ZIMSEC

For Performance Measurement

CYBERSPACE INC

COMPUTER SCIENCE

ZIMSEC 2023 PAPER2 MS

2023 SOLUTIONS

FULLY GUIDED SOLUTIONS FOR ZIMSEC EXAMS

DIRECTED BY:

LIONEL CHENYIKA [THE CYBERWAVE]

@CYBERSPACE-2024

Table of Contents

Acknowledgments

I wish to extend my heartfelt gratitude to my parents, whose unwavering support and belief in my abilities have been the cornerstone of my journey in computer science. Their encouragement has propelled me forward, shaping my aspirations and accomplishments. Additionally, I am profoundly grateful for the guidance and blessings bestowed upon me by the divine throughout this endeavor.

Appreciation

To the esteemed audience, I offer sincere appreciation for your respect and encouragement of the cyberwave that defines our collective pursuit in this field. Your recognition and support are invaluable contributions to the advancement of computer science. I am now authorized to produce educational resources by the Ministry Of Primary and Secondary Education **[MOPSE]** because of your trust in me.

Contact Information
For inquiries, permissions, or further information, please reach out to me at:

0778201864
0716057680

I am delighted to announce the publication of my book chronicling the journey to becoming a full-stack remote software engineer. This narrative encapsulates the challenges, triumphs, and invaluable lessons learned along the path to professional fulfillment. It also includes my adventures as I struggled to balance being an introvert in high school and focusing on study. The title of the book is called HALF HUMAN, HALF ANDROID

``THE PROBLEM IS NOT THE SOFTWARE, ITS THE PROGRAMMER

-THE CYBERWAVE

SECTION A

Answer any one question from this section.

1 (a) A chemical plant gives out a warning signal (w = 1) when the process goes wrong. A logic network is used to provide input and decide whether or not w = 1.

Input	Binary	Plant status
C	1 0	Chemical Rate = 10 m^3/s Chemical Rate< 10 m^3/s
T	1 0	Temperature = 87°C Temperature > 87°C
X	1 0	Concentration > 2 moles Concentration = 2 moles

A warning signal (w = 1) will be generated if either
a) Chemical Rate < 10m^3/s
OR b) Temperature > 87°C and Concentration > 2 moles
OR c) Chemical rate = 10m^3/s and Temperature > 87°C

(i) Formulate a logic expression for the conditions. [3]

(ii) Draw a logic network to represent how this chemical plant is monitored. [7]

(iii) Compute a truth table for the logic network in (ii). [5]

(b) Draw a clearly labelled diagram that represents the Von Neumann Architecture. [5]

Before we get started

Our driver is the binary equivalence in respect to the plant status i.e Chemical Rate =10m^3/s is equivalent to a high or 1 Also remember the following rules of the Cyberwave

Rule #1 –Logical Operators

OR==(+)

AND==(*)

Rule#2-Assosiativity(Ass)

When the question says something like A OR B OR C imagine as if its something like A + B +C which can be further simplified to (A+B)+C WHICH IS EQUIVALENT TO A+(B+C)

Solution 1a-Logic Circuits

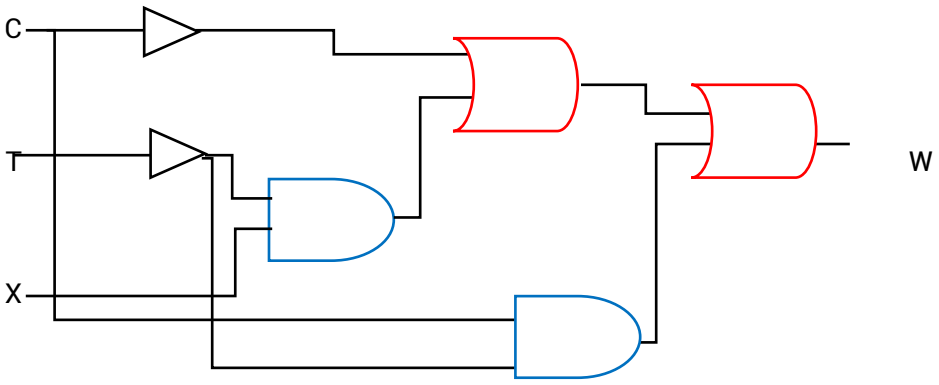
Lets first start by writing a logical expression

W=1 IF C=0 OR (T=0 AND X=1) OR (C=1 AND T=0)

NB: **RED** is for the main gates and **BLUE** is for inner gates

REMEMBER→ OR/+ is associative so brackets can be placed anywhere

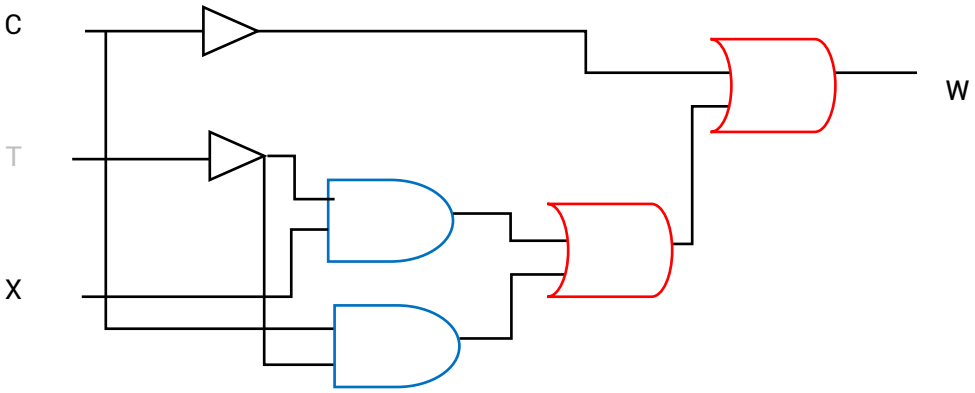
CASE 1- Using (C=0 OR (T=0 AND X=1)) OR (C=1 AND T=0)



C	T	X	C'	T'	T'.X	C'+(F)	T'.C	(G)+(H)
0	0	0	1	1	0	1	0	1
0	0	1	1	1	1	1	0	1
0	1	0	1	0	0	1	0	1
0	1	1	1	0	0	1	0	1
1	0	0	0	1	0	0	1	1
1	0	1	0	1	1	1	1	1
1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

Whether you say XTC or CTX , the answer we expect to see is 2 0s with correct inputs nomatter the position

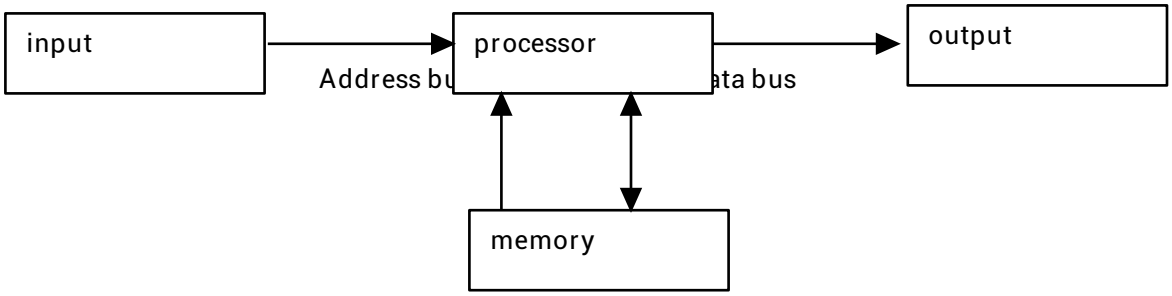
CASE2- Using C=0 OR ((T=0 AND X=1) OR (C=1 AND T=0))



