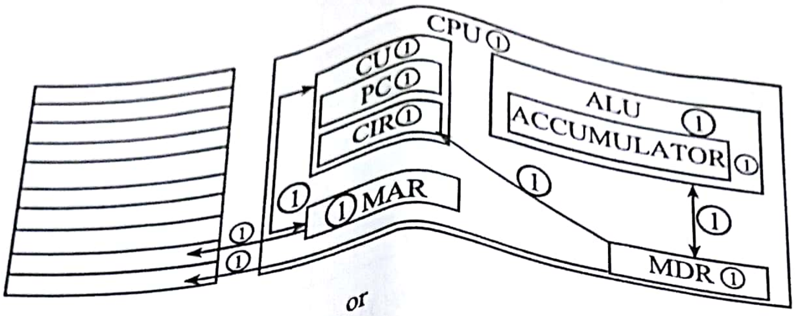
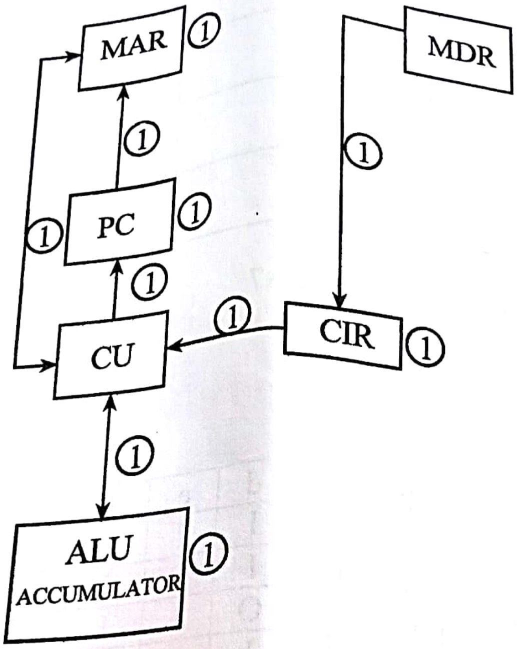
2 (a)



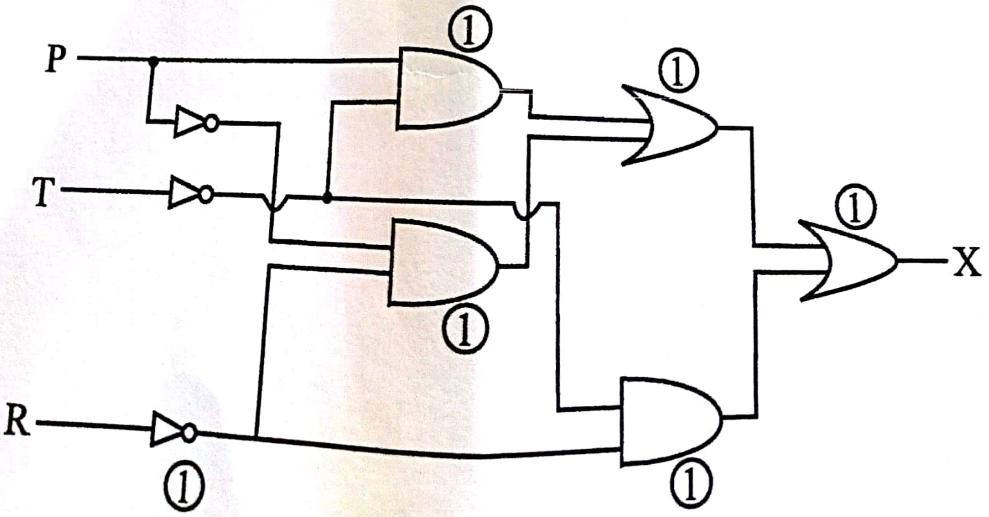
or



[Max 10]

(b) (i) X = (NOT P AND T) OR (NOT W AND NOT T) [1]

(ii)



[5]

Section A

1 (a) (i)

(Accept appropriate alternative solution) [Max 5]

(ii)

P	T	R	Workspace	X	
O	O	O		1	1
O	O	1		O	
O	1	O		I	1
O	1	1		O	
1	O	O		O	1
1	O	1		O	
1	1	O		I	1
1	1	1		I	

[Max 4]

