
	for iCol := 1 to 4 do	**1**		
	stgQ4_1.Cells[icol-1,irow] := arrPollutionData[irow,icol];	**1**		
4,2	iCountAir := 0; for k := 1 to iCount do	**1**	**15**	

	if arrPollutionData[k,2] = 'Air' then
	iSearch := 0;
	bFound := false;
		1		

	while (iCountAir > 0)	**1**		
	AND (iSearch<=iCountAir)	**1**		
	AND (not bFound) do	**1**		
	inc(iSearch);	**1**		
	if arrPollutionData[k,1] = arrAirData[iSearch,1] then	**1**		
	if strtoint(arrPollutionData[k,4]) > strtoint(arrAirData[iSearch,3]) then	**1**		
	arrAirData[iSearch,3] := arrPollutionData[k,4];	**1**		
	arrAirData[iSearch,2] := arrPollutionData[k,3]	**1**		
	bFound := true;	**1**		
	if NOT bFound then	**1**		
	inc(iCountAir);	**1**		

	arrAirData[iCountAir,1] := arrPollutionData[k,1];
	arrAirData[iCountAir,2] := arrPollutionData[k,3];
	arrAirData[iCountAir,3] := arrPollutionData[k,4];
		1		

	assign all three values to arrAirData	**1**		

4,3	sElement := cmbElement.Text;
	iRecords := 0;
	rAvg := 0;
		1	11	

	for k := 1 to iCountWater do	1		
	if pos(sElement,arrWaterData[k,2])> 0 then	1		
	inc(iRecords);	1		
	sTemp := arrWaterData[k,2];	1		
	delete(stemp, 1, pos(':', stemp)+1);	1		
	delete(stemp,pos('mg/L',sTemp)-1,length(stemp)-pos('mg/L',sTemp)+2);	1		
	rAvg := rAvg + StrToFloat(sTemp);	1		
	redQ4_3.Lines.Add(inttostr(iRecords) + ' records with ' + sElement);	1		
	redQ4_3.Lines.Add('Average '+ sElement + ' pollution: ' +floattostrf(	1		
	rAvg/iRecords,ffFixed,10,5) + ' mg/L');	1		
	TOTAL SECTION D	31	0	0