C	T	X	C'	T'	T'.X	(H)+(F)	T'.C	(G)+C'
0	0	0	1	1	0	0	0	1
0	0	1	1	1	1	1	0	1
0	1	0	1	0	0	0	0	1
0	1	1	1	0	0	0	0	1
1	0	0	0	1	0	1	1	1
1	0	1	0	1	1	1	1	1
1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

b)Solution 1b –Von Neuman Architecture

(b) Draw a clearly labelled diagram that represents the Von Neumann Architecture. [5]



2 (a) Draw the logic circuit for each of the Boolean expressions given below

(i) A . B + C [2]

(ii) $\overline{(P+Q).(P+R)}$ [3]

(b) Draw a flowchart or other diagram to illustrate how the sequence of instructions are carried out during interrupt servicing. [8]

Solution 2a- Logic Expressions

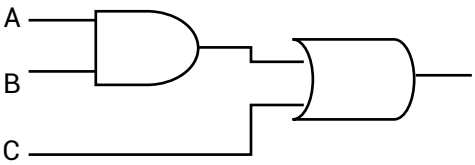
See question 1 for more info on logic gates secrets

BOMDAS Multiplication wins over addition

Lets clear some confusion

When given both + and . use BOMDAS to determine which to prioririze

i)

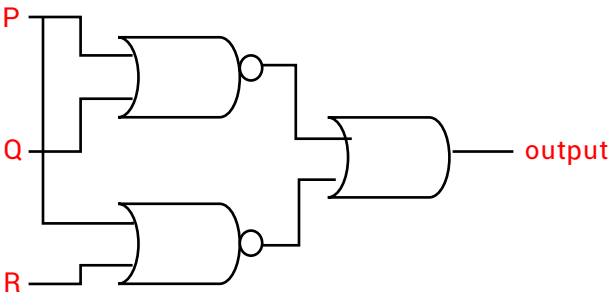


Solution2a(ii)

food for thought(De'morgans theorem)

$$\overline{A.B} = \overline{A} + \overline{B}$$

Hence from our question here, $\overline{(P+Q).(P+R)} = \overline{(P+Q)} + \overline{(P+R)}$



Solution2b-Interrupt Service Handling

Start

1. Interrupt occurs: An external event or internal condition triggers an interrupt signal.
2. Check interrupt enable flag :if the interrupt flag for the specific source is enabled, proceed to step 3. Otherwise, ignore the interrupt and continue normal program execution.
3. Acknowledge interrupt: Send an acknowledgement signal to the interrupt source to indicate the interrupt is being handled.
4. Save program state: The CPU saves the current Program Counter (PC) and other relevant registers (e.g., status register, stack pointer) onto the interrupt stack. This creates a snapshot of the program's state before the interrupt.
5. Disable further interrupts :To prevent nested interrupts, further interrupts are typically disabled while the current interrupt is being handled.
6. Determine interrupt source: Identify the specific device or event that triggered the interrupt using an interrupt vector table or other mechanism.
7. Execute interrupt service routine (ISR):Jump to the pre-defined code specific to the interrupt source (ISR). This code handles the specific task associated with the interrupt, such as reading data from an I/O device or handling an error condition.
8. Clear interrupt flag: Reset the interrupt flag for the specific source to indicate that the interrupt has been serviced.

9. Restore program state: The CPU restores the saved program state (PC, registers) from the interrupt stack, effectively returning to the point where the program was interrupted.
10. Re-enable interrupts: If previously disabled, re-enable further interrupts to allow them to be received again.
End

HOMEWORK: USING THE INSTRUCTIONS I HAVE WRITTEN ABOVE, DRAW THE RELEVANT FLOWCHART/DIAGRAM

(c) In a processor three instructions (J followed by K, followed by M) are processed using pipelining.

The following table shows the five stages that occur when instructions are fetched and executed.

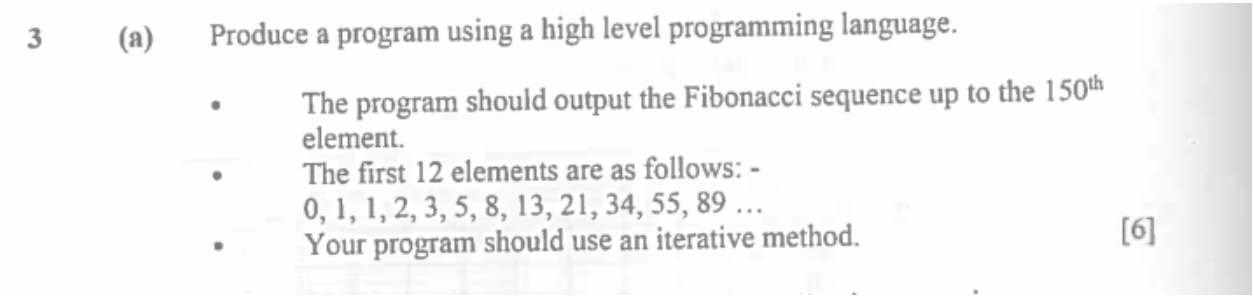
STAGE	TIME INTERVAL								
Fetch instruction	1	2	3	4	5	6	7	8	9
Decode instruction	J								
Execute instruction									
Access operand in memory									
Write result to register									

(i) The 'J' in the table indicates that instruction has been fetched in time interval 1.
Copy and complete. the table to show the time interval in which each stage of each instruction (J, K, M) is carried out. [4]

Solution 2c-Pipelining

STAGE	Time INTERVAL								
Fetch instruction	1	2	3	4	5	6	7	8	9
Decode instruction	J	K	L						
Execute instruction		J	K	L					
Access operant in memory			J	K	L				
Write result to register					J	K	L		

SECTION B



3 (a) Produce a program using a high level programming language.

- The program should output the Fibonacci sequence up to the 150th element.
- The first 12 elements are as follows: -
  0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 …
- Your program should use an iterative method. [6]