(b) Spreadsheet

	A	B	C	D	E
1	1A	(1) = NOT (A1)			
2		a			
3	1B	(1) = NOT (A3)	(1) = AND (B1,B3)		
4		b	c	(1) = OR(C2, C5)	
5				f	
6			(1) = OR(A7, B3)		(1) = AND (B3,B7)
7	1C	(1) = NOT (A7)	d		
8		e			
9					

hint: Inputs are entered in A1, A3 an A7

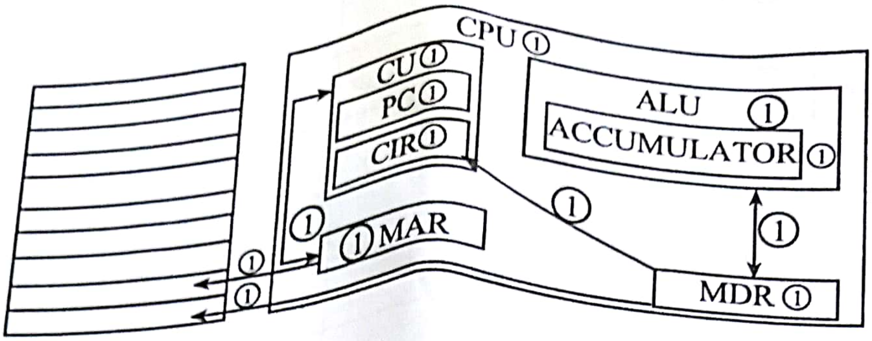
[7]

Truth Table

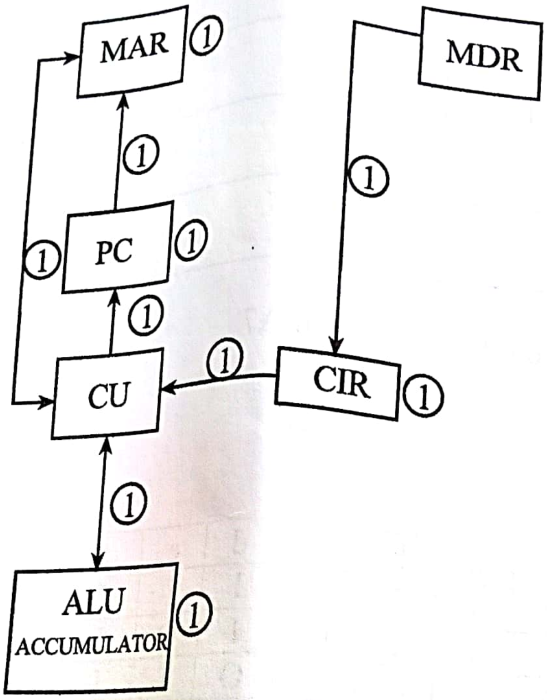
A	B	C	a	b	c	d	e	f	X	
O	O	O	I	I	O	I	I	I	I	1
O	O	I	I	I	O	I	O	I	O	
O	I	O	I	O	I	O	I	I	I	I
O	I	I	I	O	I	I	O	I	O	
I	O	O	O	I	O	I	I	I	I	I
I	O	I	O	I	O	I	O	1	O	
I	I	O	O	O	O	O	I	O	O	I
I	I	I	O	O	O	I	O	I	O	

[4]

2 (a)



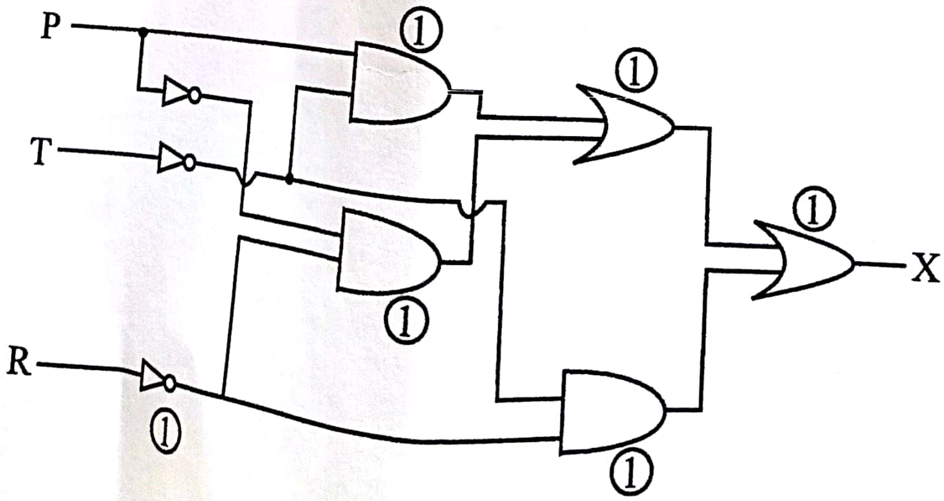
or



[Max 10]

