PROBLEM SOLVING AND PROGRAM DESIGN

IDENTIFY WAYS OF REPRESENTING ALGORITHMS.
OBJECTIVES:

Identify ways of representing algorithms: 

**Content**

Representation of algorithms as Pseudocode or Flowcharts; use of flow chart symbols: input/output process, decision, directional arrows. Use of: Read, Input, Store, Write, Output, Display, If Then; If Then Else; For loop; While loop (Questions which require nested conditionals or nested loops will not be asked)
During the development of an algorithm, the language gradually progresses from English towards a notation that resembles that of a programming language. An intermediate notation called *pseudocode* is commonly used to express algorithms. Algorithms can also be expressed as *flowcharts*. A flowchart is a pictorial representation of an algorithm.
Acceptable words for Input/Output in pseudocode:

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get</td>
<td>Display</td>
</tr>
<tr>
<td>Accept</td>
<td>Print</td>
</tr>
<tr>
<td>Input</td>
<td>Output</td>
</tr>
<tr>
<td>Read</td>
<td></td>
</tr>
</tbody>
</table>
Flowcharts

Flowcharts use special geometrical objects to designate steps of a program, which are: input, processing and output. A parallelogram is used to represent the input operation as well as the output operation, and a rectangle is used to represent a processing/assignment statement. A diamond is used to represent a decision (If Then Else) structure. An elliptical shape is used to represent the terminal indicators, START OR STOP. Directional arrows is used to indicate the flow of the logic in the algorithm.
The four basic shapes are:

- **Input/Output**
- **Processing/Assignment**
- **Decision**
- **Start/Stop**
Below are examples of how the various control structures are depicted in a flowchart: **Sequence**

- Do A
- Do B
Selection (The If Then Else Construct)

If C is true then
   do E
Else
   do D

No

C

Yes

D

E
Loop

Loop (Repetition)

While F is true do G
Pseudocode Version of the Average Algorithm

Read Num1, Num2, Num3
Average = (Num1 + Num2 + Num3)/3
Print Average
Stop.
Flowchart version of the Average Algorithm

Start

Read Num1, Num2, Num3

Aver = (Num1 + Num2 + Num3)/3

Print Aver

Stop
Problem 10

Design an algorithm that accepts two values, the hourly rate and the number of hours worked by an employee. If the number of hours exceeds 40 then the excess hours should be paid at an overtime rate of twice the hourly rate. Calculate the wages (including overtime, if any) due to the employee.
The first step is to define the solution as follows:

<table>
<thead>
<tr>
<th>INPUT</th>
<th>PROCESSING</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate, HoursWorked</td>
<td>1) Calculate overtime pay if any</td>
<td>Wages</td>
</tr>
<tr>
<td></td>
<td>2) Calculate wages</td>
<td></td>
</tr>
</tbody>
</table>
Solution Step 2

Possible Pseudocode version of the algorithm:

This algorithm calculates the wages due to an employee.
Read Rate, HoursWorked
BasicPay = Rate * 40
OvertimeHours = HoursWorked – 40
If (OvertimeHours > 0) then
    Wages = BasicPay + (OvertimeHours * 2 * Rate)
Else
    Wages = Rate * HoursWorked
EndIf
Print Wages
Stop.
BasicPay = HoursWorked * 40

OvertimeHours = HoursWorked - 40

Wages = BasicPay + (OvertimeHours * 2 * Rate)

Overtime Hours > 0

Wages = HoursWorked * Rate

Print Wages

Stop
Comparison/Selection/Decision Statements

It sometimes becomes necessary when processing data to compare the value of a variable to the value of another variable or to compare the value of a variable to that of a constant. The following *Relational Operators* are used to make such comparisons:

<table>
<thead>
<tr>
<th>Relational Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equal to</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Not equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt; =</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt; =</td>
<td>Less than or equal to</td>
</tr>
</tbody>
</table>
A *Condition* is an expression that when evaluated gives either a *True* or a *False*. This expression is called a *Boolean* expression. These conditions use the relational operators between two variables or between a variable and a constant.

Examples:
Boys in Class $\geq 35$
NumberOfBoys $\neq$ NumberOfGirls
Boolean Operators

When selection or decisions is based upon one or more expressions being True or False, it is possible to combine the expressions together using Boolean Operators **AND** or **OR**.

If the **AND** operator is used both conditions must be met in order for the total expression to be true.

If the **OR** expression is used either condition must be met in order for the total expression to be true.
A club plays cricket only on Sundays and only if it is not raining. Write an algorithm to accept the day and weather and print “Game on” if it is a suitable day for playing. What happens with each of the two operators.

Solution:
Input Day, Weather
If (Day="Sunday") And (Weather="No Rain") Then
  Print “Game ON”
EndIF
Stop

If either condition is not met “Game On” would not be printed
Example of the OR Operator

If we had used the or operator instead then the algorithm would read as follows:

Input Day, Weather
If (Day="Sunday") OR (Weather="No Rain") Then
    Print “Game ON”
EndIF
Stop

If either condition is met “Game On” would be printed. That would mean that “Game On” would be printed if it is Sunday or on any other day of the week once there is no rain.
Question 1

Write an algorithm to enter the base and height of a triangle and find and print the area. Write a pseudocode version and a flowchart version of the solution.
Algorithm AreaTriangle
This algorithm accepts values for the Base and Height of a triangle and Prints the area.

Read Base, Height
Area = (Base * Height)/2
Print Area
Stop.
Start

Read Base, Height

Area = (Base * Height)/2

Print Area

Stop
Write an algorithm to enter a number, double the number and output the result. Write a pseudocode version and a flowchart version of the solution.
Pseudocode version

**Algorithm Double**
This algorithm accepts a number and doubles it and outputs the result.

Read Num
Double = Num * 2
Print Double
Stop.
Start

Read Num

Double = Num * 2

Print Double

Stop
Question 3

Persons under 18 are not allowed in at ‘Shutters Night Club’. Write an algorithm to read a person’s age and if it is under 18 the algorithm should output ‘Underage person’. If the person is 18 years or over the algorithm should output “Person allowed”. Write a pseudocode version and a flowchart version of the solution.
Algorithm Shutters
This algorithm checks a person’s age and check to see if they are allowed in the night club.

If (Age < 18) Then
    Print “Underage Person”
Else
    Print “Person Allowed”
EndIf

Stop.
Flowchart Version

Start

Read Age

If (Age < 18) Then

No

Print "Person Allowed"

Yes

Print "Underage Person"

Stop
Write a program to input a number N. If the number is greater than 50 subtract 10 from the number, otherwise multiply the number by 2 and add 5. Print the number and the result.
Algorithm Number $N$
This algorithm accepts a number $N$ manipulates it and prints the result.
Read $N$
If $(N > 50)$ Then
    $N = N - 10$
Else
    $N = N \times 2$
    $N = N + 5$
EndIf
Print $N$
Stop.
Flowchart Version

Start

Read N

If (N > 50) Then

Yes: N = N - 10
No: N = N * 2, N = N + 5

Print N

Stop
THE END