Solution 3a-Fibonacci Sequence

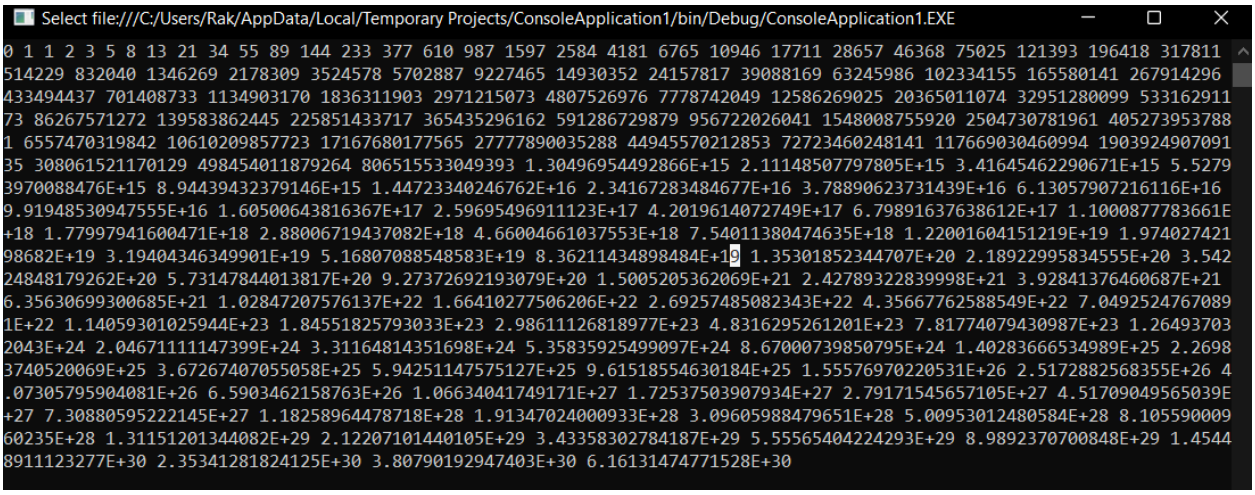
Case 1 -*Using Python (Simple and cool)*

```
fib_seq=[0,1]
for i in range(2,150):
   fib_seq.append(fib_seq[i-1]+fib_seq[i-2])
print(fib_seq)
```

```
[Running] python -u "C:\Users\Rak\AppData\Local\Temp\tempCodeRunnerFile.python"
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811,
514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352, 24157817, 39088169, 63245986, 102334155, 165580141, 267914296, 433494437,
701408733, 1134903170, 1836311903, 2971215073, 4807526976, 7778742049, 12586269025, 20365011074, 32951280099, 53316291173, 86267571272, 139583862445,
225851433717, 365435296162, 591286729879, 956722026041, 1548008755920, 2504730781961, 4052739537881, 6557470319842, 10610209857723, 17167680177565,
27777890035288, 44945570212853, 72723460248141, 117669030460994, 190392490709135, 308061521170129, 498454011879264, 806515533049393, 1304969544928657,
2111485077978050, 3416454622906707, 5527939700884757, 8944394323791464, 14472334024676221, 23416728348467685, 37889062373143906, 61305790721611591,
99194853094755497, 160500643816367088, 259695496911122585, 420196140727489673, 679891637638612258, 1100087778366101931, 1779979416004714189,
2880671943708161 20, 4660046610375530309, 7540113804746346429, 12200160415121876738, 19740274219868223167, 31940434634990099905, 51680708854858323072,
83621143489848422977, 135301852344706746049, 218922995834555169026, 354224848179261915075, 573147844013817084101, 927372692193078999176,
1500520536206896083277, 2427893228399975082453, 3928413764606871165730, 6356306993006846248183, 10284720757613717413913, 16641027750620563662096,
26925748508234281076009, 43566776258854844738105, 70492524767089125814114, 114059301025943970552219, 184551825793033096366333, 298611126818977066918552,
48316295261201016328 4885, 781774079430987230203437, 1264937032042997393488322, 2046711111473984623691759, 3311648143516982017180081,
53583592549909666408718 40, 867000739850794865 8051921, 1402836665349891529892 3761, 2269837405200686395697 5682, 3672674070550577925589 9443,
59425114757512643212875125, 96151855463018422468774568, 155576970220531065681649693, 251728825683549488150424261, 407305795904080553832073954,
65903462158763004198249 8215, 10663404174917105958145 72169, 17253750390793406377970 70384, 27917154565710512336 11642553, 45170904956503918714087 12937,
73088059522214431050203554 90, 11825896447871834976429068427, 19134702400093278081449423917, 30960598847965113057878492344, 50095301248058391139327916261,
81055900096023504197206408605, 13115120134408189533653432 4866, 21220710144010539953374 0733471, 34335830278418729487 0275058337,
55556540422429269440 15791808, 89892370700847998927429 0850145, 14544891112327726836783 06641953, 23534128182412526729525974 92098,
38079019294740253566 30904134051, 61613147477152780295835 01626149]
```

Case2-Using Vb.net

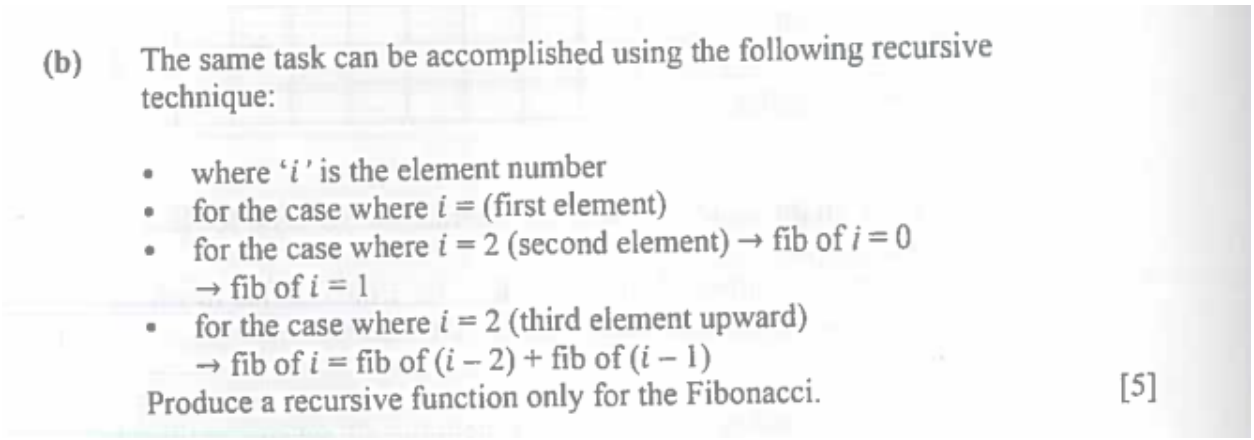
```
Module Module1

    Sub Main()
        Dim nums(149) As Double
        nums(0) = 0
        nums(1) = 1
        For i = 2 To 149
            nums(i) = (nums(i - 1) + nums(i - 2))
        Next
        For Each c In nums
            Console.Write(c & " ")
        Next
        Console.ReadLine()

    End Sub

End Module
```



Food for thought???

Do you now understand why I always encourage you to learn python????? Its less verbose.

Homework: Research why we were able to represent integers in python and why not in VB.net



(b) The same task can be accomplished using the following recursive technique:

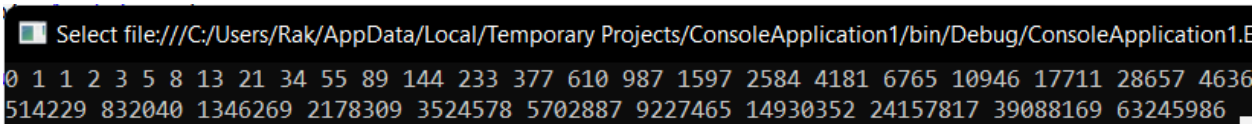
- where 'i' is the element number
- for the case where i = (first element)
- for the case where $i = 2$ (second element) → fib of $i = 0$
  → fib of $i = 1$
- for the case where $i = 2$ (third element upward)
  → fib of i = fib of $(i - 2)$ + fib of $(i - 1)$

Produce a recursive function only for the Fibonacci. [5]

Solution 3b-Recursion

Case

```
Module Module1
    Function fib(n As Integer)
        If n <= 1 Then
            Return n
        Else
            Return fib(n - 1) + fib(n - 2)
        End If
    End Function
    Sub Main()
        For i = 0 To 149
            Console.Write(fib(i) & " ")
        Next
        Console.ReadLine()
    End Sub
End Module
```



(c) For this task, you are required to produce a program using a high level programming language that works with 2 files.