(b) (i) X = (NOT P AND T) OR (NOT W AND NOT T) [1]

(ii)



[5]

(iii)

P	T	W	X	
O	O	O	I	1
O	O	I	O	
O	I	O	I	1
O	I	I	I	
I	O	O	I	1
I	O	I	O	
I	I	O	O	1
I	I	I	O	

[4]

Section B

3 (a) (i)

```
Public function length check (Byval identif as string) returns
Boolean(1)

If identiflength > 32 then (1)
      Return     false (1)
else return true (1)

   endif
End Function
```

[5]

(ii)

```
                                        (1)
Public Function Character Check (ByVal identif as string)
                                  returns Boolean (1)

     Dim i as integer     } (1)
     Dim flag as Boolean  }
     Flag = true  (1)

       For i  =  1 to identif . length (1)
       If mid (identif, i, 1) > "z"  (1)  or mid (identif, i, 1)  <  "a" (1)
(1)    THEN Flag = False  (1)

       Endif
     Next i
       return  flag (1)

  End Function
```

[10]

(b) (i)

```
Dim    TestString as String
   TestString  =  Console.readline(1)
   If not (1) lengthcheck (TestString)(1) then
     TestString   =  left (TestString.32)
                      (1)            (1)

Endif
```

```
Console.writeline(TestString) (1)
```

[Max 5]

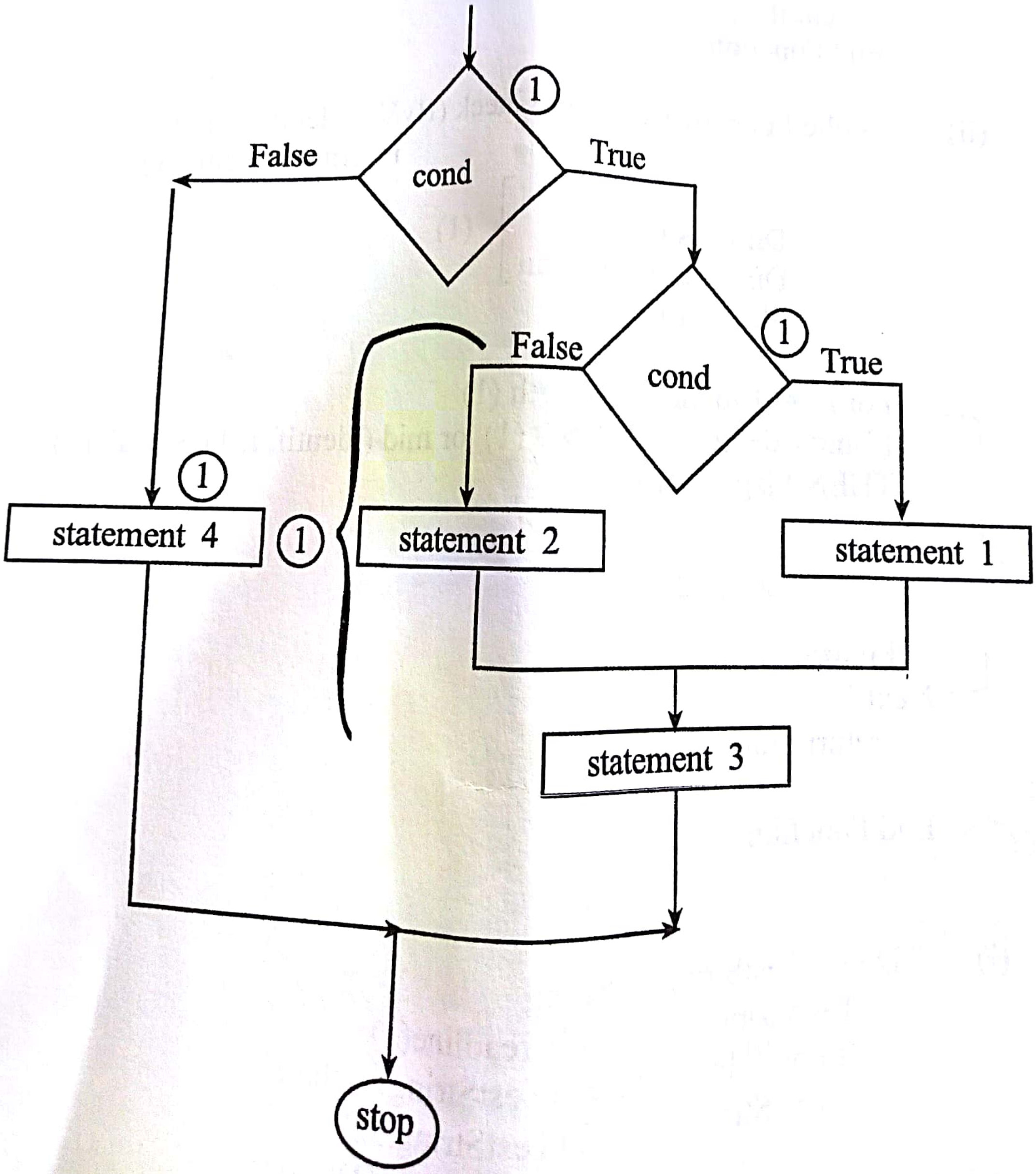
(ii) NB: continued from 3 (b) (i) above

```
Dim newString as String (1)
If not CharacterCheck (TestString) then
                  (1)           (1)
    For i = 1 to TestString. length
        If Character Check (mid (TestString, i, 1)
                 (1)       (1)                (1)
        Then
(i)         newstring = newstring & mid (TestString, i, 1)
                           (1)       (1)         (1)
        endif
    next i
console.writeline("revised identifier is:-" & newstring)
                                               (1)
```

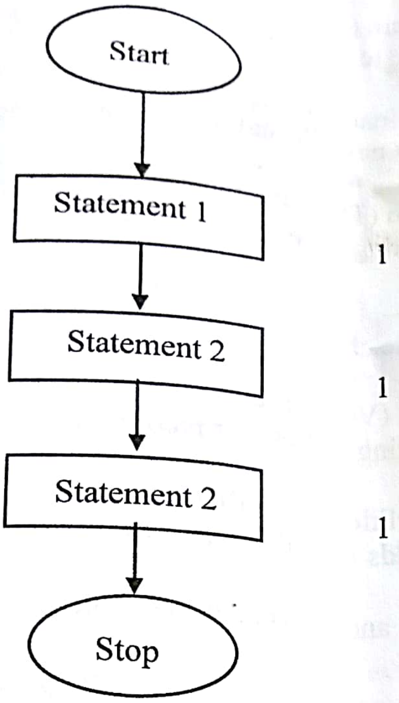
[12]

§§§§§§§§23

3 (c) (i)

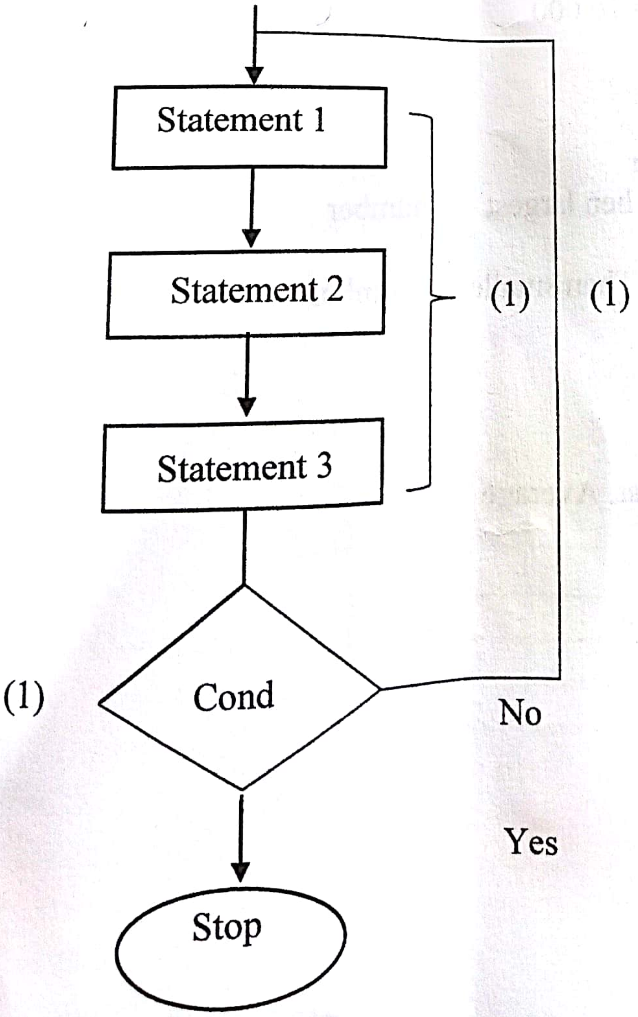


(ii)


Start
Statement 1
1
Statement 2
1
Statement 2
1
Stop

[2]

(iii)


Statement 1
Statement 2
Statement 3
(1)
(1)
(1)
Cond
No
Yes
Stop

[3]

(d) (i) Carfile = New Filestream ("Carfile.DAT", Filemode.open) [1]

- Used to link file to the file name [1]

Carfile writer = New Binary Writer (Carfile) [1]

- Used to create a new file and open it for writing [1]

Carfile.position = Harsh (This.car.vehicles ID) [1]

- Used to get starting address for record [1]

Carfile Writer.close ( ) [1]

- Uscd to close file channel [1]

Carfile.Positon = Harsh (Vehicle ID) [1]

- Used to get starting address for record [1]

Mycar.vehicle ID = CarFile.Reader.ReadString [1]

- Used to read fields from the binary file [1]

NB: CarFile, vehicle ID and This car are variables.
Accept any variations

Accept any 4 [Max8]

(ii) Any two statements from the above except Harsh

1 mark each [Max 2]

*Accept programming language variations

4 (a)

```
    Average = 0, Sum = 0
    Largest = 0, smallest = 20 000                          [1]
    For C = 1 to 50                                         [1]
       Input number
       Sum = sum + number                                   [1]
(1)    If number > largest Then largest = number            [1]
       Else                                                 [1]
       If number < smallest Then smallest = number          [1]
       Endif
    Next C                                                  [1]
    Average = sum/ C
    Display smallest, largest, Average                      [1]
```

Accept other variations [1]

[10]

(b)

```
Dim average, sum, largest, smallest As Integer