- opens file 1 for reading
- opens file 2 for writing

(i) Use a text editor to create a text file with the following content: -

	* use Consolas / courier Font for blue text
< Line 1>	This is my source File
< Line 2>	Number 1 is:
< Line 3>	Number 2 is:
< Line 4>	My name is:
	* end of Consolas / courier

Produce a screen shot of the typed file showing the text content as above. [3]

Solution 3c[i]-File handling

Just follow the instructions

```
<Line1>     This is my source File

<Line2>     Number1 is:

<Line3>     Number2 is:

<Line4>     My name is:
```

Solution 3c[ii]-The code

(ii) Produce program code that reads the content of File 1 and then writes it to File 2 as follows: -

* use Consolas / Courier font
< Line 1>
< Line 2> < centre number >
< Line 3> < candidate number >
< Line 4> < Name, Surname >
< Date / time stamp >

Sample Output
This is my source file
Number 1 is: 0100015
Number 2 is: 0059
My Name is: Chipo Made
01/01/ 2020
* end of Consolas / courier font

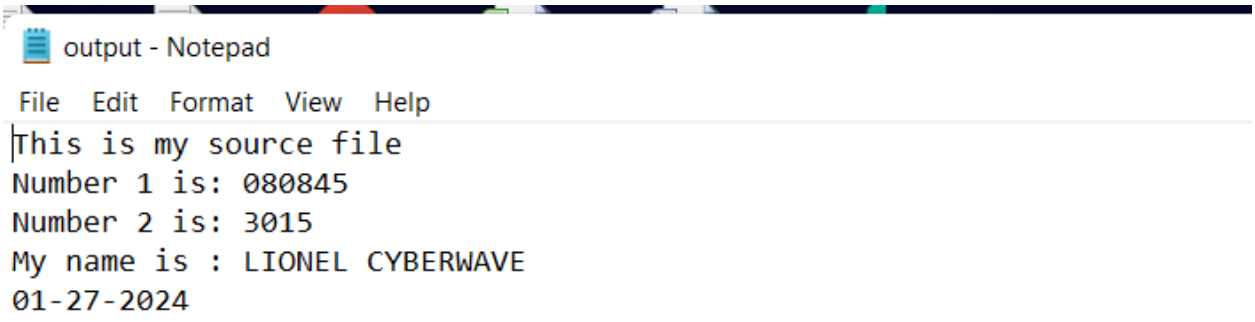
* NB: Submit only your source code for this question. [14]

6023/2 N2023

```
Imports System.IO
Imports System.Console
Module Module1
  Sub Main()
    Dim input_path, input_file, Output_file, centre_num, cand_num, name_surn, date_time As String
    WriteLine("Enter centre num")
    centre_num = ReadLine()
    WriteLine("Enter candidate num")
    cand_num = ReadLine()
    date_time = DateString
    Console.WriteLine("Enter name")
    name_surn = ReadLine()
    input_path = "C:\Users\Rak\Desktop\input.txt"
    Output_file = "C:\Users\Rak\Desktop\output.txt"
    Dim content As String = File.ReadAllText(input_path)
    content = content.Replace("<Line1>", "This is my source file")
    content = content.Replace("<Line2>", "Number 1 is: ")
    content = content.Replace("<Line3>", "Number 2 is: ")
    content = content.Replace("<Line4>", "My name is : ")
    content = content.Replace("<centre number>", centre_num)
    content = content.Replace("<candidate number>", cand_num)
    content = content.Replace("<Name,Surname>", name_surn)
    content = content.Replace("<Date/time stamp>", date_time)
    File.WriteAllText(Output_file, content)
    WriteLine("File updated successfully")
    Console.ReadLine()
  End Sub
End Module
```

Console file output

Output file



Please note I have removed the fact that the font should be consolas because some of you will hate me!!!!!

(d) Produce a program using the high level programming language that initialises an array of 20 characters with capital letters of the alphabet.
* N.B ASCII code for 'A' = 65 [4]

Solution 3d-Arrays and the Chr Function

```
Module Module1
  Sub Main()
    Dim alphabet(19) As Char
    For i = 0 To 19
      alphabet(i) = Chr(65 + i)
    Next
    For Each i In alphabet
      Console.WriteLine(i)
    Next
    Console.ReadLine()
  End Sub
End Module
```

The goal is to initialize only

(e) Produce a program by copying and modifying your code from **(d)** above or otherwise.
The program in this case works with a 2 D array, each element containing 2 characters. (capital letters and small letters)
- Array should be a 20 × 20 elements.

*N. B ASCII code for 'a' = 97. [10]

Solution 3e-2D Arrays

```
Module Module1
  Sub Main()
    Dim array(19, 19) As Char
    For i = 0 To 19
      For j = 0 To 19
        array(i, j) = Chr(65) & Chr(97)
      Next j
    Next i
    For j = 0 To 19
      Console.WriteLine(array(i, j))
    Next j
    Console.WriteLine("")
    Console.ReadLine()
  End Sub
End Module
```

(f) Multiplication can be considered to be repeated addition.
Produce a program using pseudocode or a high level programming language that multiplies 2 positive integers.

- The program should implement an iterative function (A full description of function is required).
- There should (only this line is required) be a calling statement in the main program. [8]

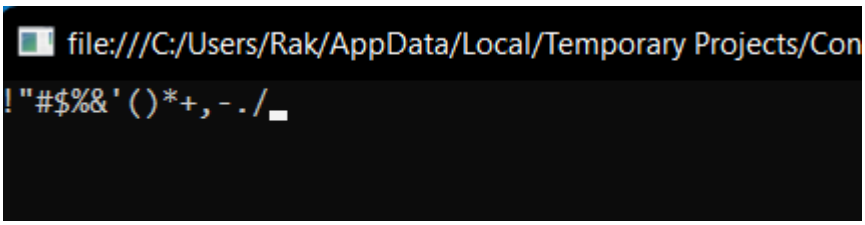
Solution 3f-Vb Functions

```
Module Module1
    Function multiply(num1 As Integer, num2 As Integer  'Function Desciption, the count will be iteratively incremented until the loop structure is complete @cyberwave2024
        Dim count As Integer = 0
        For i = 1 To num2
            count += num1
        Next
        Return count

    End Function
    Sub Main()
        Console.WriteLine(multiply(10, 3))
        Console.ReadLine()
    End Sub
    End Module
```

4 (a) Produce a program that prints out characters whose values range from exclamation to backslash in a list box.
NOTE ASCII value of ! = 33 and / = 47 [4]

Solution 4a-functions

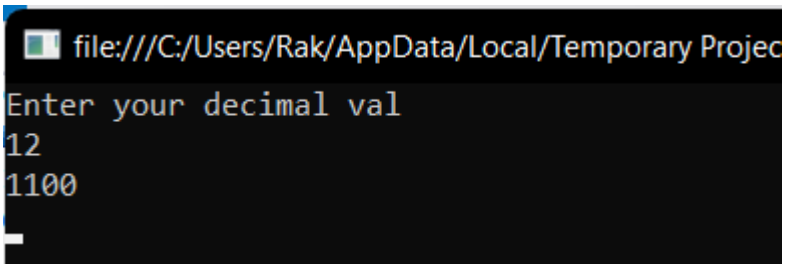
```
Module Module1
    Sub Main()
        For i = 33 To 47
            Console.Write(Chr(i))
        Next
        Console.Read()
    End Sub
    'kkkkk ZIMSEC is not serious
End Module
```



(b) Design an interface and code for a program that converts unsigned integer number (whole number) to its binary equivalent. [9]

Solution 4B

```
Module Module1
    Sub Main()
        Console.WriteLine("Enter your decimal val")
        Dim dec_num As Integer = Console.ReadLine

        Dim binary_string As String

        Do
            binary_string = (dec_num Mod 2).ToString & binary_string
            dec_num \= 2
        Loop While dec_num > 0
        Console.WriteLine(binary_string)
        Console.Read()
    End Sub
End Module
```


file:///C:/Users/Rak/AppData/Local/Temporary Projec
Enter your decimal val
12
1100

(c) Produce a code that inputs 10 items and then outputs the weight of the heaviest and the lightest item. The code should also output the average weight of all items. Produce a screen shot for your answer. [6]

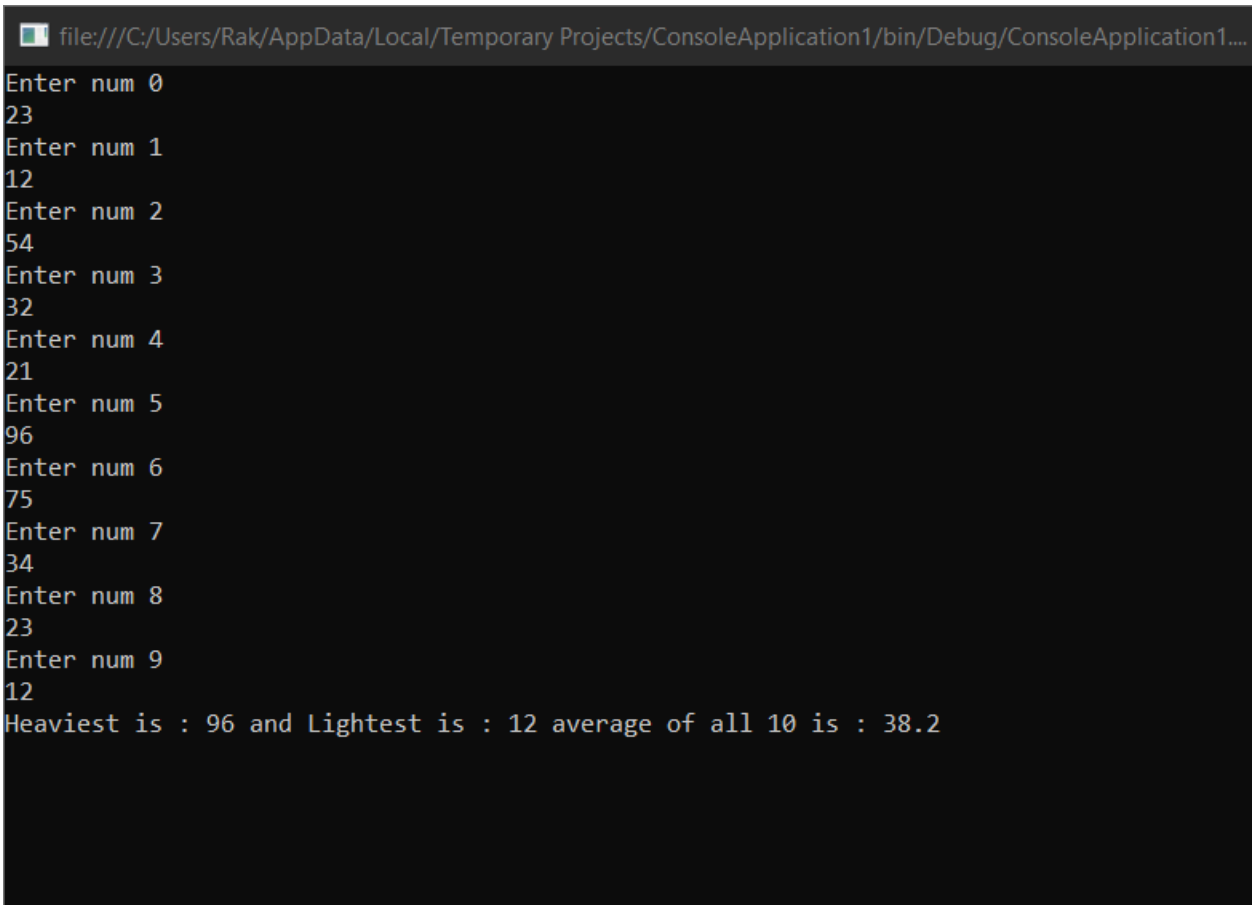
Solution 4C

```
Module Module1
    Sub Main()
        Dim nums(9) As Integer
        Dim count As Integer = 0
        Dim avg As Double

        For j = 0 To 9
            Console.WriteLine("Enter num {0}", j)
            nums(j) = Console.ReadLine
        Next

        For Each num In nums
            count += num
        Next
        avg = count / 10
        Array.Sort(nums)
        Dim large As Integer = nums(UBound(nums))
        Dim small As Integer = nums(LBound(nums))
        Console.WriteLine("Heaviest is : {0} and Lightest is : {1} average of all 10 is : {2}", large, small, avg)

        Console.Read()
    End Sub
End Module
```

----------→ Or look how I used the inbuilt functions here ←----------

```
Module Module1
    Sub Main()
        Dim nums(9) As Integer
        Dim count As Integer = 0
        For j = 0 To 9
            Console.WriteLine("Enter num {0}", j)
            nums(j) = Console.ReadLine
        Next

        For Each num In nums
            count += num
        Next
         Array.Sort(nums)

        Console.WriteLine("Heaviest is : {0} and Lightest is : {1} average of all 10 is : {2}", nums.Max, nums.Min,
    nums.Average)

        Console.Read()
    End Sub
End Module
```

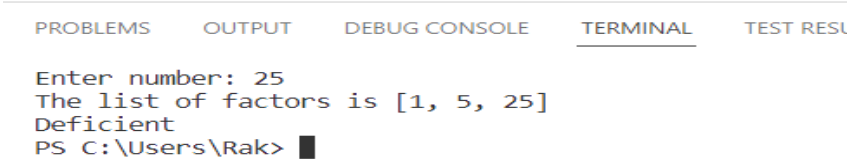
nums.Max, nums.Min, nums.Average


file:///C:/Users/Rak/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/ConsoleApplication1....
Enter num 0
23
Enter num 1
12
Enter num 2
54
Enter num 3
32
Enter num 4
21
Enter num 5
96
Enter num 6
75
Enter num 7
34
Enter num 8
23
Enter num 9
12
Heaviest is : 96 and Lightest is : 12 average of all 10 is : 38.2

(d) A perfect positive number is called perfect if the sum of its factors is equal to twice the number. It is called abundant if the sum of its factors is greater than twice the number. It is called deficient if the sum of its factors is less than twice the number.
Produce a program code that reads a number and prints out whether it is perfect, abundant or defient. [9]

Solution 4D

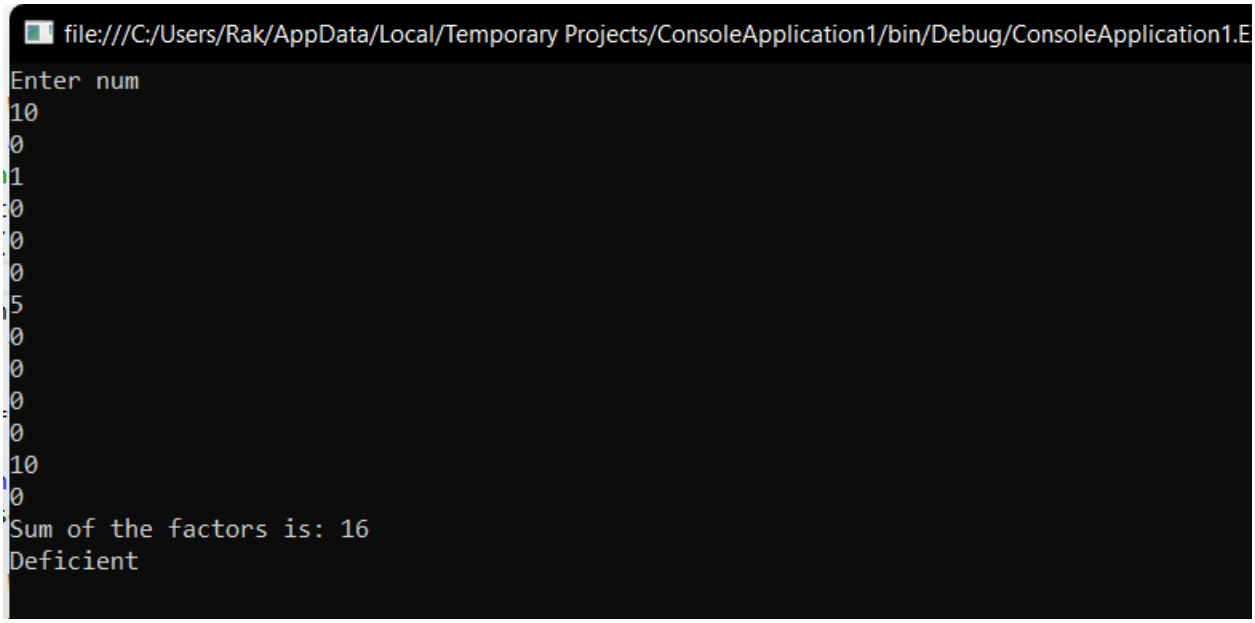
Case 1-Python




```
def factorize(num):
   factors=[]
   for i in range(1,num+1):
      if num % i==0:
         factors.append(i)
   return factors

num=int(input("Enter number: "))
ls=factorize(num)
print(f"The list of factors is {ls}")
sum_fact=sum(ls)

if sum_fact==2*num:
   print("Perfect")
elif sum_fact > 2*num:
   print("abundant")
else:
   print("Deficient")
```