Dim count integer, numbers As integer                                      [2]
average = 0, sum = 0, smallest = 20 000, largest = 0                       [2]
For count = 1 to 50
        Number= Input Box ("Enter number")                                 [1]
        Sum = sum + number                                                 [1]
(1)     If number > largest Then
        Largest = number                                                   [1]
Else
        If number < smallest Then
        Smallest = number                                                  [1]
Endif
Next count
  Average = sum/count
  Msgbox/ ("Average is:" & average)                                        [1]
  Msgbox/ ("largest is:" & largest)          (1)
  Msgbox/ ("Smallest is:" & smallest)
```

Accept other variations

[11]

(c) (i)

[7]

(ii)

Item	Ptr	Printed Output	
"T"	1	"J"	
"T"	4	"U"	[2]
"T"	5	"S"	[2]
"T"	7	"T"	[2]

(d)

```
Display "Type in a number or zero(o) to stop"     [1]
Input number                                       [1]
WHILE number < > O DO                              [1]
   Square = number * number                        [1]
   Display "The square is" square                  [1]
   Display "Type in a number or zero(o) to stop"   [1]
   Accept number                                   [1]
ENDWHILE                                           [1]
                                              [Max 7]
```

(e)

```
      (1) (1)     (1)
Dim Array [8]  as Integer
     (1)        (1)
Dim counter as Integer
            (1)        (1)
    ┌ For counter = 0  to 7
(1) │ Array [counter]  =  Input Box  ("Enter number")
    │       (1)              (1)
    └ Next Counter
      .
      .
      .
                                                   [9]
```