Case2-Vb.Net

```
Module Module1

    Sub Main()
        Console.WriteLine("Enter num")
        Dim num As Integer = Console.ReadLine
        'i intend to store the factors in an array of at least [num] slots
        Dim factors(num + 1) As Integer
        factors(num) = num
        For i = 1 To num + 2
            If num Mod i = 0 Then
                factors(i) = i
            End If
            i += 1
        Next
        For Each v In factors
            Console.WriteLine(v)
        Next
        Console.WriteLine("Sum of the factors is: " & factors.Sum)
        Dim sum_of_fact As Integer = factors.Sum
        If sum_of_fact = 2 * num Then
            Console.WriteLine("Perfect")
        ElseIf sum_of_fact > 2 * num Then
            Console.WriteLine("abundant")
        Else
            Console.WriteLine("Deficient")
        End If
        Console.ReadLine()
    End Sub

End Module
```



(e) The employees of a company all have 8-character personnel numbers.
An algorithm will do the following:
Enter the number and then test if it contains exactly 8 characters, less than 8 characters or more than 8 characters.
Display appropriate messages for each condition.

(i) Produce an algorithm in the form of a pseudocode for the above narration. [6]

Solution 4E

1. START
2. Enter Employee Number
3. Read the Employee Number
4. IF Employee_numer length < 8 Then:

   Display("EN less than 8")
5. Else IF Employee_number >8 Then:

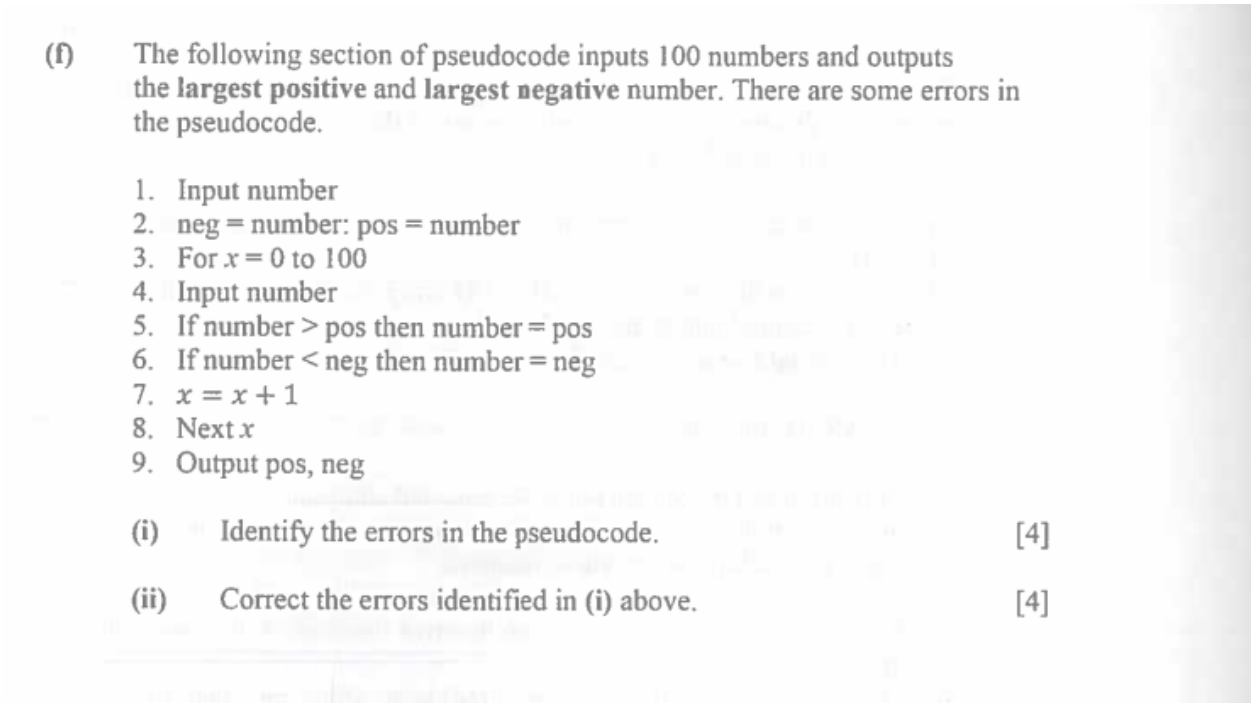
   Display("EN greater than 8")
6. Else Display("accepted")
7. END

(ii) produce a code for your pseudocode in (i) above. [8]

Solution 4E[ii]

```
Module Module1
   Sub Main()
      Dim id_num As String
      Console.WriteLine("Enter your employee number")
      id_num = Console.ReadLine
      If Len(id_num) < 8 Then
         Console.WriteLine("characters are less than required")
      ElseIf Len(id_num) > 8 Then
         Console.WriteLine("characters are more than required")
      Else
         Console.WriteLine("personel number is valid")
      End If
      Console.Read()
   End Sub
End Module
```

HOMEWORK:WHY DID CYBERWAVE MAKE ID_NUM A STRING???



(f) The following section of pseudocode inputs 100 numbers and outputs the **largest positive** and **largest negative** number. There are some errors in the pseudocode.

1. Input number
2. neg = number: pos = number
3. For $x = 0$ to 100
4. Input number
5. If number > pos then number = pos
6. If number < neg then number = neg
7. $x = x + 1$
8. Next x
9. Output pos, neg

(i) Identify the errors in the pseudocode. [4]

(ii) Correct the errors identified in (i) above. [4]

Solution 4F

1. Line 1 and 4 are the same, the variables should be unique i.e input test_num
2. Line 2, separate the lines to initialise
3. Line 5, ~~num=pos,~~ pos=num
4. Line 6~~, num=neg~~, neg=num

ii)

1. input test_num
2. neg=test_num
3. pos=test_num
4. For x= 0 to 100
5. Input number
6. If number>pos then pos=number
7. If number<neg then neg=number
8. X=x+1
9. Next x
10. Output pos
11. Output neg

SECTION C

Section C [30 marks]

Answer any **one** question from this section.

5 (a) Gabaza video club maintains a database of movies to be rented out. Without normalisation, all information is stored in one table as shown below.

Member ID	Title	Full Name	Address	Movie Rented	Category
1	Mrs	Jane Moyo	902 Mutare	Pirates Titans	Action, Action
2	Mr	Robert Dube	54 Gweru	Forgotten Daddy	Educational Comedy
3	Mrs	Anesu Ncube	100 Airport	Titans	Action

Normalise the above table to second normal form. [14]

Its supposed to be Educational, Comedy

Solution 5A-SQL

Food for thought…

1. A table is in 1NF if and only if an attribute (column) of a table do not hold multiple values i.e it should have only atomic values.

2. The table in 1NF will lead to too much duplication of data, therefore a table will be in second normal form (2NF) if and only if it is in 1NF and every non-key attribute is fully dependent on the primary key i.e. there are no partial dependencies.

Step 1- 1NF (No multiple values Atomicity)

Member ID	Title	Full Name	Address	Movie Rented	Category
1	Mrs	Jane Moyo	902 Mutare	Pirates	Action
1	Mrs	Jane Moyo	902 Mutare	Titans	Action
2	Mr	Robert Dube	54 Gweru	Forgotten	Educational
2	Mr	Robert Dube	54 Gweru	Daddy	Comedy
3	Mrs	Anesu Ncube	100 Airport	Titans	Action

Step2-2NF(No partial Dependancy-(Reduce Duplication))

Rentals

Member ID	Movie ID
1	1
1	2
2	3
2	4
3	2

Table2-Members Table

Member 1D	Title	Full Name	Address
1	Mrs	Jane Moyo	902 Mutare
2	Mr	Robert Dube	54 Gweru
3	Mrs	Anesu Ncube	100 Airport

Table2-Movies Table

Movie ID	Movie Rented	Category
1	Pirates	Action
2	Titans	Action
3	Forgotten	Educational
4	Daddy	Comedy

*Please note that there are several ways to answer this question, ZIMSEC will only check to see if your final result follows the policy of database normalization. See **ACID** concept*

(b)

Student-ID	F Name	Surname	Sport	Venue
4241	Sibo	Nare	cricket	Harare
4741	Takunda	Mafu	soccer	Harare
4858	Farai	Muganu	volley	Kwekwe
4740	Sean	Gutsa	volley	Mutare
4848	Layla	Tanguni	cricket	Kwekwe
4742	Sindiso	Moyo	cricket	Mutare
4444	Thubani	Malahleka	soccer	Mutare

(i) Using DBMS of your choice, identify the primary key and create the sport table above. [4]

Solution 5B

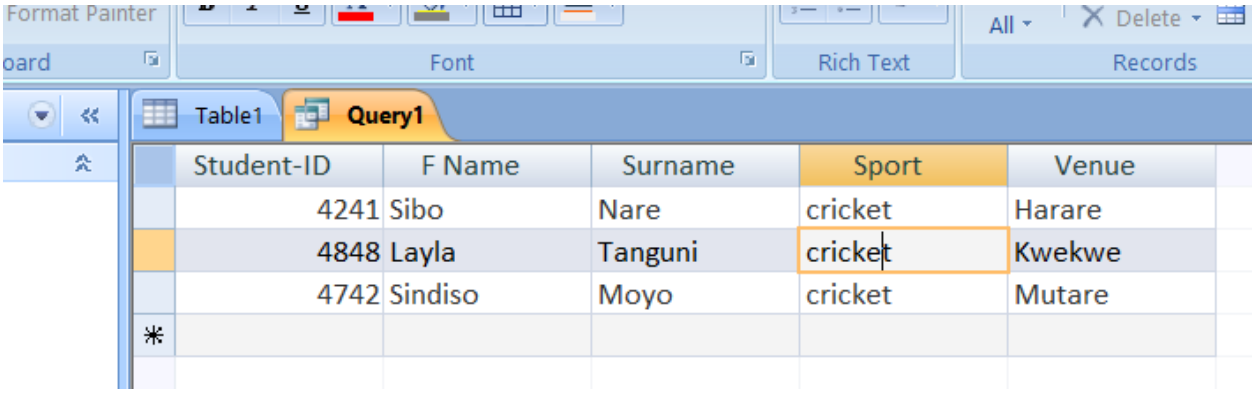
Just copy the table on Microsoft Access

Primary key-**Student-ID**

Sport Table				
Student-ID	F Name	Surname	Sport	Venue
4241	Sibo	Nare	cricket	Harare
4741	Takunda	Mafu	soccer	Harare
4858	Farai	Muganu	volley	Kwekwe
4740	Sean	Gutsa	volley	Mutare
4848	Layla	Tanguni	cricket	Kwekwe
4742	Sindiso	Moyo	cricket	Mutare
4444	Thubani	Malahleka	soccer	Mutare

(ii) Create a query to display the records of students who play cricket, print the screen shot and the SQL code. [4]

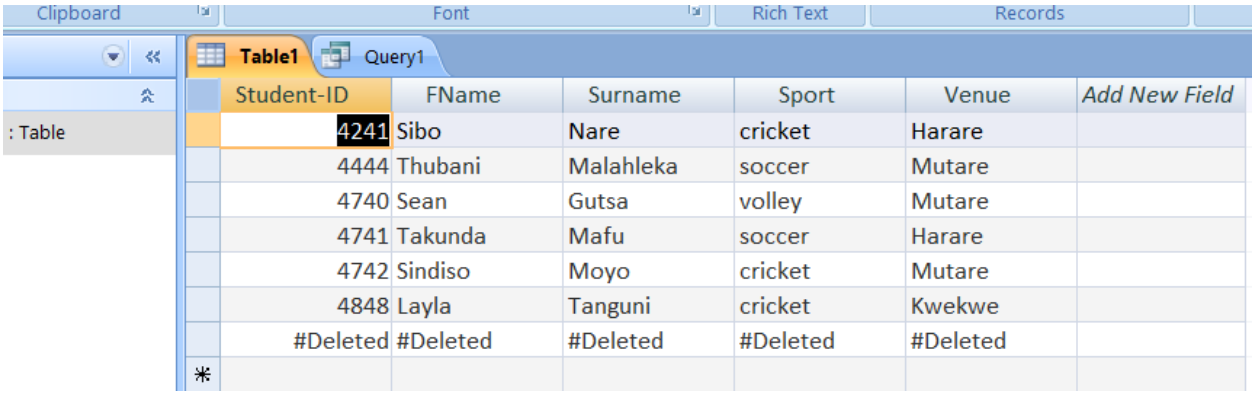
```
SELECT * FROM Table1
WHERE Sport="cricket"
```



Student-ID	F Name	Surname	Sport	Venue
4241	Sibo	Nare	cricket	Harare
4848	Layla	Tanguni	cricket	Kwekwe
4742	Sindiso	Moyo	cricket	Mutare

(iii) Produce the delete query to remove Farai Muganu from the table. Print the screen shot and SQL code. [4]

```
DELETE * FROM Table1
WHERE FName = "Farai" AND Surname="Muganu"
```



Student-ID	FName	Surname	Sport	Venue	Add New Field
4241	Sibo	Nare	cricket	Harare	
4444	Thubani	Malahleka	soccer	Mutare	
4740	Sean	Gutsa	volley	Mutare	
4741	Takunda	Mafu	soccer	Harare	
4742	Sindiso	Moyo	cricket	Mutare	
4848	Layla	Tanguni	cricket	Kwekwe	
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	

(iv) Produce the updated query to add the following record 4748, Kenneth, Shawa, soccer, Gweru. Print the result and the SQL. [4]

```
INSERT INTO Sport_Table
        ([Student-ID], FName, Surname, Sport, Venue)
VALUES ('4748', 'Kenneth', 'Shawa', 'soccer', 'Gweru')
```

Student-ID	FName	Surname	Sport	Venue
4241	Sibo	Nare	cricket	Harare
4741	Takunda	Mafu	soccer	Harare
4740	Sean	Gutsa	volley	Mutare
4848	Layla	Tanguni	cricket	Kwekwe
4742	Sindiso	Moyo	cricket	Mutare
4444	Thubani	Malahleka	soccer	Mutare
1	4748	Kenneth	Shawa	Gweru
4748	Kenneth	Shawa	soccer	Gweru

6 A school website has a relational database, SCHOOLWEB, that stores the website's information.
The Database has 3 tables to store student details, image details and text that they post. The tables are as shown below:

Student (**Student ID**, Surname, First Name, Date of Birth)
Photo (**Photo ID**, Student ID, Comment, Upload Date)
Text Post (**Post ID**, Student ID, Date of post, The Text)