Accept other variation of the program

SECTION C

```
               (1)
5   CREATE TABLE students

(a) (i)  Student ID(1) int (1) (4)  PRIMAY KEY,
         Surname char(20),
             (1)     (1)
         First_name char (20),
              (1) (1)
         Class  char (2)

             (1)      (1)
         Fees_paid number (4)
                (1)    (1)
         );; (1)                                   [14]
```

```
(ii) ┌ SELECT     Surname, firstname
     │                 (1)
(1) ─┤ FROM       students
     │              (1)
     └ WHERE      class = "1A" AND  feespaid < 100;
                    (1)      (1)        (1)                  [6]
```

(iii) **DELETE FROM students** WHERE **first_name = "Ralph(1)";** [3]

(b)

[7]

(a) (i) CREATE DATABASE schoolfees; [1]

(ii) **student details table**

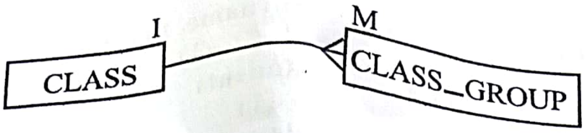
```
CREATE TABLE   Student_details          [1]
(Student_name text (10),                [1]
Address text (50),                      [1]
Student_ID text (5) PRIMARY KEY,        [1]
class text (10),                        [1]
);                                      [1]
```

Invoice table

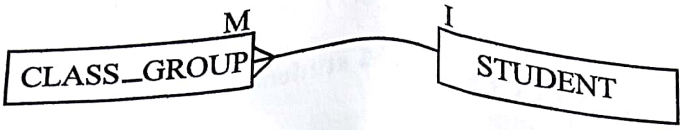
```
CREATE TABLE   Invoice                  [1]
(student_ID text (5) FOREIGN KKEY,      [1]
fees amount currency,                   [1]
fees_paid currency,                     [1]
invoice_number text (5),                [1]
due date                                [1]
fees_balances currency,                 [1]
);
```

```
SELECT student_ID, fees amount, fees_paid,               [1]
            invoice_numbr, due_date, fees amount         [1]
       fees_paid as fees_balance                         [1]
From Invoice                                             [1]

WHERE Fees Balance > O                                   [1]

ORDER BY INVOICE_number                                  [1]
```

(b) (i)



[1]



[1]

(ii)

```
STREET student ID, FirstName                    [1]
FROM        student                             [1]
WHERE       Tutor_group  =  "3W"                [1]
ORDER By Last Name ASC                          [1]
```

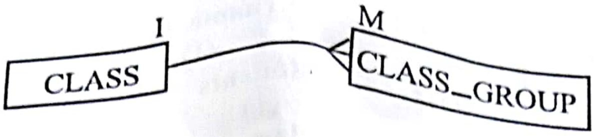
(iii)

```
SELECT STUDENT, LastName
FROM STUDENT, CLASS_GROUP                                 [1]
WHERE classID  =  "No13"                                  [1]
AND class_Group. StudentID = student.studentID            [1]
                                                          [1]
```

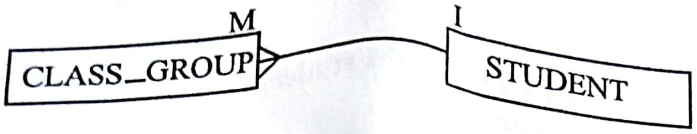
OR

```
SELECT STUDENT, Last Name                                 [1]
FROM STUDENT INNER JOIN  CLASS_GROUP                      [1]
ON CLASS_GROUP.student ID = STUDENT.studentID             [1]
WHERE classID = "No13"                                    [1]
```

(b) (i)



[1]



[1]

(ii)

```
STREET student ID, FirstName
FROM        student
WHERE       Tutor_group  =  "3W"
ORDER By Last Name ASC
```

[1]
[1]
[1]
[1]

(iii)

```
SELECT STUDENT, LastName
FROM STUDENT, CLASS_GROUP
WHERE classID  =  "No13"
AND class_Group. StudentID = student.studentID
```

[1]
[1]
[1]
[1]

OR

```
SELECT STUDENT, Last Name
FROM STUDENT INNER JOIN  CLASS_GROUP
ON CLASS_GROUP.student ID = STUDENT.studentID
WHERE classID = "No13"
```

[1]
[1]
[1]
[1]

ZIMBABWE SCHOOL EXAMINATIONS COUNCIL

General Certificate of Education Advanced Level

MARKING SCHEME

NOVEMBER 2018