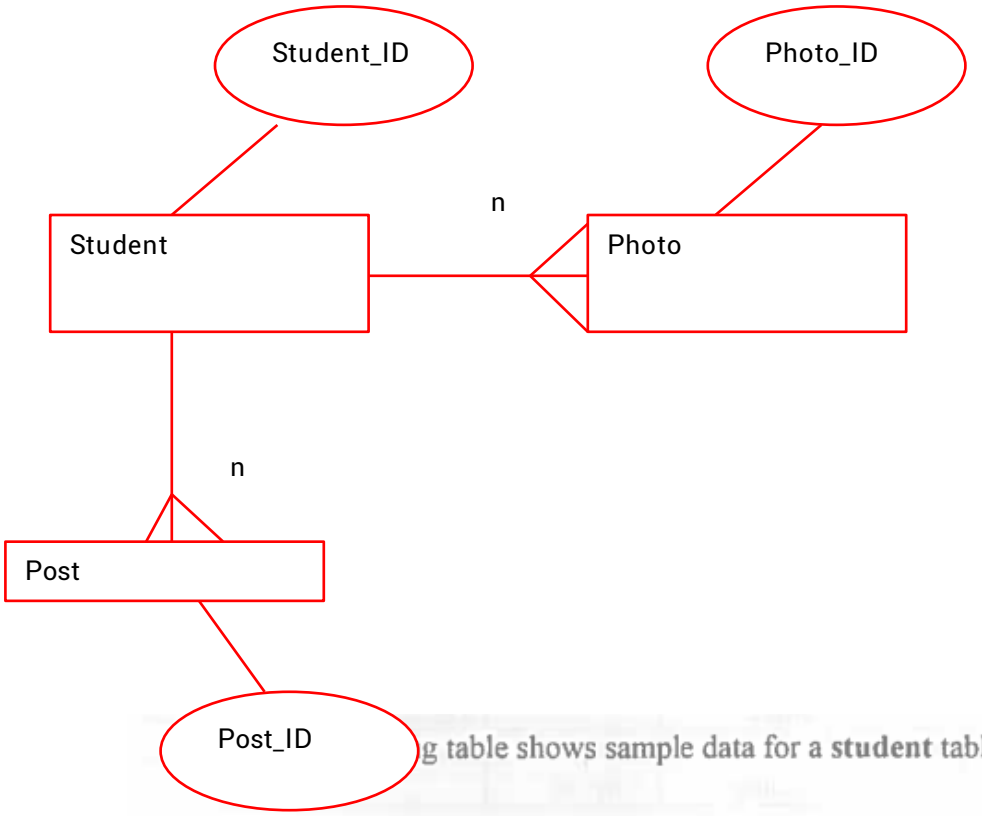
(a) **(i)** Explain how the relationship between the tables **Student** and **Photo** has been implemented. [2]

(ii) Draw the ERD to show the relationship between the 3 tables. For fields, only include the primary keys on the ERD. [7]

Solution 6

1. A student can have many photos [one to many]
2. A student can have many text posts [one to many]
3. A photo belongs to one student [many to one]
4. A text post belongs to one student [many to one]

Solution 6[ii]



g table shows sample data for a **student** table.

Student ID	First Name	Surname	Date of Birth
tino123	Tinotenda	Mutamba	01/01/95
jackmoyo	Jacob	Moyo	23/02/56
muteroa	Anesu	Mutero	03/03/97
bigboss3	Mazvita	Tongogara	04/03/82

(i) Produce an SQL statement to create the student table above. [5]

Solution 6B

```
CREATE TABLE Student_table(
Student_ID VARCHAR(25),
First_Name VARCHAR(25),
Surname VARCHAR(25),
Date_of_Birth Date
);
```

Please this table is not complete, use the knowledge I gave you on SQL to insert the records into the table

(ii) Produce an SQL statement to add a new field, country, to the student table. [2]

```
ALTER TABLE student
ADD country VARCHAR(255);
```

(iii) Produce an SQL statement to display all attributes for students who were born before 1990. [3]

```
SELECT * FROM student
WHERE date_of_birth<"1990-01-01";
```

(iv) Produce an SQL statement to display all records in descending order of student surname. [3]

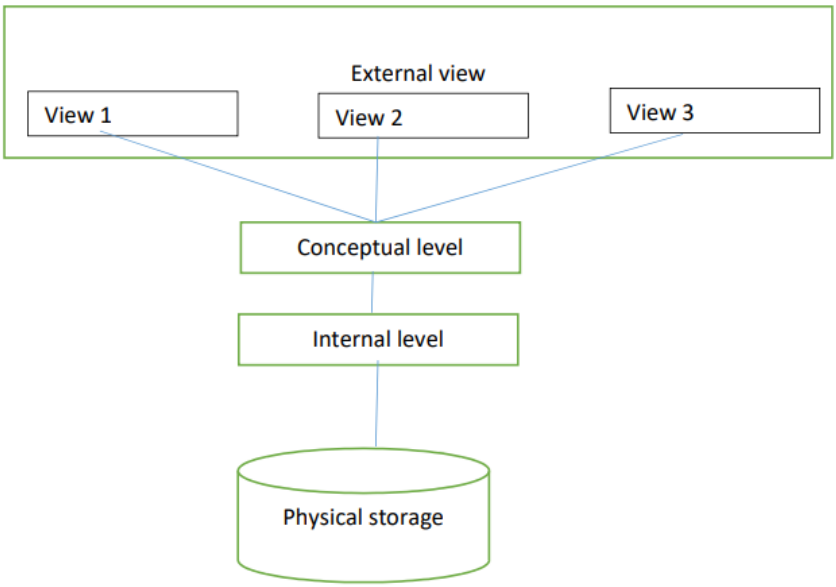
SELECT * FROM student

ORDER BY surname DESC;

(c) Draw a well-labelled diagram showing the 3-layered schema of the Database Management System (DBMS).
Give a brief description of each layer. [8]

See CYBERSPACE 2023 sneak peek paper 1 for detailed explanation

b) illustrate with the aid of a diagram the ANSI three level architecture of the database approach



1. External level (view)
    - Users' view of the database.
    - It refers to different views of the data seen by each user of the database.
    - Each user of a database has his/her own views of the database, e.g. general database users
    - see and describe the database differently as done by the managers because they have different views of the same database.
    - Describes that part of database that is relevant to a particular user Thus other users cannot view some data that the managers can views
2. Conceptual (logical) Level (view)
    - It is seen as an integration of all user views of the data
    - This is an abstract representation of the whole database.

- Describes what data is stored in database and relationships among the data
- Describes the data as seen by the application that is making use of the DBMS
- Involves identification of entity types, unique identifiers,
- Logical level describes what data are stored in the database
- Describes what relationships exist among those data.
- Logical level describes the entire database in terms of a small number of simple structures
- .Database administrator uses the logical level of abstraction

3. Internal level (view)

- Physical representation of the database on the computer
- Describes how the data is stored in the database
- It refers to the structure used for storage of data.
- Also called the physical level
- It encompasses the logical arrangements of the data for storage.
- Describes how data will be stored on the physical media
- The physical level describes complex low-level data structures in detail

Hope you Enjoyed Physically ----------

Philippians 4:6-7 "Do not be anxious about anything, but in every situation, by prayer and petition, with thanksgiving, present your requests to God. And the peace of God, which transcends all understanding, will guard your hearts and your minds in Christ Jesus. ----------0716057680 LIONEL CYBERWAVE

ONLINE CLASSES ARE ALWAYS OPEN, REACH OUT TO US ON

ccyberwavee@gmail.com

tinashechenyika8@gmail.com

0716057680

-----GOD BLESS YOU ENGINEER-----

YOUTUBE CHANNEL: **ZIMSEC COMPUTER SCIENCE TUTORIALS**